

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

На правах рукопису

Зіарманд Артур Нісарович

УДК 658:512.011: 681.326: 519.713

МОДЕЛІ І МЕТОДИ КІБЕРФІЗИЧНОГО КОМП'ЮТИНГУ
ДЛЯ ЦИФРОВОГО МОНІТОРИНГУ І ХМАРНОГО УПРАВЛІННЯ
ТРАНСПОРТОМ

05.13.05 – комп'ютерні системи та компоненти

ДИСЕРТАЦІЯ

на здобуття наукового ступеня кандидата технічних наук

Цей примірник дисертації ідентичний за змістом
з іншими примірниками, що подані до спеціалізованої
вченої ради Д 64.052.01

Учений секретар спеціалізованої вченої ради Д 64.052.01

Є.І. Литвинова

Науковий керівник:

доктор технічних наук, проф.

Хаханов Володимир Іванович

Харків – 2017

АНОТАЦІЯ

Зіарманд Артур Нісарович. Кваліфікаційна наукова праця на правах рукопису – Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.13.05 «комп'ютерні системи та компоненти». – Харківський національний університет радіоелектроніки, Міністерство освіти і науки України, Харків, 2017.

Мета дослідження – підвищення якості та безпеки дорожнього руху за рахунок створення кіберфізичної моделі комп'ютерної online взаємодії водія з хмарними сервісами керування автомобілем на основі цифрового моніторингу дорожньої інфраструктури і транспортних потоків, при використанні розумних сенсорів, засобів телекомунікації та навігації.

Задачі дослідження:

1) Розробка моделі транспортного комп'ютерного – кіберфізичної взаємодії автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

2) Створення метрики і критеріїв оцінювання якості інфраструктури для online аналізу кіберфізичного простору, пошуку квазі-оптимального маршруту і зменшення часу його виконання.

3) Розробка методу аналізу кіберфізичної інфраструктури дорожнього руху на основі алгоритму Дейкстри для online пошуку квазіоптимального маршруту в умовах виникнення колізій.

4) Створення архітектури розумного хмарного світлофора на основі використання логічних операцій, що дає можливість істотно збільшити пропускну здатність перехрестя доріг.

5) Практична реалізація моделей і методів цифрового моніторингу і хмарного керування автомобілем в рамках створення кіберфізичної системи дорожнього руху та подальше їх тестування.

Наукова новизна:

1) Вперше запропоновано модель транспортного комп'ютингу, яка характеризується кіберфізичною взаємодією автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

2) Вперше запропоновано архітектуру розумного хмарного світлофора, яка характеризується використанням логічних операцій і часом простою зеленого сигналу, що дає можливість істотно збільшити пропускну здатність транспортних потоків на перехресті доріг.

3) Удосконалено метрику і критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання.

4) Удосконалено алгоритм Дейкстри, який відрізняється можливістю аналізу кіберфізичної інфраструктури дорожнього руху для online пошуку квазіоптимального маршруту транспортного засобу в умовах виникнення колізій.

Практична реалізація моделей і методів цифрового моніторингу і хмарного керування автомобілем виконана в рамках створення і верифікації програмних компонентів кіберфізичної архітектури дорожнього руху «Cloud Traffic Control» з подальшим тестуванням модельних потоків транспортних засобів на ділянках дорожньої інфраструктури.

Отримані в процесі виконання дослідження наукові висновки і результати є достовірними, що підтверджується проведеними експериментами, тестуванням і верифікацією запропонованих моделей і методів моніторингу та управління дорожнім рухом.

Практична значимість наукових досліджень підтверджується істотним зменшенням виконання маршрутів руху при моделюванні реальних процесів на фрагментах моделей дорожньої інфраструктури. Результати дисертації в складі моделей, методів і архітектури кіберфізичної інфраструктури

впроваджені в навчальний процес Харківського національного університету радіоелектроніки (акт про впровадження від 03.09.2017) при читанні курсів: «Дискретна математика», «Cloud-Fog кіберфізичні системи». Розроблена архітектура розумного хмарного світлофора, а також модель транспортного комп'ютингу можуть бути імплементовані як компонент проекту при синтезі хмарного сервісу (Довідка від 28.09.2017, ТОВ «Перший інститут надійного програмного забезпечення»).

Результати дисертаційної роботи відображено у 18 друкованих працях: 7 статей, серед яких 3 у наукових журналах, що входять до «Переліків наукових фахових видань України» (з них 3 – у міжнародних наукометричних базах), 3 статті в міжнародних наукових журналах за кордоном (з них 1 – в міжнародній наукометричній базі Scopus, 1 – у міжнародній наукометричній базі ORCID); а також 11 публікацій у міжнародних наукових конференціях (з них 7 – за кордоном, 6 входять до наукометричної бази Scopus). Здобувач має 7 публікацій, що входять до наукометричної бази Scopus, та має індекс Хірша $h=1$.

СПИСОК ОПУБЛІКОВАНИХ РОБІТ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Список публікацій здобувача, в яких опубліковані основні наукові результати дисертації:

1. Немченко В.П. Моделирование сетевых протоколов при построении тестовых последовательностей / В.П. Немченко, А.Н. Зиарманд, Ю.А. Чепелев // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2011. – №5(90). – С. 18-21. (Індексується GoogleScholar, Реферативна база “Наукова періодика України”, РИНЦ (eLibrary), National Library of Ukraine named after Vernadsky).

2. Немченко В.П. Использование энергосберегающих технологий в современных сетях / В.П. Немченко, А.Н. Зиарманд, А.С. Изотов // Вестник Херсонского национального технического университета. – 2012. – №1(44). – С. 146-148. (Індексується РИНЦ (eLibrary), Google Scholar, National Library of Ukraine named after Vernadsky).

3. *Зиарманд А.Н.* Модели и методы мониторинга и управления транспортом / *А.Н. Зиарманд, В.И. Хаханов* // Радиоэлектроника и информатика. – 2016. – №3(74). – С. 71-87. (Входит до міжнародних наукометричних баз Index Copernicus, Google Scholar, OECSP, OAJI, Scholar Steer, SIS, Cyberleninka, CiteFactor, TIU Hannover, I2OR).

4. Vladimir Hahanov. Internet-driven Cyber Control of Traffic / Vladimir Hahanov, Wajeb Gharibi, Svetlana Chumachenko, Evgeniya Litvinova, Vladimir Miz, *Arthur Ziarmand* // Australian Journal of Scientific Research. – 2014. – Volume IV, No 1(5). – Pp. 217-224.

5. Hahanov V. Cloud-Driven Traffic Monitoring and Control Based on Smart Virtual Infrastructure / V. Hahanov, Wajeb Gharibi, E. Litvinova, S. Chumachenko, *A. Ziarmand*, I. Englesi, I. Gritsuk, V. Volkov, A. Khakhanova // SAE Technical Paper. USA. 2017-01-0092, 2017. 6 p. doi:10.4271/2017-01-0092. (Входить до міжнародної наукометричної бази Scopus, ORCID).

6. *Зиарманд А.Н.* Модель облачного сервиса для поиска оптимального пути / С. В. Чумаченко, Е. И. Литвинова, В. И. Хаханов, *А. Н. Зиарманд* // Paradigmata poznání. Vědecko vydavatelské centrum «Sociosféra-CZ», s.r.o., Praha, Česká republika. – 2017. – Вып. 3. – С.63-83. doi: 10.24045/pp.2017.3.6 (The journal is indexed by Electronic Research Library, Russia; Research Bible, China; Scientific Indexing Services, USA; Cite Factor, Canada; General Impact Factor, India; Scientific Journal Impact Factor, India; CrossRef, USA; ORCID, USA).

7. *Зиарманд А.Н.* Теоретическая суть проекта «Smart Roads» / *А.Н. Зиарманд* // Автоматизированные системы управления и приборы автоматики. – 2013. – Вып. 162. – С. 28-34. (Входит до міжнародних наукометричних баз Google Scholar, Cyberleninka). <http://cyberleninka.ru/article/n/teoreticheskaya-sut-proekta-smart-roads>

Результати, які засвідчують апробацію матеріалів дисертації:

8. Hahanov V. I. Cloud traffic control system / V. I. Hahanov, O. A. Gus, *A. Ziarmand*, Ngene Christopher Umerah, A. Arefjev // Proc. of the IEEE East-West

Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 72-76. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

9. *Ziarmand A. Smart road infrastructure / A. Ziarmand // Proc. Of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 430-434. (Work in Progress). (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

10. *Ziarmand A. Cloud Service for Traffic Control / Artur Ziarmand, Vladimir Hahanov, Volodymyr Miz, Anastasya Hahanova, Aleksey Priymak // Proc. of the XII International IEEE Conference “Modern Problems of Radio Engineering, Telecommunications, and Computer Science”. 2014. – Lviv-Slavske, Ukraine. – P. 557-559. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

11. *Hahanov V. Cyber physical system – smart cloud traffic control / V. Hahanov, Wajeb Gharibi, L.S. Abramova, S. Chumachenko, E. Litvinova, A. Hahanova, V. Rustinov, V. Miz, A. Zhalilo, A. Ziarmand // Proc. of the IEEE Design & Test Symposium. – 2014. – Kiev, Ukraine. – P. 49-66. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

12. *Ziarmand A. Transport monitoring and control systems / A. Ziarmand, D. Kucherenko, T. Soklakova // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 474-477. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

13. *Зиарманд А.Н. Формальная модель киберсистемы / А.Н. Зиарманд, И.И. Чугуров // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2014. – С. 65.*

14. *Зиарманд А.Н. Формальна модель хмарного управління транспортом / А.Н. Зиарманд, К.Э. Мороз // Материалы XIX Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2015. – С. 4.*

15. *Зиарманд А.Н. I-Cloud Traffic Control / А.Н. Зиарманд, В.А. Мизь // 6-я Международная студенческая конференция и конкурс научных работ по*

вопросам информационной безопасности «CyberSecurity for the Next Generation». – 2014. – “Kaspersky Office”, Москва, РФ. – С. 13.

Публікації, які додатково відображають наукові результати дисертації:

16. Chumachenko S. Quantum data structures for SoC design / S. Chumachenko, A. Shkil, A. Hahanova, A. Ziarmant, A. Pryimak // Proc. of the 13th International Conference “IEEE Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”. – 2015. Polyana-Lviv. – Ukraine. – P. 355-357. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

17. Soklakova T. Big data visualization in smart cyber university / T. Soklakova, A. Ziarmant, S. Osadchyieva // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 469-473. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

18. *Зиарманд А.Н.* Анализ компетентности пользователя компьютерной системы как диагностический эксперимент / Зиарманд А.Н., Кучеренко Д.Е. // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2011. – С. 40-41.

Ключові слова: комп'ютинг, кіберфізична система, розумне хмарне управління транспортом, метрика, критерії якості, моделі, методи, програмні компоненти, модифікований алгоритм Дейкстри, розумна архітектура, модель транспортного комп'ютингу.

SUMMARY

Ziarmand Artur Nisarovich. Qualifying scientific work on the rights of the manuscript – A dissertation for defence the degree of a candidate of technical sciences (doctor of philosophy) in the specialty 05.13.05 "computer systems and components". – Kharkiv National University of Radio Electronics, Ministry of Education and Science of Ukraine, Kharkiv, 2017.

Research goal – improvement the road traffic quality and safety through the development of a cyber-physical model of computing online interaction between driver and cloud vehicle management services based on digital monitoring of road infrastructure and traffic flows, using intelligent sensors, telecommunications and navigation.

Research tasks:

1) Transport computing model development – cyber-physical interaction between car and cloud service by evolutionary movement the streetlight from physical to virtual space for digital traffic flows monitoring and quasi-optimal traffic control.

2) Creating an intelligent cloud traffic streetlight architecture based on the usage of logical operations, which makes it possible to significantly increase the cross-roads throughput.

3) Creating metrics and criteria for assessing the quality of infrastructure for online analysis of cyberspace, searching for a quasi-optimal route and reducing the time of its execution.

4) Development of a method for analysing the cyber-physical road traffic infrastructure based on the Dijkstra algorithm for online quasi-optimal route search under the conditions of collision.

5) Practical realization of digital monitoring and cloud management models and methods within the frameworks of creation the cyber-physical traffic system and their subsequent testing.

Scientific novelty:

1) For the first time, a model of transport computing was proposed, which is characterized by the cyber-physical interaction between car and cloud service by the evolutionary movement the traffic light from physical to virtual space for digital monitoring of traffic flows and quasi-optimal traffic control.

2) For the first time, the architecture of a smart cloud traffic signal has been proposed, which is characterized by the usage of logical operations and the time of idle green signal, which makes it possible to significantly increase the throughput of traffic flows at the crossroads of roads.

3) The metric and criteria for assessing the quality of infrastructure are improved, which is distinguished by the possibility of cyberspace online analysis to find the quasi-optimal route and reduce the time of its execution.

4) The Dijkstra algorithm is improved, which is characterized by the possibility of analysing the cyber-physical traffic infrastructure for online search of the quasi-optimal vehicle route in the event of collisions.

The practical implementation of models and methods of digital monitoring and cloud car management was carried out in the framework of the creation and verification of the software components of the cyberspace traffic architecture "Cloud Traffic Control", followed by the testing of model streams of vehicles in the areas of road infrastructure.

The research findings and results obtained in the course of the research are reliable, which is confirmed by the experiments carried out, testing and verification of the proposed models and methods of traffic flow monitoring and control.

The practical significance of scientific research is confirmed by a significant decrease in the execution of traffic routes when modelling real processes on fragments of road infrastructure models. The results of the dissertation as a set of models, methods and models of cyber-physical infrastructure were introduced into the educational process in Kharkov National University of Radio Electronics (act on implementation from 09.09.2017) in disciplines «Discrete Mathematics», «Cloud-Fog Cyber Physical Systems». The developed architecture of an intelligent cloud

traffic light, as well as a transport computation model, can be implemented as a component of the project in the synthesis of cloud services (Certificate, 28 September 2017, LLC "The First Institute of Reliable Software").

The results of the dissertation work are reflected in 18 publications: 7 articles, among which 3 are in the scientific journals, which are included in the "Lists of scientific professional editions of Ukraine" (3 of them are in international science-computer databases), 3 articles in international scientific journals abroad (from them 1 – in the international science-based base Scopus, 1 – in the international science-based base ORCID); as well as 11 publications in international scientific conferences (7 of them – abroad, 6 are part of Scopus and IEEEExplore science-based databases). The applicant has 7 publications included in the Scopus Science Centre and has the Hirsch index $h = 1$.

List of publications

List of publications, in which the main scientific results of the thesis have been published:

1. Немченко В.П. Моделирование сетевых протоколов при построении тестовых последовательностей / В.П. Немченко, А.Н. Зиарманд, Ю.А. Чепелев // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2011. – №5(90). – С. 18-21. (Індексується GoogleScholar, Реферативна база “Наукова періодика України”, РИНЦ (eLibrary), National Library of Ukraine named after Vernadsky).

2. Немченко В.П. Использование энергосберегающих технологий в современных сетях / В.П. Немченко, А.Н. Зиарманд, А.С. Изотов // Вестник Херсонского национального технического университета. – 2012. – №1(44). – С. 146-148. (Індексується РИНЦ (eLibrary), Google Scholar, National Library of Ukraine named after Vernadsky).

3. Зиарманд А.Н. Модели и методы мониторинга и управления транспортом / А.Н. Зиарманд, В.И. Хаханов // Радиоэлектроника и информатика. – 2016. – №3(74). – С. 71-87. (Входить до міжнародних

наукометричних баз Index Copernicus, Google Scholar, OECSP, OAJI, Scholar Steer, SIS, Cyberleninka, CiteFactor, TIU Hannover, I2OR).

4. Vladimir Hahanov. Internet-driven Cyber Control of Traffic / Vladimir Hahanov, Wajeb Gharibi, Svetlana Chumachenko, Evgeniya Litvinova, Vladimir Miz, *Arthur Ziarmant* // Australian Journal of Scientific Research. – 2014. – Volume IV, No.1(5). – Pp. 217-224.

5. Hahanov V. Cloud-Driven Traffic Monitoring and Control Based on Smart Virtual Infrastructure / V. Hahanov, Wajeb Gharibi, E. Litvinova, S. Chumachenko, A. *Ziarmant*, I. Englesi, I. Gritsuk, V. Volkov, A. Khakhanova // SAE Technical Paper. USA. 2017-01-0092, 2017. 6 p. doi:10.4271/2017-01-0092. (Входить до міжнародної наукометричної бази Scopus, ORCID).

6. *Зиарманд А.Н.* Модель облачного сервіса для пошуку оптимального пути / С. В. Чумаченко, Е. И. Литвинова, В. И. Хаханов, *А. Н. Зиарманд* // Paradigmata poznání. Vědecko vydavatelské centrum «Sociosféra-CZ», s.r.o., Praha, Česká republika. – 2017. – Вып. 3. – С.63-83. doi: 10.24045/pp.2017.3.6 (The journal is indexed by Electronic Research Library, Russia; Research Bible, China; Scientific Indexing Services, USA; Cite Factor, Canada; General Impact Factor, India; Scientific Journal Impact Factor, India; CrossRef, USA; ORCID, USA).

7. *Зиарманд А.Н.* Теоретическая суть проекта «Smart Roads» / *А.Н. Зиарманд* // Автоматизированные системы управления и приборы автоматики. – 2013. – Вып. 162. – С. 28-34. (Входить до міжнародних наукометричних баз Google Scholar, Cyberleninka). <http://cyberleninka.ru/article/n/teoreticheskaya-sut-proekta-smart-roads>

Results, which confirm approbation of the thesis:

8. Hahanov V. I. Cloud traffic control system / V. I. Hahanov, O. A. Gus, A. *Ziarmant*, Ngene Christopher Umerah, A. Arefjev // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 72-76. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

9. *Ziarmand A. Smart road infrastructure / A. Ziarmand // Proc. Of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 430-434. (Work in Progress). (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

10. *Ziarmand A. Cloud Service for Traffic Control / Artur Ziarmand, Vladimir Hahanov, Volodymyr Miz, Anastasya Hahanova, Aleksey Priymak // Proc. of the XII International IEEE Conference “Modern Problems of Radio Engineering, Telecommunications, and Computer Science”. 2014. – Lviv-Slavske, Ukraine. – P. 557-559. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

11. *Hahanov V. Cyber physical system – smart cloud traffic control / V. Hahanov, Wajeb Gharibi, L.S. Abramova, S. Chumachenko, E. Litvinova, A. Hahanova, V. Rustinov, V. Miz, A. Zhalilo, A. Ziarmand // Proc. of the IEEE Design & Test Symposium. – 2014. – Kiev, Ukraine. – P. 49-66. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

12. *Ziarmand A. Transport monitoring and control systems / A. Ziarmand, D. Kucherenko, T. Soklakova // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 474-477. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).*

13. *Зиарманд А.Н. Формальная модель киберсистемы / А.Н. Зиарманд, И.И. Чугуров // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2014. – С. 65.*

14. *Зиарманд А.Н. Формальна модель хмарного управління транспортом / А.Н. Зиарманд, К.Э. Мороз // Материалы XIX Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2015. – С. 4.*

15. *Зиарманд А.Н. I-Cloud Traffic Control / А.Н. Зиарманд, В.А. Мизь // 6-я Международная студенческая конференция и конкурс научных работ по вопросам информационной безопасности «CyberSecurity for the Next Generation». – 2014. – “Kaspersky Office”, Москва, РФ. – P.13.*

Publications that additionally reflect the scientific results of the thesis:

16. Chumachenko S. Quantum data structures for SoC design / S. Chumachenko, A. Shkil, A. Hahanova, A. Ziarmand, A. Pryimak // Proc. of the 13th International Conference “IEEE Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”. – 2015. Polyana-Lviv. – Ukraine. – P. 355-357. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

17. Soklakova T. Big data visualization in smart cyber university / T. Soklakova, A. Ziarmand, S. Osadchyieva // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 469-473. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

18. *Зиарманд А.Н.* Анализ компетентности пользователя компьютерной системы как диагностический эксперимент / Зиарманд А.Н., Кучеренко Д.Е. // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2011. – С. 40-41.

Key words: Computing, Cyberphysical System, Smart Cloud Traffic Control, Metrics, Quality Criteria, Models, Methods, Software Components, Modified Dijkstra Algorithm, smart architecture, transport computing model.

АННОТАЦИЯ

Зиарманд Артур Нисарович. Квалификационная научная работа на правах рукописи – Диссертация на соискание ученой степени кандидата технических наук (доктора философии) по специальности 05.13.05 «компьютерные системы и компоненты». – Харьковский национальный университет радиоэлектроники, Министерство образования и науки Украины, Харьков, 2017.

Цель исследования – повышение качества и безопасности дорожного движения за счет создания киберфизической модели компьютерингового online взаимодействия водителя с облачными сервисами управления автомобилем на основе цифрового мониторинга дорожной инфраструктуры и транспортных потоков, при использовании умных сенсоров, средств телекоммуникации и навигации.

Задачи исследования:

1) Разработка модели транспортного компьютеринга – киберфизического взаимодействия автомобиля с облачным сервисом посредством эволюционного перемещения светофора из физического в виртуальное пространство для цифрового мониторинга транспортных потоков и квазиоптимального управления дорожным движением.

2) Создание архитектуры умного облачного светофора на основе использования логических операций, дающей возможность существенно увеличить пропускную способность перекрестка дорог.

3) Создание метрики и критериев оценивания качества инфраструктуры для online анализа киберфизического пространства, поиска квази-оптимального маршрута и уменьшения времени его исполнения.

4) Разработка метода анализа киберфизической инфраструктуры дорожного движения на основе алгоритма Дейкстры для online поиска квазиоптимального маршрута в условиях возникновения коллизий.

5) Практическая реализация моделей и методов цифрового мониторинга и облачного управления автомобилем в рамках создания

киберфизической системы дорожного движения и последующее их тестирование.

Научная новизна:

1) Впервые предложена модель транспортного компьютеринга, которая характеризуется киберфизическим взаимодействием автомобиля с облачным сервисом посредством эволюционного перемещения светофора из физического в виртуальное пространство для цифрового мониторинга транспортных потоков и квазиоптимального управления дорожным движением.

2) Впервые предложена архитектура умного облачного светофора, которая характеризуется использованием логических операций и временем простоя зеленого сигнала, что дает возможность существенно увеличить пропускную способность транспортных потоков на перекрестке дорог.

3) Усовершенствована метрика и критерии оценивания качества инфраструктуры, которая отличается возможностью online анализа киберфизического пространства для поиска квази-оптимального маршрута и уменьшения времени его исполнения.

4) Усовершенствован алгоритм Дейкстры, который отличается возможностью анализа киберфизической инфраструктуры дорожного движения для online поиска квазиоптимального маршрута транспортного средства в условиях возникновения коллизий.

Практическая реализация моделей и методов цифрового мониторинга и облачного управления автомобилем выполнена в рамках создания и верификации программных компонентов киберфизической архитектуры дорожного движения «Cloud Traffic Control» с последующим тестированием модельных потоков транспортных средств на участках дорожной инфраструктуры.

Полученные в процессе выполнения исследования научные выводы и результаты являются достоверными, что подтверждается проведенными

экспериментами, тестированием и верификацией предложенных моделей и методов мониторинга и управления дорожным движением.

Практическая значимость научных исследований подтверждается существенным уменьшением исполнения маршрутов движения при моделировании реальных процессов на фрагментах моделей дорожной инфраструктуры. Результаты диссертации в составе моделей, методов и архитектуры киберфизической инфраструктуры внедрены в учебный процесс Харьковского национального университета радиоэлектроники (акт о внедрении от 03.09.2017) при чтении курсов: «Дискретна математика», «Cloud-Fog киберфизические системы». Разработанная архитектура умного облачного светофора, а также модель транспортного компьютеринга могут быть реализованы как компонент проекта при синтезе облачного сервиса (Справка от 28.09.2017, ООО «Первый институт надежного программного обеспечения»).

Результаты диссертации отражены в 18 печатных работах: 7 статей, среди которых 3 – в научных журналах, включенных в «Перечень научных специализированных изданий Украины» (из них 3 – в международных наукометрических базах), 3 статьи – в международных научных журналах за рубежом (из них 1 – в международной наукометрической базе Scopus, 1 – в международной наукометрической базе ORCID); а также 11 публикаций в международных научных конференциях (из них 7 – за рубежом, 6 входят в наукометрическую базу Scopus). Соискатель имеет 7 публикаций в наукометрической базе Scopus и индекс Хирша $h=1$.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

Список публикаций соискателя, в которых опубликованы основные научные результаты диссертации:

1. Немченко В.П. Моделирование сетевых протоколов при построении тестовых последовательностей / В.П. Немченко, А.Н. Зиарманд, Ю.А. Чепелев // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2011. – №5(90). – С. 18-21. (Індексується

GoogleScholar, Реферативна база “Наукова періодика України”, РИНЦ (eLibrary), National Library of Ukraine named after Vernadsky).

2. Немченко В.П. Использование энергосберегающих технологий в современных сетях / В.П. Немченко, А.Н. Зиарманд, А.С. Изотов // Вестник Херсонского национального технического университета. – 2012. – №1(44). – С. 146-148. (Индексується РИНЦ (eLibrary), Google Scholar, National Library of Ukraine named after Vernadsky).

3. *Зиарманд А.Н.* Модели и методы мониторинга и управления транспортом / А.Н. Зиарманд, В.И. Хаханов // Радиоэлектроника и информатика. – 2016. – №3(74). – С. 71-87. (Входить до міжнародних наукометричних баз Index Copernicus, Google Scholar, OECSP, OAJI, Scholar Steer, SIS, Cyberleninka, CiteFactor, TIU Hannover, I2OR).

4. Vladimir Hahanov. Internet-driven Cyber Control of Traffic / Vladimir Hahanov, Wajeb Gharibi, Svetlana Chumachenko, Evgeniya Litvinova, Vladimir Miz, *Arthur Ziarmand* // Australian Journal of Scientific Research. – 2014. – Volume IV, No.1(5). – Pp. 217-224.

5. Hahanov V. Cloud-Driven Traffic Monitoring and Control Based on Smart Virtual Infrastructure / V. Hahanov, Wajeb Gharibi, E. Litvinova, S. Chumachenko, A. *Ziarmand*, I. Englesi, I. Gritsuk, V. Volkov, A. Khakhanova // SAE Technical Paper. USA. 2017-01-0092, 2017. 6 p. doi:10.4271/2017-01-0092. (Входить до міжнародної наукометричної бази Scopus, ORCID).

6. *Зиарманд А.Н.* Модель облачного сервиса для поиска оптимального пути / С. В. Чумаченко, Е. И. Литвинова, В. И. Хаханов, А. Н. *Зиарманд* // Paradigmata poznání. Vědecko vydavatelské centrum «Sociosféra-CZ», s.r.o., Praha, Česká republika. – 2017. – Вып. 3. – С.63-83. doi: 10.24045/pp.2017.3.6 (The journal is indexed by Electronic Research Library, Russia; Research Bible, China; Scientific Indexing Services, USA; Cite Factor, Canada; General Impact Factor, India; Scientific Journal Impact Factor, India; CrossRef, USA; ORCID, USA).

7. *Ziarmand A.H.* Теоретическая суть проекта «Smart Roads» / *A.H. Ziarmand* // Автоматизированные системы управления и приборы автоматики. – 2013. – Вып. 162. – С. 28-34. (Входит до міжнародних наукометричних баз Google Scholar, Cyberleninka). <http://cyberleninka.ru/article/n/teoreticheskaya-sut-proekta-smart-roads>

Результаты, которые подтверждают апробацию материалов диссертации:

8. Hahanov V. I. Cloud traffic control system / V. I. Hahanov, O. A. Gus, A. Ziarmand, Ngene Christopher Umerah, A. Arefjev // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 72-76. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

9. *Ziarmand A.* Smart road infrastructure / *A. Ziarmand* // Proc. Of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 430-434. (Work in Progress). (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

10. *Ziarmand A.* Cloud Service for Traffic Control / *Artur Ziarmand, Vladimir Hahanov, Volodymyr Miz, Anastasya Hahanova, Aleksey Priymak* // Proc. of the XII International IEEE Conference “Modern Problems of Radio Engineering, Telecommunications, and Computer Science”. – 2014. – Lviv-Slavske, Ukraine. – P. 557-559. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

11. Hahanov V. Cyber physical system – smart cloud traffic control / V. Hahanov, Wajeb Gharibi, L.S. Abramova, S. Chumachenko, E. Litvinova, A. Hahanova, V. Rustinov, V. Miz, A. Zhalilo, A. Ziarmand // Proc. of the IEEE Design & Test Symposium. – 2014. – Kiev, Ukraine. – P. 49-66. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

12. *Ziarmand A.* Transport monitoring and control systems / *A. Ziarmand, D. Kucherenko, T. Soklakova* // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 474-477. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

13. *Зиарманд А.Н.* Формальная модель киберсистемы / А.Н. Зиарманд, И.И. Чугуров // *Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке».* – Харьков, Украина. – 2014. – С. 65.

14. *Зиарманд А.Н.* Формальная модель умного управления транспортом / А.Н. Зиарманд, К.Э. Мороз // *Материалы XIX Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке».* – Харьков, Украина. – 2015. – С. 4.

15. *Зиарманд А.Н.* I-Cloud Traffic Control / А.Н. Зиарманд, В.А. Мизь // *6-я Международная студенческая конференция и конкурс научных работ по вопросам информационной безопасности «CyberSecurity for the Next Generation».* – 2014. – “Kaspersky Office”, Москва, РФ. – С. 13.

Публикации, которые дополнительно отражают научные результаты диссертации:

16. Chumachenko S. Quantum data structures for SoC design / S. Chumachenko, A. Shkil, A. Hahanova, A. Ziarmand, A. Pryimak // *Proc. of the 13th International Conference “IEEE Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”.* – 2015. Polyana-Lviv. – Ukraine. – P. 355-357. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

17. Soklakova T. Big data visualization in smart cyber university / T. Soklakova, A. Ziarmand, S. Osadchyieva // *Proc. of the IEEE East-West Design & Test Symposium (EWDTS).* – Yerevan, Armenia. – 2016. – P. 469-473. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

18. *Зиарманд А.Н.* Анализ компетентности пользователя компьютерной системы как диагностический эксперимент / Зиарманд А.Н., Кучеренко Д.Е. // *Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке».* – Харьков, Украина. – 2011. – С. 40-41.

Ключевые слова: компьютеринг, киберфизическая система, умное облачное управление транспортом, метрика, критерии качества, модели, методы, программные компоненты, модифицированный алгоритм Дейкстры, умная архитектура, модель транспортного компьютеринга.

ЗМІСТ

ВСТУП	21
1 МОДЕЛІ І МЕТОДИ МОНІТОРИНГУ І УПРАВЛІННЯ ТРАНСПОРТОМ	30
1.1 Хмарний комп'ютинг і системи управління транспортом.....	30
1.2 Моделі міських інфраструктур дорожнього руху	37
1.3 Системи позиціонування та навігації транспортних засобів.....	42
1.4 Електронне обладнання сучасного автомобіля.....	49
1.5 Кіберфізичні системи хмарного управління транспортом	52
1.6 Постановка мети і завдань дослідження.....	60
2 АНАЛІЗ ЯКОСТІ ТОПОЛОГІЧНИХ СТРУКТУР ДОРОЖНЬОГО РУХУ.	64
2.1 Критерії якості інфраструктури дорожнього руху	64
2.2 Функції прохідності дорожньої інфраструктури	70
2.3 Метрика e-інфраструктури дорожнього руху	75
2.4 Топології зв'язків компонентів системи	80
2.5 Спрощений критерій якості топології	84
2.6 Приведена метрика топології графа.....	90
2.7 Висновки до розділу 2	97
3 МОДИФІКАЦІЯ АЛГОРИТМУ ДЕЙКСТРИ ДЛЯ ПОШУКУ МАРШРУТУ	99
3.1 Вступ	99
3.2 Метод пошуку найкоротших шляхів у дорожній інфраструктурі.....	101
3.3 Метод пошуку найкоротших відстаней між парами вершин.....	108
3.4 Пошук мінімальних шляхів для графа з одиничними дугами	112
3.5 Визначення мінімальних шляхів в трикутній топології	119
3.6 Пошук мінімальних шляхів в прямокутній топології	127
3.7 Висновки до розділу 3	135
4 КІБЕРФІЗИЧНА СТРУКТУРА ХМАРНОГО ТРАНСПОРТНОГО КОМП'ЮТИНГУ	137
4.1 Інноваційна привабливість "Cloud Traffic Control"	138
4.2 Структура сервісів хмарного управління транспортом	146
4.3 Розумний хмарний світлофор.....	150
4.4 Структурно-аналітична модель хмарного управління.....	157
4.5 Модельні фізичні експерименти на road-інфраструктурі	169
4.6 Висновки до розділу 4	175
ВИСНОВОК.....	178
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	180
ДОДАТОК А. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ	198
ДОДАТОК Б. АПРОБАЦІЯ РЕЗУЛЬТАТІВ	201
ДОДАТОК В. ДОКУМЕНТИ, ЩО ПІДТВЕРДЖУЮТЬ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ.....	203
ДОДАТОК Г. ПЕРЕВАГИ ХМАРНИХ СЕРВІСІВ КІБЕРСИСТЕМИ	206
ДОДАТОК Д. ЛІСТИНГИ ПРОГРАМ.....	242

ВСТУП

Актуальність дослідження. Кіберфізична інфраструктура дорожнього руху і транспортний засіб створюють бінарну систему активної взаємодії для задоволення потреб людства. Вони неявно формулюють дві актуальних доктрини, здатні радикально змінити існуючий світ: 1) Світлофорів немає у водній стихії і у повітряному просторі, значить, їх не повинно бути на дорогах. Прибрати всі знаки з дорожньої інфраструктури, означає створити зелену планету і чисті міста. 2) Зробити автомобіль розумним, з автономним управлінням, означає звести до нуля тисячі щоденних аварій на планеті. Щоб реалізувати дані доктрини необхідно всього лише економічно обгрунтовано вирішити такі технологічні ринкові проблеми: 1) Довести глибину позиціонування фізичного об'єкта в оцифрованому просторі до 10 сантиметрів. 2) На кілька порядків підвищити пропускну здатність бездротових телекомунікаційних каналів зв'язку для підтримки е-інфраструктури. 3) Створити надійні масштабовані комп'ютерні сервіси цифрового моніторингу і хмарного управління трафіком. 4) Змусити населення планети повірити в інноваційну кіберкультуру управління транспортом. Наслідки від реалізації двох доктрин – трильйони доларів чистої економії, нова якість життя громадян і зелена планета. Ось кілька очевидних трансформацій у фізичному світі, що безпосередньо призводять до соціально-екологічних і економічних результатів: 1) Прибрати світлофори і дорожні знаки та автомобільні номери означає зберегти сотні тисяч тонн металу, сотні тисяч кіловат-годин електроенергії і мільярди доларів на їх установку і експлуатацію.

Запропоноване дослідження розглядає проблему створення надійних масштабованих комп'ютерних сервісів цифрового моніторингу і хмарного управління трафіком, яка інтегрує два найактуальніших і популярних наукових напрямки: транспорт і комп'ютер для отримання якісно нових умов життя в розумному автомобілі, використовує кіберфізичну інфраструктуру дорожнього руху.

Виняткова активність в області транспортного комп'ютингу світових лідерів з академічних кіл (MIT, Berkeley, Stanford, Tokio, Harvard) та автомобільно-комп'ютирової індустрії (General Motors, Mitsubishi, Mercedes, BMW, IBM, NASA, Google, Synopsys, Microsoft) стимулюють наукові розробки наступних активних вчених: Aleksander Sladkowski, Wiesław Pamuła, Ling Pang, Danda Rawat, Dimitris Charitos, Jeffrey Adler, John France, Huang Guibin, Luke Mirowski, David Wallom, Durmus Cesur, Masaya Yoshikawa, Jeff Massimilla, David Kleinbaum, Dustdar Schahram, Nick Antonopoulos, Petrovskiy AV, Горбачов П.Ф., Абрамова Л., Ларін О.М., Харченко В.С., Igor Gritsuk, Vladimir Volkov.

Зв'язок роботи з науковими програмами, держбюджетними темами.

Розробка теми дисертації здійснювалася відповідно до планів держбюджетних НДР і міжнародних договорів, виконуваних на кафедрі АПОТ Харківського національного університету радіоелектроніки у період з 2011 року, в тому числі: 1) Договір про дружбу і співробітництво між ХНУРЕ та компанією «Aldec Inc.» (USA) № 04 від 01.11.2011; 2) Фундаментальна держбюджетна НДР «Мультипроцесорна система пошуку, розпізнавання та прийняття рішень для інформаційної комп'ютерної екосистеми», № 269 (2011-2013), №ДР 0111U002956; 3) Фундаментальна держбюджетна НДР «Персональний віртуальний кіберкомп'ютер та інфраструктура аналізу кіберпростору», №258 (2012-2014); 4) Curricula Development for New Specialization: Master of Engineering in Microsystems Design 530785-TEMPUS-1-2012-1-PL-TEMPUS-JPCR MastMST (2012-2016). 6) Фундаментальна держбюджетна НДР №ДР 0115U-000712 "Cyber Physical System – Smart Cloud Traffic Control" (2015-2017).

Автор дисертаційної роботи брав участь при виконанні зазначених договорів і програм, як виконавець, розробник і програміст мікросервісів кіберфізичної інфраструктури дорожнього руху. Автор також брав участь у C++-кодуванні програмних модулів хмарного управління трафіком і оптимізації виконання маршрутів.

Науково-практична задача дослідження: створення кіберфізичних моделей комп'ютингових сервісів для цифрового моніторингу транспортних потоків і хмарного керування автомобілями з метою істотного зменшення часу виконання замовлених маршрутів.

Сутність дослідження – створення компонентів кіберфізичної інфраструктури безпечного дорожнього руху за рахунок поступового перенесення дорожніх знаків, світлофорів у кіберпростір і точного позиціонування транспорту з метою хмарного online керування автомобілем на основі розумної суперпозиції замовленого маршруту з оперативним цифровим моніторингом дорожньої обстановки, яка відображається на сенсорному дисплеї автомобіля. На рис. 1 представлена тріада транспортного комп'ютингу (cloud-car-streetlight), яка еволюційно перетворюється в cloud-car computing шляхом перенесення керуючих знаків з фізичної інфраструктури до кібер-хмари для їх відображення на моніторі автомобільного комп'ютера.

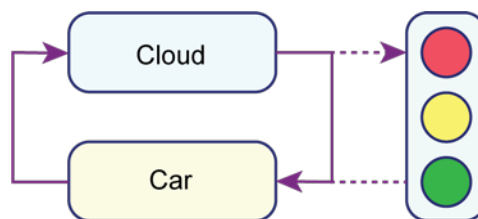


Рис. 1 – Транспортний комп'ютинг

Ринкова привабливість дослідження. Впровадження хмарного сервісу управління транспортом призведе до збереження екології планети і до зменшення: 1) часу проходження замовлених маршрутів; 2) споживання енергетичних ресурсів і матеріальних витрат на створення і експлуатацію світлофорів, дорожніх знаків, автомобільних номерів; 3) числа аварій і крадіжок автомобілів.

Об'єкт дослідження – процеси цифрового моніторингу та кібер управління розумним автомобілем у масштабованій дорожній кіберфізичній

інфраструктурі, на якій позиціонуються маршрути руху всіх транспортних засобів при виконанні замовлених маршрутів.

Предмет дослідження – кіберфізична система комп'ютирової взаємодії хмарних сервісів управління транспортом та інфраструктурою з розумними засобами цифрового моніторингу дорожньої обстановки з метою online оптимального управління автомобілем для виконання замовленого маршруту.

Методи дослідження – топології інфраструктур дорожнього руху, булева алгебра, теорія множин, теорія графів, теорія цифрових автоматів і оптимізації шляхів на графах, паралельні методи хмарних обчислень і структури даних; кіберфізичний комп'ютинг, теорія алгоритмів, методи, засоби, мови проектування і моделювання; методи і критерії якості оцінювання топологічних структур; засоби синтезу та аналізу програмно-апаратної інфраструктури хмарних сервісів.

Мета дослідження – підвищення якості та безпеки дорожнього руху за рахунок створення кіберфізичної моделі комп'ютирової online взаємодії водія з хмарними сервісами керування автомобілем на основі цифрового моніторингу дорожньої інфраструктури і транспортних потоків, при використанні розумних сенсорів, засобів телекомунікації та навігації.

Задачі дослідження:

1) Розробка моделі транспортного комп'ютингу – кіберфізичної взаємодії автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

2) Створення архітектури розумного хмарного світлофора на основі використання логічних операцій, що дає можливість істотно збільшити пропускну здатність перехрестя доріг.

3) Створення метрики і критеріїв оцінювання якості інфраструктури для online аналізу кіберфізичного простору, пошуку квазі-оптимального маршруту і зменшення часу його виконання.

4) Розробка методу аналізу кіберфізичної інфраструктури дорожнього руху на основі алгоритму Дейкстри для online пошуку квазіоптимального маршруту в умовах виникнення колізій.

5) Практична реалізація моделей і методів цифрового моніторингу і хмарного керування автомобілем в рамках створення кіберфізичної системи дорожнього руху та подальше їх тестування.

Наукова новизна:

1) Вперше запропоновано модель транспортного комп'ютингу, яка характеризується кіберфізичною взаємодією автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

2) Вперше запропоновано архітектуру розумного хмарного світлофора, яка характеризується використанням логічних операцій і часом простою зеленого сигналу, що дає можливість істотно збільшити пропускну здатність транспортних потоків на перехресті доріг.

3) Удосконалено метрику і критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання.

4) Удосконалено алгоритм Дейкстри, який відрізняється можливістю аналізу кіберфізичної інфраструктури дорожнього руху для online пошуку квазіоптимального маршруту транспортного засобу в умовах виникнення колізій.

5) Практична реалізація моделей і методів цифрового моніторингу і хмарного керування автомобілем виконана в рамках створення і верифікації програмних компонентів кіберфізичної архітектури дорожнього руху «Cloud

Traffic Control» з подальшим тестуванням модельних потоків транспортних засобів на ділянках дорожньої інфраструктури.

Отримані в процесі виконання дослідження наукові висновки і результати є достовірними, що підтверджується проведеними експериментами, тестуванням і верифікацією запропонованих моделей і методів моніторингу та управління дорожнім рухом. Практична значимість наукових досліджень підтверджується істотним зменшенням виконання маршрутів руху при моделюванні реальних процесів на фрагментах моделей дорожньої інфраструктури. Результати дисертації в складі моделей, методів і архітектури кіберфізичної інфраструктури впроваджені в навчальний процес Харківського національного університету радіоелектроніки (акт про впровадження від 03.09.2017) при читанні курсів «Дискретна математика», «Cloud-Fog кіберфізичні системи». Розроблена архітектура розумного хмарного світлофора, а також модель транспортного комп'ютингу можуть бути реалізовані як компонент проекту при синтезі хмарного сервісу (Довідка від 28.09.2017, ТОВ «Перший інститут надійного програмного забезпечення»).

Особистий внесок здобувача. Всі наукові і практичні результати отримані автором особисто. У роботах, опублікованих зі співавторами (Додаток А), здобувачеві належать:

[1] – моделі локальних комп'ютерних мереж для аналізу графових топологій з метою оптимізації обміну даними між обчислювачами на основі використання мережевих протоколів;

[2] – модель організації обміну даними в локальній обчислювальній мережі, яка зменшує енерговитрати за рахунок активізації необхідних логічних шляхів на тлі сплячих пристроїв;

[3] – модель транспортного комп'ютингу для кіберфізического взаємодії автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом;

[4] – метрика і критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання;

[5] – модель кіберфізичної взаємодії транспорту з хмарним сервісом для цифрового моніторингу транспортних потоків і управління дорожнім рухом на основі хмарних світлофорів, а також експерименти на дорожній інфраструктурі;

[6] – вдосконалений алгоритм Дейкстри, який відрізняється можливістю аналізу кіберфізичної інфраструктури дорожнього руху для online пошуку квазіоптимального маршруту транспортного засобу в умовах виникнення колізій; архітектура розумного хмарного світлофора, яка характеризується використанням логічних операцій і часом простою зеленого сигналу, що дає можливість істотно збільшити пропускну здатність транспортних потоків на перехресті доріг;

[7] – моделі та критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання;

[8] – моделі і методи цифрового моніторингу і хмарного керування автомобілем в рамках кіберфізичної архітектури дорожнього руху «Cloud Traffic Control»;

[9] – розумна інфраструктура дорожнього руху на основі використання хмарних інтелектуальних світлофорів;

[10] – моделі та способи реалізації хмарних сервісів моніторингу та управління транспортними потоками на основі платформи Google;

[11] – структура кіберфізичної системи хмарного керування транспортними потоками на основі використання супутникової системи навігації, електронних цифрових карт, смартфонів і цифрової ідентифікації автомобілів;

[12] – моделі транспортного моніторингу і цифрового керування автомобілем і розумним світлофором;

[13] – формальна комп'ютерингова модель кіберсистеми для моніторингу та управління кіберфізичними процесами і явищами;

[14] – формальна модель хмарного управління транспортом на основі використання розумних сенсорів і віртуальних світлофорів;

[15] – кіберфізична архітектура хмарного управління транспортом на основі цифрової ідентифікації автомобілів;

[16] – кубітні структури даних для паралельного аналізу графових топологій транспортних потоків;

[17] – обчислювальні архітектури для аналізу великих даних для управління транспортними потоками;

[18] – метрика вимірювання процесів і явищ для моніторингу та управління транспортними потоками.

Апробація результатів дисертації. Результати роботи були представлені та обговорені на наступних конференціях: IEEE East-West Design and Test Symposium 2013 (Russia), 2014 (Ukraine), 2015 (Georgia), 2016 (Armenia), 2017 (Serbia); Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті» 2011, 2014, 2015 рр., Харків, Україна; XII International IEEE Conference "Modern Problems of Radio Engineering, Telecommunications, and Computer Science", 2014, Lviv-Slavske, Ukraine; 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015, Lvov, Ukraine; Міжнародна студентська конференція і конкурс наукових робіт з питань інформаційної безпеки «CyberSecurity for the Next Generation», 20-23.02.2014, "Kaspersky Office", Москва, Росія.

Автор також брав участь у інноваційних проектах та розробках, презентації їх на виставках з отриманням призових місць:

1. Хаханов В.И. Облачное цифровое управление дорожным движением (Smart Cloud Traffic Control) / В.И. Хаханов, С. В. Чумаченко, Е. И. Литвинова, А.Н. Зуарманд, В.А. Мизь, О.И. Филипенко // Каталог инновационных проектов и разработок ХНУРЭ. 2013. <http://nure.ua/university/structure/science/innovacionnye-proekty-i-razrabotki/1-2-oblachnoe-cifrovое-upravlenie-dorozhnym-dvizheniem-smart-cloud-traffic-control/>

2. Хаханов В.И. Создание системы «Умные дороги» для организации инфраструктуры дорожного движения (Smart Roads Infrastructure) / В.И. Хаханов, С.В. Чумаченко, Е.И. Литвинова, *А.Н. Зиарманд*, В.А. Мизь, И.И. Чугуров, О.И. Филипенко // Каталог инновационных проектов и разработок ХНУРЭ. 2013. <http://nure.ua/university/structure/science/innovacionnye-proekty-i-razrabotki/1-3-sozdanie-sistemy-umnye-dorogi-dlya-organizacii-infrastruktury-dorozhnogo-dvizheniya-smart-roads-infrastructure/>

3. Хаханов В.И. Использование квантовых вычислений для поиска кратчайшего пути (Q-ROUTE) / В.И. Хаханов, С.В. Чумаченко, Е.И. Литвинова, *А.Н. Зиарманд*, В.А. Мизь, О.И. Филипенко // Каталог инновационных проектов и разработок ХНУРЭ. 2013. <http://nure.ua/university/structure/science/innovacionnye-proekty-i-razrabotki/1-4ispolzovanie-kvantovykh-vychislenij-dlya-poiska-kratchajshogo-puti-q-route/>

4. *Зиарманд А.Н.* I-Cloud Traffic Control / А.Н. Зиарманд, В.А. Мизь // Международный конкурс студенческих научных работ по вопросам информационной безопасности «CyberSecurity for the Next Generation». – 2014. – “Kaspersky Office”, Москва, РФ. Диплом за кращу презентацію.

5. *Зиарманд А.Н.*, Хаханов В.И. Создание инфраструктуры умных дорог // Публикация в журнале международного форума «Kharkiv-IT» (Харьков, Пятихатки). 2013. Диплом за 3 місце в номінації та золота медаль.

Результати дисертаційної роботи відображено у 18 друкованих працях: 7 статей, серед яких 3 у наукових журналах, що входять до «Переліків наукових фахових видань України» (з них 3 – у міжнародних наукометричних базах), 3 статті – в міжнародних наукових журналах за кордоном (з них 1 – в міжнародній наукометричній базі Scopus, 1 – у міжнародній наукометричній базі ORCID); а також 11 публікацій у міжнародних наукових конференціях (з них 7 – за кордоном, 6 входять до наукометричної бази Scopus). Здобувач має 7 публікацій, що входять до наукометричної бази Scopus, та має індекс Хірша $h=1$.

Структура дисертації представлена 269 сторінками (з них 179 сторінок основного тексту) і містить: 4 розділи, 61 рисунок, 25 таблиць, список джерел з 159 назв (на 19 с.), 5 додатків (на 71 с.).

1 МОДЕЛІ І МЕТОДИ МОНІТОРИНГУ ТА УПРАВЛІННЯ ТРАНСПОРТОМ

Пропонуються основні напрямки сталого розвитку кіберфізичного транспортного компютингу, пов'язані із засобами телекомунікацій, точного позиціонування, online e-картографії, диференціальної навігації, Internet of Car (e-інфраструктура автомобіля, доріг і хмарних сервісів). Представлені компоненти кіберфізической системы моніторингу та управління дорожнім рухом в реальному часу, яка ґрунтується на використанні глобальних систем позиціонування і навігації (GPS, GPRS), інтелектуальних дорожніх контролерів, засобів радіочастотної ідентифікації автомобілів та інфраструктури дорожнього руху.

1.1 Хмарний комп'ютинг і системи управління транспортом

У сучасному світі розвивається e-інфраструктура дорожнього руху в рамках порівняно нового поняття Internet of Car. В [1, 2, 3, 4, 5, 6, 7] представлена інформація про використання існуючих і майбутніх e-інфраструктур для поліпшення якості життя людей. Розробка e-інфраструктури для організації та проведення наукових досліджень є одним з ключових напрямків на ринку електронних технологій. Одним із прикладів застосування e-інфраструктури є науково-освітні заходи, які ґрунтуються на використанні хмарних або web-сервісів в режимі online. Крім того, e-інфраструктура є необхідною умовою в даний час для функціонування державних і соціальних служб, для створення розумних урядів, міст, будинків, а також моніторингу та управління транспортними потоками в режимі online. Крім того, електронна інфраструктура дає можливість створювати тимчасові творчі колективи, що включають фахівців з різних сфер діяльності, в тому числі, і з інших країн для вирішення складних науково-технічних проблем, що припускають наявність хмарних сервісів, а також розумних речей для збору, зберігання, аналізу великих даних, реалізації сервісів моделювання на основі

використання сучасних спеціалізованих логічних процесорів [8]. Одним із прикладів, що підтверджують значимість е-інфраструктури, є створення інституції, яка називається The European Strategy Forum on Research Infrastructures (ESFRI). Дана організація координує зусилля європейських університетів і компаній для створення стандартів і прототипів е-інфраструктур, спрямованих на забезпечення ефективного і квазі-оптимального функціонування кіберфізичних сервісів, пов'язаних з проблемами моніторингу та управління у всіх сферах людської діяльності. Сучасне уявлення е-інфраструктури в Європі включає мережу обчислювальних і інформаційних дата центрів, хмарні сервіси, гаджети аутентифікації, авторизації та інтерфейсні зв'язки з хмарами [9, 10]. Практика останніх 10 років свідчить про затребуваність е-інфраструктури і хмарних сервісів для більшої частини населення планети в режимі online 24/7 [11, 12, 13, 14, 15]. Хмарні сервіси [16] забезпечують легкий масштабований доступ до додатків, ресурсів і послуг, які повністю керуються провайдером хмарних сервісів. Властивість динамічного масштабування займає ключове значення для задоволення потреб користувачів і замовників [17], яке так само дозволяє істотно скоротити витрати на розгортання власних ресурсів і виділення персоналу для управління службою. У сучасному мобільному світі з високою пропускнуою здатністю і повсюдним доступом до мережі, користувачі можуть вільно співпрацювати через кордони, таким чином, у створюваному середовищі повинні бути чітко описані межі області доступу до даних і послуг для учасників різних приватних і державних організацій. Отже, довіру і ідентичність займає ключову позицію як окремий шар у е-інфраструктурі екосистеми. Федеративні авторизації і аутентифікації інфраструктури спрощують доступ до міжорганізаційних ресурсів, дозволяючи контролювати безпечний доступ [18] до різноманітно розташованих джерел, що дозволяє забезпечити користувачам надійне середовище зберігання цінних і цілісних даних без будь-якої шкоди. Даний шар з'єднує потужність обчислювальної

мережі та інфраструктури даних, створюючи відкритий, надійний і безпечний європейський простір.

В [19] представлено стратегічне бачення британської електронної інфраструктури, де е-інфраструктура – це об'єднання політичних, технологічних і адміністративних служб для легкого і економічно-ефективного, спільного використання розподілених електронних ресурсів. Е-інфраструктуру слід розглядати як живу екосистему, яка є відкритою і доступною для міжнародних співробітництв і постійно адаптується до мінливих вимог життя. Прориви у створенні окремих компонентів електронної інфраструктури повинні звільнити вчених від складних і відволікаючих бізнес-обчислень. Слід зазначити, що для благополучного розвитку досліджень в цій області [20] важлива відкрита конкуренція, співробітництво і технологічні інновації. Е-інфраструктура може бути встановлена тільки шляхом спільного і дійсно загального стратегічного плану зусиль між користувачами і організаціями, що беруть участь в розробці і експлуатації електронних інфраструктур. Досягнення значного прогресу на шляху до розвитку е-інфраструктури здійснюється за рахунок інвестування: у 2011 році британський уряд виділив 160 млн £ в області високопродуктивних обчислень і мереж; у 2012 році 189 млн £ для роботи з великими обсягами даних і енерго-ефективних обчислень. Таким чином, е-інфраструктура в Великобританії сьогодні є складною взаємодією сервісів, електронних пристроїв, мереж, послуг передачі даних, кібербезпеки і навичок людей, які створюють роботу електронної інфраструктури. Як показує практика, здатність генерувати, збирати і обробляти великі дані для експлуатації в промисловості, у наукових колах і державних установах дає реальну можливість підвищити конкурентоспроможність і зростання економіки кожної держави. Е-інфраструктура має вирішальне значення для інновацій в цілому ряді галузей. Наприклад, вона відіграє важливе значення повного використання можливостей і економії ресурсів [21], які приносять технологічні розробки. Так само займає важливе значення в екосистемі ресурсів, що дозволяє

виконувати великомасштабне моделювання, аналіз і візуалізацію великих даних. Е-інфраструктура відноситься до екосистеми ресурсів, яка здійснює розподілене співробітництво, великомасштабне моделювання, обчислення та аналіз великих даних, містить розробку та використання передових інструментів для використання потенціалу наявних даних і ресурсів (рис. 1.1).

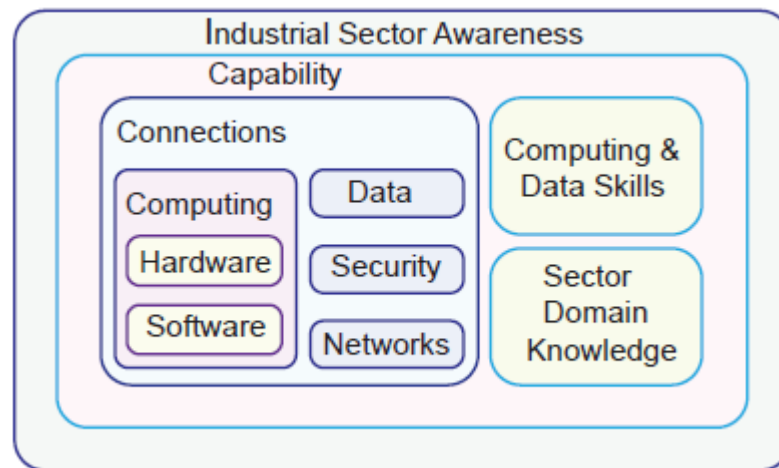


Рис. 1.1 – Подання компонентів Е-інфраструктури

В [22] розглядається історія розвитку систем управління міським транспортом і описується використання комп'ютерних систем для управління трафіком. Показано, що підвищення продуктивності комп'ютерів, поява LAN-мереж і хмарних обчислень (cloud computing) дозволили удосконалити технології управління, що пройшли шлях розвитку від централізованих моделей до розподілених систем на основі мобільних гаджетів і хмарних технологій. Запропоновано систему PtMS (parallel transportation management systems), ядром якої є «Artificial transportation system». Тут паралельно з діючою транспортною системою використовується віртуальне середовище, яке містить бази даних зі стандартними шаблонами управління трафіком, агентами і сценаріями руху. Як мобільні сенсори виступають будь-які детектори трафіку. Суть дослідження зводиться до інтелектуальної хмарної обробки дорожніх ситуацій, які відслідковуються дорожніми сенсорами. Актuatorні впливи з управління трафіком приймаються на основі заздалегідь

прописаних алгоритмів і стратегій. У разі відсутності шаблону рішення складне завдання декомпозується на більш прості. Переваги. Даний підхід "local simple, remote complex", запропонований в статті, орієнтований на інтелектуальну організацію управління транспортом в рамках Smart Cloud Traffic Control. Запропонована PtMS є складовою частиною системи «Adapts» (Adaptive Platforms for Transportation Systems), яка складається з 3 рівнів: організації, координації та виконання завдань по обробці дорожніх ситуацій. Організаційний рівень представлений 4 функціями: декомпозиція задачі, планування, інкапсуляція і управління. Це дає можливість істотно підвищити якість прийнятих регуляторних впливів за рахунок ускладнення алгоритмів. Недоліки. За певний проміжок часу можна обробити обмежену кількість перехресть. Для запуску і тестування системи на сервері накладаються істотні обмеження за обчислювальною потужністю і пам'яттю. Такі обмеження не відповідають реальним дорожнім умовам у великих містах, тому система повинна використовувати кілька серверів для online вирішення завдань управління транспортом на великій кількості перехресть. Для зберігання даних про трафік і дорожніх ситуаціях необхідні дата центри. У статті запропоновано два рішення даної проблеми: обладнати центри суперкомп'ютерами [23], або використовувати технології хмарних обчислень, такі як Google's Map-Reduce, IBM Blue Cloud і Amazon.

В [24] розглядається стандарт IEEE 802.21, який підтримує різні типи систем бездротового зв'язку, такі як сімейство IEEE 802.11, IEEE 802.16 (WiMAX) та технології стільникової передачі даних, розроблені групами 3GPP і 3GPP2. IEEE 802.21 надає незалежну передачу – Media Independent Handover (MIH) даних в стільникових мережах. Фреймворк IEEE 802.21 складається з компонентів: функція MIH (MIHF) і Service Access Points (SAP). Функція MIH є модулем, який розробляється для того, щоб надати теоретичні сервіси на більш високих рівнях за допомогою незалежного медіа-інтерфейсу. MIHF складається з трьох основних сервісів: сервіс медіа-незалежних подій (MIES), сервіс медіа-незалежних команд (MICS), сервіс медіа-незалежної інформації

(MIS). Стандартні сервісні точки доступу та сервісні примітиви дозволяють додаткам MINT мати загальний доступ до різнорідних медіа-шарів. Пропонується система FANSCI (fast handover scheme for car-to-car communication), призначена для адаптації стандарту IEEE 802.21 передачі даних, необхідних для безперервного і безперебійного зв'язку в гетерогенних мережах. FANSCI складається з двох компонентів: вишка (встановлюється в центрі управління мережею) та автомобільний компонент (встановлюється в автомобілі), реалізація FANSCI продемонстрована на рис. 1.2.

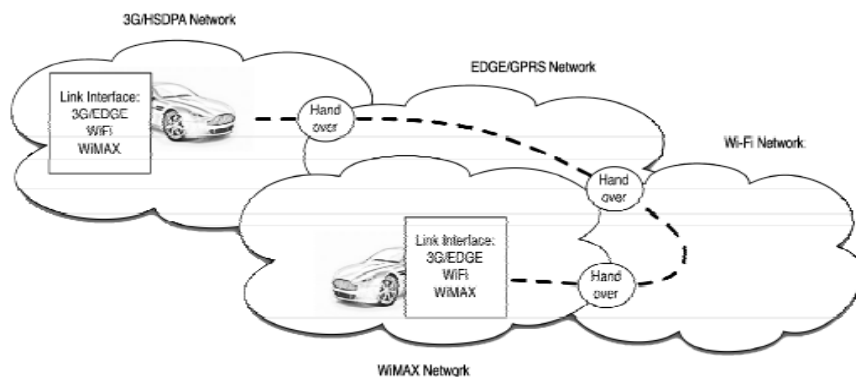


Рис. 1.2 – Реалізація FANSCI

Автомобільний компонент містить: комунікаційний модуль Car-to-Car (що складається з менеджера з'єднання, колектора GPS-інформації, колектора даних з сенсорів автомобіля, колектора інформації про двигун) і модуль обробки даних на стороні клієнта (що складається з менеджера ініціації зв'язку, трекера сигналу, буферного комп'ютера і модуля передачі даних). Компонент вишки включає: комунікаційний модуль Car-to-Car (що складається з обробника трафіку, менеджера аутентифікації, менеджера маршрутизації) і пропонований модуль для передачі даних (що складається з менеджера запитів на передачу даних, функціональний MINT модуль, буфер трафіку і редиректор трафіку). З технологічної точки зору, технологія FANSCI може бути імплементована у будь-які бездротові мережі, оскільки вона заснована на стандарті IEEE 802.21, який базується на IEEE 802.11 (Wi-Fi) і IEEE 802.16 (WiMAX). Водій може постійно отримувати інформацію про пробки, дорожню ситуацію, інші машини, що рухаються по перехрестю,

несправні двигуни автомобілів, що їдуть попереду, та інших, що знаходяться поблизу, а також про доступні гетерогенні мережі. Впровадження технології FANSCI для здійснення комунікації між автомобілями сприяє створенню безпілотних автомобілів.

В [25] демонструється хмарний контроль для прийняття відповідних заходів щодо уникнення небезпечного водіння. Автономні транспортні засоби, що рухаються у міському середовищі, обробляють інформацію і приймають рішення, які стосуються руху в режимі «stop-and-go», відповідно до дорожніх знаків і правил, обгону інших ТЗ, зміни смуги руху, входу і виходу з потоку трафіку. Автономні ТС обладнані різними типами радарів, інерційною навігацією, системою GPS, компасом і системами управління, які як правило, коштують набагато дорожче самих ТЗ. Запропоновано новий control algorithm-cloud, якому властива хороша продуктивність і гнучкість. Для створення системи автономного водіння вирішуються три завдання: 1) самостійне керування автомобілем, яке містить старт, прискорення, уповільнення руху, різки або плавні повороти; 2) рух автомобіля уздовж дороги або по центру; 3) безпечне водіння у трафіку, яке вимагає механізму визначення перешкод як для водія, так і для пішохода. Програмна архітектура складається з чотирьох частин: інтерфейс сенсорів для дороги, збір інформації про смугу руху і перешкоди, планувальник маршруту та інтерфейс ТС. Апаратна архітектура так само складається з чотирьох частин: Sensor Layer, Preprocess Layer, Integration Layer and Vehicle Layer. Для запобігання зіткнень автоматично керовані автомобілі повинні вміти розпізнавати ситуацію на смузі проїзду, а також на сусідніх смугах так, як це робить водій для безпечного водіння. Для цього в хмарному алгоритмі управління (control algorithm-cloud) використовується метод MOR (My Own Range). У нормальних умовах, для отримання MOR, необхідно розрахувати кут і відстань між машиною і перешкодами, які виявляються за допомогою ультразвукових датчиків і камер. Однак, є особливі випадки, наприклад, коли машина, що їде по сусідній смузі, має намір здійснити зміну смуги, включивши сигнал повороту. У таких

ситуаціях необхідно зробити деяке прогнозування. Його можна здійснити, відштовхуючись від дистанції і кута повороту попереду автомобіля, що рухається. Наміри про подальший рух також можна визначити за рахунок режимів водіння, які в статті поділяють на 4 основних: 1) нормальне водіння, 2) уповільнення і гальмування, 3) обгін з лівої смуги, 4) обгін з правої смуги. Обчислюються коректні команди для прискорення, гальмування і рульового управління. Це завдання виконується за допомогою контролерів швидкості і рульового управління, ґрунтуючись на правилах.

1.2 Моделі міських інфраструктур дорожнього руху

В [26] пропонується інноваційний механізм паралельного мережевого управління трафіком для хмарних обчислень, заснований на планувальнику пакетів НТВ (Hierarchical Token Buckets). Відмінністю між хмарою і традиційними центрами даних є масштаб трафіку даних і одночасний доступ користувачів. Хмара має гарантувати доступ до всіх ресурсів і послуг, за допомогою налаштованих SLA. Однак в хмарі можуть бути сотні тисяч користувачів, які одночасно звертаються до ресурсу, що створює надзвичайно сильний тиск на хмару. Для використання хмарних технологій потрібні великі смуги пропускання, однак на сьогоднішній день за допомогою НТВ можна досягти лише швидкостей 0,5 Gbps, що унеможливлює застосування даного методу в хмарах. Для вирішення даної проблеми можна використовувати механізми програмного і апаратного розпаралелювання оброблюваних даних на багатоядерних процесорах, за умови використання багаторівневої мережевої структури і модифікації послідовного коду для паралельної обробки. Використовується технологія паралелізації багатоядерних процесорів FIFO без блокування. Результати моделювання підтверджують, що паралельна пропускна здатність НТВ, що заснований на FIFO без блокування, може легко перевищити швидкість 1 Гбіт/с і досягти максимальної швидкості 2 Гбіт/с, збільшивши продуктивність до 300% у порівнянні з традиційним

НТВ. Дана технологія розроблялася тільки під Linux, завдання роботи з іншими операційними системами не розглядалися.

В [27] представлено підхід управління світлофором за допомогою агента, який ефективно керує рухом відповідно до його поточного стану. У розглянутій дорожній інфраструктурі кожен агент контролює світлофор на перехресті по циклу «спостереження – рішення – дія»: агент спостерігає за поточним станом руху у вузлі, потім використовує отриману інформацію для порівняння з правилами руху та ефективного управління потоком. Значно скорочено середній час затримки кожного автомобіля на світлофорах в порівнянні з іншими підходами. Щоб спростити завдання управління рухом, світ реальних агентів замінюється віртуальним симулятором руху. Агент, запропонований в роботі для управління світлофором, складається з трьох компонентів: спостерігача руху, бази знань і механізму виведення.

Багато дослідників намагаються модернізувати поточну систему трафіку з фіксованими часовими затримками світлофора в інтелектуальну систему з часовою затримкою в залежності від частоти трафіку. Були створені математичні моделі для світлофорів, а потім класичні алгоритми і ідеї, які використовувалися для управління світловим сигналом. Винахід [28] включає мікропроцесор, пристрій ручного введення, примусовий комутаційний пристрій і інтелектуальний пристрій виявлення. Система може автоматично регулювати параметри управління світлофором відповідно до змін в потоці транспорту у різних напрямках, підвищуючи ефективність руху на перехрестях. У роботі [29] представлена нейронна мережа (НС) є основою для законів управління, при цьому оцінка ваги НС, яка відбувається в режимі реального часу у замкнутому режимі, здійснюється за допомогою алгоритму стохастичної апроксимації одночасного збудження станів. Процедура навчання може тривати місяці або навіть роки. Результати моделювання показали 10%-ве скорочення часу очікування транспортних засобів. Були розміщені 2 електромагнітних датчики на кожній смузі для підрахунку кількості транспортних засобів. Запропоновано контролер нечіткої логіки,

який відповідає за тривалість зеленого світла відповідно до умов трафіку, і кінцевий автомат для управління послідовністю станів, через який повинен циклічно проходити контролер нечіткого трафіку.

У [30, 31] запропоновано новий підхід до точного прогнозу трафіку на перехресті за допомогою технології бездротових сенсорних мереж, що включають взаємодію розумних вузлів: управління, індикатори та транспортних засобів.

В [32-34] запропоновано інтегральний контролер швидкості і алгоритм управління перевантаженням, призначений для інтернету, для вирішення проблеми перевантаженості на сюжеті ділянці дороги складної дорожньої мережі. Запропоновано два типи пристроїв, встановлених уздовж дороги, на в'їзді і виїзді сегмента дороги для контролю обсягу трафіку і світлофорів. При виборі стану світлофора пристрої виконують «опитування» попередньо станів трафіку і динамічно встановлюють періоди часу для сигналів світлофора в залежності від умов потоку, що прибуває. Враховується також зворотний зв'язок від умов убування потоку. Досвід останніх років свідчить про те, що конструювання традиційних дорожніх інфраструктур і використання виключно традиційних підходів до управління не тільки коштовно, але і неефективно вирішує питання транспортних перевантажень. Аналіз показав, що однією з важливих причин заторів на дорогах є неефективне управління транспортним потоком. Состояние світлофора має динамічно підлаштовуватися під умови трафіку в режимі реального часу.

Інтелектуальна система управління рухом [35-36] використовує реалізацію двопроцесорної інтелектуальної системи в FPGA для відстеження частоти трафіку. Існують різні способи контролю трафіку, наприклад, обробка відеозображення [37], підрахунок транспортних засобів через пересічні масиви датчиків наближення – торкання. Нова структура управління трафіком – мобільна інтелектуальна система управління трафіком (MITCS), розроблена для Тайваню, запропонована у [38]. Аналогічним чином, інтелектуальна система управління трафіком працює у Пекіні під назвою SCOOT [39]. Автори

прагнуть створити інтелектуальну систему, яка розуміє транспортний рух і сигнальну систему руху, яка сприймає трафік, використовуючи набір безконтактних датчиків і сенсорних датчиків, встановлених на дорозі, обчислює щільність, використовуючи код, вбудований в мікроконтролер, і застосовує зміну On-The-Go. Ця система змінює затримки, що застосовуються в поточному робочому циклі світлофора, для максимального комфорту учасників дорожнього руху. Візуалізація здійснюється за допомогою світлодіодів високої інтенсивності.

В [40] представлена система управління трафіком, де навантаження дорожнього потоку безперервно вимірюється датчиками, підключеними до системи на базі мікроконтролерів. Світлофори пов'язані між собою мережею, через яку відбувається обмін інформацією про навантаження і синхронізації. Тривалість кожного циклу світлофора змінюється динамічно. Структура системи складається з трьох модулів. У першому частота трафіку визначається датчиками наближення або сенсорними датчиками на основі магнітного захоплення. У другому модулі запрограмований вбудований процесор визначає затримки сигналів світлофора відповідно до частоти трафіку. У третьому модулі вихідні дані від різних портів мікроконтролера надходять на різні світлофори. Інтелектуальна система управління трафіком, показана на рис. 1.3, працює наступним чином: 1) Датчики наближення посилають імпульс кожен раз, коли автомобіль наближається до нього. 2) Імпульс посилюється і перетворюється в цифровий імпульс. 3) Вихід подається на мікроконтролер. 4) Відповідно до кількості активованих датчиків визначається час зупинки і переходу. 5) Зміна часу здійснюється в одному циклі.

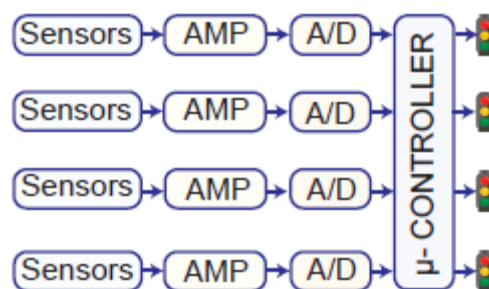


Рис. 1.3 – Інтерфейс датчика з мікро контролером

У цьому недорогому прототипі були використані малогабаритні датчики наближення і світлодіоди для терміналів світлофора. Але в реальності порт мікроконтролера не може управляти високопотужними світловими лампами. Це можна зробити за допомогою реле, яке переключається через транзистор. Число пристроїв виявлення трафіку зі світлофорами може бути збільшено за рахунок використання чіпа інтерфейсу 8255 PPI з портом мікроконтролера. Перевагою даної системи є низька вартість – 400 доларів США і висока точність, в той час як існуючі системи стоять понад 30 000 доларів.

В [41-43] запропоновано управління сигналами світлофора за рахунок обчислення параметрів нечіткого управління трафіком нейромережі. В [44, 45] використовуються інтелектуальні агенти з можливістю Q-learning для управління одним перетином. Результати моделювання показали, що ці підходи краще, ніж метод управління з фіксованим часом. Показана самонавчальна модель управління світлофором, яка заснована на нечіткій кластеризації і генетичному алгоритмі. Використовуються теорія нечіткої логіки і машинне навчання. Шляхом нечіткої кластеризації схеми управління поміщають числа прибуваючих автомобілів в базу даниху вигляді набору правил. Вони визначають управління сигналом в певних дорожніх ситуаціях відповідно до досвіду і принципами управління трафіком. Генетичний алгоритм використовується для поліпшення набору змінних правил і є частиною процесу управління сигналами, виконуючи роль розрахунку оптимального управління. В [46] представлена скоординована система управління дорожнім рухом на основі мультиагента. Вона має ту ж структуру, що і модель в [47-49], яка являє собою систему управління стилем піраміди з трьома рівнями, які представляють собою глобальне, групове та індивідуальне планування зверху вниз.

В [50] реалізована розподілена і скоординована система управління сигналами руху, яка ґрунтується на мультиагентах (DTCMAS) без центру управління. Кожне перехрестя контролюється локальним агентом незалежно один від одного, проте зв'язок між ними існує, і вона відбувається у суміжних

перехрестях. Таким чином, якщо масштаб дорожньої мережі збільшується, складність алгоритму управління залишається незмінною. Внутрішня структура агента показана на рис. 1.4. Кожен агент управління сигналами складається з частин: інтерфейс зв'язку, процесори сусідніх з'єднань, локального з'єднання і виявлення автомобіля.

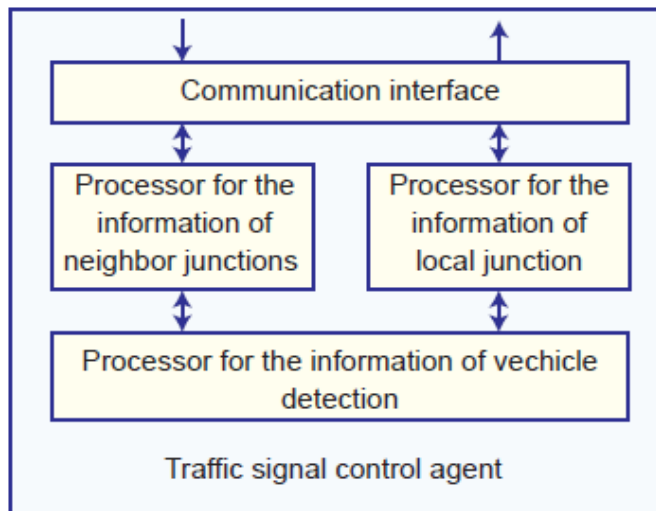


Рис. 1.4 – Структура агента управління сигналами.

Оскільки кожен агент містить базу даних і механізм міркувань, DTCMAS не тільки повністю враховує інформацію про стан руху на перехрестях, але також виконує глобальну оптимізацію для управління рухом дорожньої мережі і зменшує складність алгоритму управління. Результат моделювання показує, що підхід управління сигналами DTCMAS зменшує загальну затримку зупинки автомобілів в середньому на 37,8% у порівнянні з методами фіксованого часу і знижує загальну затримку автомобілів у середньому на 17,8%, в порівнянні з іншими методами за тих же умов.

1.3 Системи позиціювання та навігації транспортних засобів

В [51] розглядається проблема зростання кількості автомобілів в місті, а також невідповідність пропускну здатності доріг такому інтенсивному росту. Пропонується використовувати системи динамічної (DVNS) і статичної (SVNS) навігації [52] для моніторингу та управління транспортними засобами. Дорожня мережа і архітектура динамічного трафіку з використанням

просторово-часової моделі даних може бути описана за допомогою технологій: 1) зіставлення і злиття простору мульти-модального дорожнього трафіку і атрибутів інформації; 2) оновлення інформації про дорожню мережу для різних проміжків часу; 3) створення бази даних масштабованої дорожньої мережі. Система DVNS складається з трьох рівнів: базових і топологічних мереж, а також інформації про дорожній рух. Уявити функціонування дорожньої мережі можна, ґрунтуючись на вузлах або на сторонах руху. У запропонованому підході позиціонування [53, 54] транспортних засобів здійснюється шляхом використання GPS-сигналу, а у випадках щільної міської забудови, тунелів застосовується фільтр Кальмана [55]. Показані методології та моделі моніторингу дорожніх мереж, принципи управління ними на основі просторово-часових критеріїв, а також короткострокове прогнозування дорожньої ситуації з метою створення високоефективної масштабованої мульти-модальної навігаційної системи. Запропонований підхід дозволяє вирішити завдання збору і обробки інформації про дорожній рух, що включає потоки трафіку в режимі реального часу, дорожні правила, а також інформацію про дорожні пригоди з метою подальшого управління об'єктами дорожньої інфраструктури. Використання тільки однієї технології бездротового зв'язку для визначення місця розташування транспортного засобу скорочує область використання даної системи до зони покриття GPS [56, 57]. Хмарні сервіси не розглядаються. В [58] запропоновано систему динамічної навігації, представлена її архітектура і методика організації та управління даними. Традиційні системи навігації [59] транспорту мають недоліки: 1) Відсутність актуальної інформації про дорожню ситуацію і про трафік в режимі реального часу; 2) Навігація без урахування змін у трафіку і інформації дорожньої ситуації; 3) Відсутність можливості для з'єднання з іншими мережами обслуговування. Система динамічної навігації складається з трьох частин (рис. 1.5): Traffic information Provider, Service Center і Navigation Terminal.

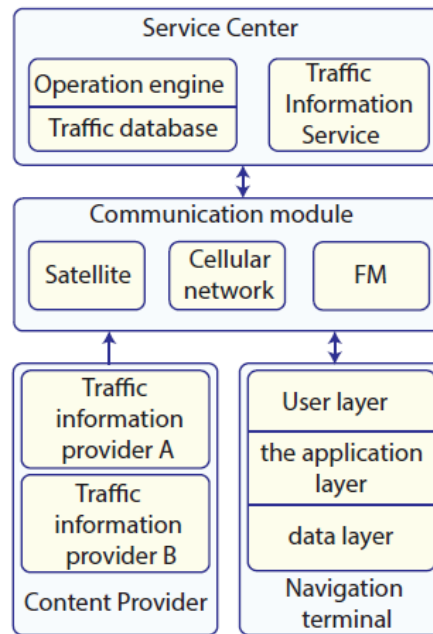


Рис. 1.5 – Системна архітектура

Traffic information Provider – основна служба дорожнього руху в Китаї, яка отримує інформацію про дорожній трафік з відеодетектора, мікрохвильових детекторів, систем ідентифікації автомобільних прав і системи моніторингу проїзду на червоне світло. Так само використовуються транспортні засоби, які при русі в потоці збирають дані за допомогою бездротових мереж в режимі реального часу з метою подальшої обробки масивів даних в сервісному центрі. Service Center – розподіл і обробка всієї навігаційної інформації, яка складається з таких компонентів: operation engine (ПЗ проміжного шару, яке надає сукупність інтерфейсів для перетворення інформації в уніфікованому вигляді і зберігання її в базі даних), база даних трафіку (використовується на додаток до інформації про real-time трафіку для надання рекомендованого оптимального маршруту і очікуваного часу прибуття з більшим ступенем точності), служба дорожньої інформації (надає всі види сервісних модулів: сервіс обмеження трафіку, сервіс пробок і сервіс погоди, новинний і сервіс розваг) і комунікаційний модуль (забезпечує зв'язок між сервісним центром, контент-провайдером і мобільними терміналами). Navigation terminal – здійснює функції відображення карти, рекомендації маршруту, планування маршруту і пошук. Термінали масштабуються,

складаються з комунікаційного модуля, рівня даних, додатку і виконання. В системі GIS-T (Geographic Information System for Transportation) навігаційні дані складаються з статичних даних, не змінних на протязі тривалого часу (існуючі дороги, сервісні станції та заправні станції) і динамічних (потік руху, дорожні події). Просторові характеристики показують зміни розподілу завантажених ділянок дороги в різний час. Часові характеристики показують, що стан завантаженості на одній і тій же дорозі змінюється з плином часу. Запропонована система орієнтована на збір актуальної інформації про дорожню ситуацію і про трафік в режимі реального часу; на створення навігації з урахуванням змін трафіку і інформації про дорожню ситуацію; можливості забезпечення з'єднання з іншими мережами обслуговування. Експериментальна система, що базується на методиці запропонованої в даній статті, вже імплементована в урядовій організації збору та надання інформації про трафік в Пекіні. Багато розвинених країн займаються розробкою систем позиціонування (VNS) – "VICS" в Японії, "TravTek" в США і "Ali-Scout" в Європі. Недоліки. Через проблеми урбанізації важко реалізовувати системи VNS з real-time інформацією про трафік. У запропонованій системі не організовано управління інтеграцією real-time інформації з дорожніми мережами. В [60] пропонується система автомобільної навігації, заснована на гібридному генетичному алгоритмі. Обчислення маршруту – задача комбінаторної оптимізації, поряд із завданням "мандрівного торговця". Для її вирішення використовується генетичний алгоритм (ГА) [61, 62], як потужний засіб оптимізації. Логіка роботи полягає в пошуку маршруту з декількома точками проїзду до прибуття в кінцевий пункт призначення. Завдання вибору маршруту розподіляється на 2 підзадачі: вибір точок проїзду (вирішується за допомогою ГА), вибір між цими точками (вирішується за допомогою алгоритму Дейкстри). Надається можливість визначити оптимальний спосіб побудови маршруту, виконуючи найбільш важливу функцію для системи автомобільної навігації. Алгоритми Дейкстри і Мазе визначають найкоротший маршрут між двома точками, проте не вирішують задачу з кількома точками.

В [63] пропонується модель руху зі змінною швидкістю, ядром якого є контролер, заснований на хмарній нейронній мережі для здійснення рульового управління. В рамках проекту «Інтелектуальна система транспортування» було розроблено сукупність підходів моделювання трафіку, які вирішують завдання, пов'язані з пробками і дорожньо-транспортними пригодами. Всі підходи засновані на системі Equation Based Modeling (EBM). Незважаючи на успіхи в моделюванні мережі реального трафіку, досі не були емульовані складні процеси поведінки водія на дорогах [64]. Моделювання дій водія – складне завдання, що включає теорію управління, робототехніку та психологію. Симуляція рульового керування водієм є ключовою технологією в автономних системах руху. При виконанні алгоритму навчання нечітка область виражається системою лінійних рівнянь для локальної апроксимації нелінійної функції. Це відображає накопичений досвід водія з такими ж умовами водіння. Отриманий в результаті моделювання сценарій зміни смуги демонструє ефективність алгоритму навчання, заснованого на хмарній моделі і нейронних мережах. Запропонована модель дозволяє частково здійснювати "driver-free car control", з огляду на рух попереду і ззаду транспортного засобу, що рухається.

В [65] описується проект AutoNomos – управління пробками на однополосній дорозі. Динамічно змінювана система, як дорожня пробка, складається з множини постійно змінюваних об'єктів, наприклад, машин, розташованих в різних місцях, що продовжують рухатися в кінці або на початку пробки. Для отримання корисної інформації, що співвідноситься з кінцем затору, потрібно враховувати зміщення одного автомобіля по відношенню до іншого, разом з усіма відповідними даними. Подальший розвиток проекту направлено на інтелектуальну організацію управління трафіком на двополосному шосе (з перехрестями і різними виїздами), а також на вирішення проблеми «пляшкового горлечка» (вузьких місць) в трафіку, наприклад, викликані конвоєм, повільними вантажівками, аваріями або спеціалізованими транспортними засобами.

В [66] пропонується рішення ключової проблеми міського руху – інтелектуальна модель управління міською магістраллю. Показується вплив контролю швидкості на потік трафіку; розробляється стратегія попереднього контролю для забезпечення максимальної ефективності системи міських магістралей, особливо в умовах завантаженості; представлена система хмарного прогнозування [67] пробок на дорогах, яка дозволяє реалізувати стратегію випереджувального керування транспортним засобом. Заропонована інтегрована розумна динамічно керована архітектура (рис. 1.6) складається з 3 підсистем: 1) вимірювання даних про трафік на магістралі; 2) управління інформацією про трафік; 3) попереднього контролю швидкості для поліпшення рівня ефективності та безпеки дорожнього руху.

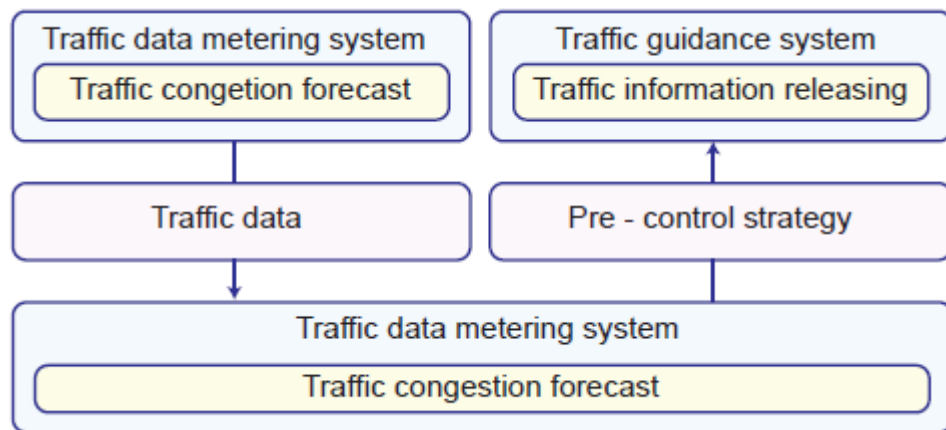


Рис. 1.6 – Архітектура інтелектуальної системи управління

Зниження швидкості призводить до зменшення заторів у «вузьких місцях», що може поліпшити загальну продуктивність магістральної системи. «Вузьке місце» в даному контексті означає не тільки те місце, яке має більш низьку пропускну здатність трафіку [68], ніж інші частини магістралі, а й місце, в якому деякі затори будуть відбуватися на основі прогнозу заторів на дорогах. У пропонованому рішенні не вказується застосування хмарних технологій.

В [69] розглядається система адаптивного управління дорожнім сигналом (TSC), як продовження розвитку інтелектуальної транспортної системи (ITS). Досліджується система спостереження на основі аналізу

непередбачуваних і важко вимірюваних порушень, що впливають на дорожню ситуацію. Використано методології відео-спостережень, які включають сегментацію, класифікацію і відстеження об'єктів для визначення стану міської магістралі в режимі реального часу. Алгоритм управління адаптивним сигналом дорожнього руху погоджує перемикання світлофорів в ситуаціях «руху по прямій», «повороту направо» і «повороту наліво». Ефективне, безпечне і менш забруднюване транспортування людей і товарів вимагає оптимального використання інфраструктури за допомогою застосування різних заходів контролю дорожнього руху. Керування транспортними засобами безпосередньо залежить від ефективності та актуальності застосовуваних методологій управління [70, 71, 72, 73, 74, 75, 76, 77, 78, 79]. Для цього визначаються і аналізуються об'єкти, потім вимірюється кількість автомобілів, після чого інформація про транспорт екстраполюється. Існують три основні методики виявлення транспортних засобів на магістралі: «часова відмінність», «оптичний потік» і «віднімання фону». Методи часового диференціювання і оптичного потоку дозволяють обробляти зображення і знаходити рухомі об'єкти, але вони не підходять для нерухомих або повільно рухомих об'єктів. Результати експерименту з використання системи управління сигналами [70-80] дорожнього руху підтверджують ефективність адаптивного підходу TSC на основі відео-спостереження.

Метод S-TRAC (system-wide traffic-adaptive control) [81] був введений як засіб для створення оптимальних таймінгів для сигналів у режимі реального часу. S-TRAC має кілька особливостей, які роблять його практично здійсненним і теоретично обґрунтованим у вирішенні проблем управління транспортом. Серед цих особливостей: (1) не потребує ніякої моделі потоку трафіку в масштабі всієї системи; (2) S-TRAC автоматично адаптується до довгострокових змін у системі (наприклад, сезонних змін), забезпечуючи при цьому команди реагування в режимі реального часу; (3) S-TRAC може працювати з існуючими апаратними засобами і конфігураціями датчиків в використаній мережі.

1.4 Електронне обладнання сучасного автомобіля

Тенденції розвитку конструкцій автомобілів обумовлені економічними, соціальними і екологічними причинами: підвищення паливної економічності і безпеки автомобілів, забруднення навколишнього середовища відпрацьованими. Автомобільні технології набирають темпи розвитку, завдяки комп'ютеризації та впровадження «smart» компонентів в бортові системи автомобілів. Фінансові та енергетичні кризи також спричинили за собою зростання інтересу і наукові дослідження в області створення електромобілів і гібридних автомобілів, що використовують енергію з відновлюваних джерел. Збільшення обсягів трафіку у великих мегаполісах призводить до ідеї використання оцифрованого трафіку і систем моніторингу та управління ним, а також постійні дорожні пробки та часткові випадки ДТП вимушені прийняти концепцію «free human control car» і максимально автоматизувати процес управління рухом транспортних засобів [82-84]. Розробкою і моделюванням автомобілів майбутнього цікавляться і ведуть дослідження такі компанії як: Apple, Google, Tesla, Synopsys, світові автогіганти BMW, VW, Mercedes, Toyota, Ford, Honda. Існує багато наукових центрів з розвитку інфраструктури автомобілів, в яких співпрацюють університети і компанії, наприклад, центр автомобільних досліджень (CAR) в Коледжі інженерії штату Огайо [85]. Hitachi automotive systems group пропонує різні рішення з питань розвитку сучасного автомобіля [86]: екологічно чисті високоефективні системи управління двигунами для скорочення викидів вуглекислого газу і запобігання забруднення повітря; поставка інтегрованого електромеханічного рішення прокладання дороги для електроустаткування з механічним і електронним управлінням; послуги хмарної інформаційної мережі, багатофункціональні навігаційні системи та інше інформаційне обладнання. Компанія надає широкий вибір сенсорів [87]: Airflow Sensor (slot-in, robust and silicon types), Differential Pressure Sensor, Exhaust Pressure Sensor,

Pressure Sensor, Hall Effect Type Revolution Sensor. Глобальна мережа заводів, магазинів розташована по всьому світу: Європа, Азія, Китай, Америка.

Автомобіль майбутнього оснащений бортовим комп'ютером, сенсорами [88], актюатором [89], радаром [90], лидара [91], системою GPS [92], супутникові антени [93], системою безпеки. Архітектура транспортного засобу (рис. 1.7) дозволяє взаємодіяти з хмарними технологіями і використовувати всілякі сервіси [94] для поліпшення якості пересування.

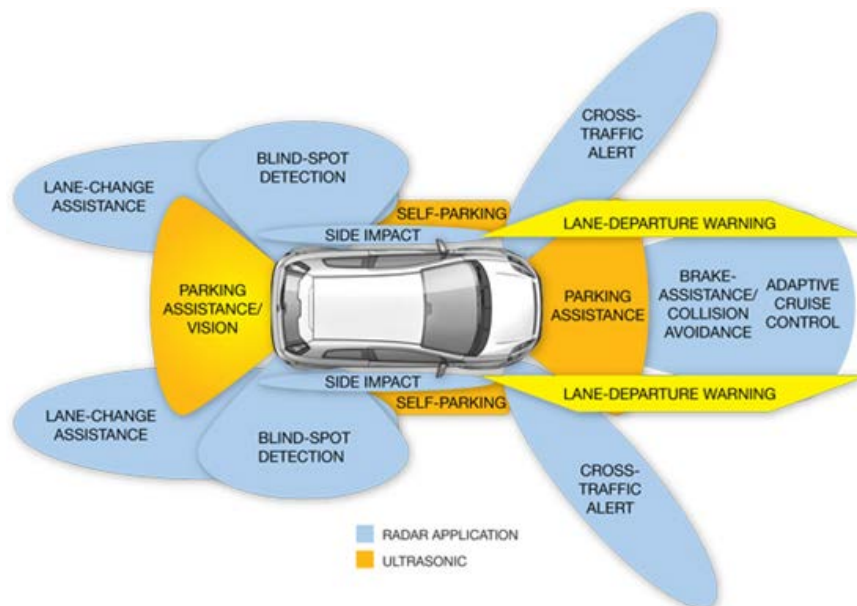


Рис. 1.7 – Використання радар технологій

Вбудовані Satellite системи дозволяють здійснювати бездротовий зв'язок автомобіля з хмарними сервісами, об'єктами інтелектуальної дорожньої інфраструктури і зв'язку з мобільним пристроєм водія, а також іншими транспортними засобами в потоці в режимі реального часу. Використання сенсорів і актюаторів дозволяє здійснювати збір даних і моніторинг параметрів, а також здійснювати на підставі цього управління всіма аналоговими і цифровими системами автомобіля. Вбудований бортовий комп'ютер відповідає за обробку даних, одержуваних від датчиків і різних систем автомобіля, призначених для бездротового зв'язку з хмарними сервісами і дорожньою інфраструктурою. Взаємодія машин один з одним відбувається за рахунок встановлення контакту шляхом зв'язку з хмарою, або

шляхом використання каналів і протоколів зв'язку на ближній дистанції [95]. У разі викрадення машини власник отримує унікальну можливість повноцінного віддаленого спостереження за місцем розташування автомобіля, використовуючи GPS, вбудовані бортові камери і камери зовнішнього огляду автомобіля, а також шляхом отримання на мобільний телефон або інший пристрій SMS, e-mail або іншого виду повідомлень з даними про останньому зафіксованому в системі розташування його автомобіля. Спеціальним службам дорожнього контролю, поліції буде посланий сигнал про несанкціонований доступ до автомобіля, передані координати і параметри автомобіля. Як відповідна реакція може бути повне блокування ТЗ аж до дистанційного блокування двигуна, КПП і гальмівної системи. З розвитком обладнання автомобілів (рис. 1.8) змінюються і удосконалюються елементи дорожньої інфраструктури: розумні світлофори, віртуальні дорожні карти, знаки дорожнього руху. Сучасні світлофори стають інтелектуальним засобом взаємодії транспортних засобів з хмарним сервісом, будучи приймачем сигналів, які надходять від автомобілів через хмару [96].



Рис.1.8 – Сучасний автомобіль

Таким чином, комп'ютеризоване оснащення сучасного автомобіля робить його все більш високоінтелектуальним роботом, що розумно взаємодіє з навколишнім середовищем і хмарою за допомогою сенсорів і актюаторів при його точному позиціонуванні в просторі за рахунок каналів позиціонування і навігації.

1.5 Кіберфізичні системи хмарного управління транспортом

Процеси несинхронізованого розвитку двох компонентів: реального світу і кіберпростору в даний час набувають векторний формат, спрямований на створення структурованих і взаємно інтегрованих компонентів КіберГеоСистеми (КГС). Остання еволюціонує шляхами створення в кіберпросторі хмарних сервісів точного моніторингу та оптимального управління недосконалим реальним світом на основі моделей взаємно однозначного відображення всіх земних процесів і явищ. Спрощено КГС можна представити двома модулями: «хмарного» кібер управління і технічного земного виконання, представленого ресурсами, засобами і людьми. Механізми управління і виконання не повинні перетинатися з матеріальних і людських ресурсів. Формально КГС представляється у вигляді двох механізмів: Cyberity і Humanity, які пов'язані між собою чотирма сигналами: Моніторингу, Управління та Ініціювання обох механізмів з метою реалізації бажань. Аналітична форма завдання СН-системи і її структурний еквівалент зображені на рис. 1.9.

$$\begin{aligned}
 S &= (C, H, M, U, X, R, Y, P), \\
 C \square H &= \square; \\
 \begin{cases} Y(t) = f[X(t), (R(t), M(t), Y(t-1))]; \\ P(t) = g[X(t), (R(t), U(t), Y(t))]. \end{cases}
 \end{aligned}$$

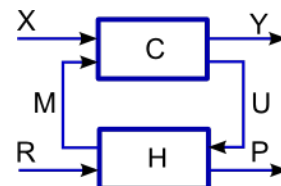


Рис. 1.9 – Аналітична і структурна форми завдання СН-системи

Тут представлені (C, H, M, U, X, R, Y, P) відповідно: блоки управління (менеджери і кібери – cybers) і виконання (інфраструктура, виконавці та

роботи); сигнали моніторингу та управління, входи керуючих ідей і виконавчих ресурсів (час – гроші – матеріали), виходи індикації стану алгоритму реалізації ідеї і випуску продукції (сервісу).

В рамках створення КГС актуальними і ринково привабливими є такі напрями досліджень: 1. Технології диференціювання мобільних систем на спеціалізовані та структурований кіберпростір хмарних сервісів. 3. Квантові паралельні обчислення і спеціалізовані мультипроцесори. 4. Сенсорні інтелектуальні пристрої для реального світу і швидкодіючі роботи для кіберпростору. 5. Тривимірні мультипроцесори, 3D-обчислювальні процеси і багатовимірний кіберпростір. Тут мова йде про неминучі переходи обчислювальних пристроїв, процесів і компонентів в 3D-вимірювання у зв'язку з появою тривимірного FinFETs транзистора. 6. Це приведе весь комп'ютерний світ до реалізації абсолютно нової парадигми – паралельного програмування обчислювальних структур в трьох вимірах. 7. Важливим є створення 3D-мультипроцесора на об'ємному кристалі, що стане можливим у найближчі 5 років. Тут слід впроваджувати трикутну, а також тетраедну структуру міжкомпонентних сусідніх зв'язків як базових елементів для формування обчислювальних обсягів мікро- і мультипроцесоров, а також кіберпростору сервісів. 8. Стає актуальною розробка 3D-принтерів для реалізації технологій друкування (виготовлення) тривимірних оптимальних спеціалізованих архітектур, функціонально орієнтованих під конкретні алгоритми вирішення.

Еволюція кіберпростору планети розподіляється на наступні періоди: 1) 1980-і роки – формування парку персональних комп'ютерів; 2) 1990-ті роки – впровадження Інтернет-технологій у виробничі процеси і побут людини; 3) 2000-і роки – підвищення якості життя за рахунок впровадження мобільних пристроїв і хмарних сервісів; 4) 2010-і роки – створення цифрової інфраструктури моніторингу та активного управління рухомими об'єктами (повітряний, морський, наземний транспорт і роботи). Тому в даний час привабливою проблемою є системна інтеграція хмарного сервісу моніторингу

та управління, блоків радіочастотної ідентифікації транспорту, а також цифрових засобів дорожньої інфраструктури для оптимального on-line керування транспортними засобами та дорожнім рухом з метою вирішення соціальних, гуманітарних, економічних та екологічних проблем.

Що є основою кіберпростору планети? Кристал кремнію і його аналоги. Сучасна мікроелектроніка дає можливість створювати вже не плоскі, а об'ємні транзисторні структури (3D – FinFETs) в 14-7, 5, 3.5 і 2.5 нанометровому діапазоні, порівнянними з розміром атома. Це означає появу об'ємних 3D-System-on-Chip. Переваги таких об'ємних кристалів істотно впливають на характеристики промислових виробів в частині: енергоспоживання, розмірності, швидкодії, вартості та якості за рахунок зменшення не тільки розмірності компонентів, але і зв'язків між ними. Таким чином, мікросвіт кіберпростору переходить в 3D-вимірювання. Макросвіт все ще залишається плоским при об'єднанні в систему націнок компонентів, комп'ютерів, мереж, хмарних сервісів кіберпростору. Які аргументи можна висунути за перехід макросвіту в 3D-простір? Компактність інформаційних обсягів, швидкодія переміщення в кіберпросторі і його розмірність. Трикутна плоска структура системи, де всі вершини є сусідніми, має істотний недолік в двох вимірах – кодування трьох вершин або ребер займає три коди, а це означає, що один код двохбітового вектора не використовується. Тому створити примітивну структуру, де всі вершини є сусідніми, а їх кількість дорівнює чотирьом для повного використання кодового простору двох бітів, значить знову відкрити дивовижну за своїми властивостями 3D-фігуру – тетраедр. Він має шість ребер або відстаней, хог-сума яких дорівнює нулю. При завданні фігури два ребра є надлишковими, що можна використовувати для зменшення інформаційних обсягів до 66% при зберіганні і передачі даних. Формування кібернетичного простору на основі використання примітивів-тетраєдрів дає можливість оптимізувати ставлення структурної складності простору до середньої відстані між двома точками.

Куди сьогодні прагне реальний кібернетичний світ? Корпоративні мережі, персональні комп'ютери, а також окремі сервіси (програмні продукти) йдуть в хмари кіберпростору, які мають яскраво виражену тенденцію до розшарування Інтернету по спеціалізованим сервісів, рис. 1.10. Якщо сьогодні 4 мільярди користувачів з'єднуються в інтернеті (1 zettabytes = 10^{21} = 2^{70} байт) за допомогою 50 мільярдів гаджетів, то через п'ять років кожен активний користувач буде мати не менше 10 пристроїв для зв'язку з кіберпростором. Стає неможливим використання персональних комп'ютерів без часткового або повного їх відображення або синхронізації на хмарах Інтернету. Це дає можливість вирішувати проблему віддаленого доступу до особистих даних і сервісів персонального комп'ютера при переміщенні користувачів в просторі. Економічний чинник неефективного використання придбаних додатків, розміщених в гаджетах і персональних комп'ютерах, змушує користувача відмовлятися від їх покупки на користь майже безкоштовної оренди сервісів на хмарах. Все згадане вище є істотним аргументом і незаперечним доказом неминучого переходу всього людства в кіберпростір віртуальних мереж і комп'ютерів, розташованих в професійно надійних хмарах сервісів. Хмарні технології знімають практично всі згадані вище проблеми надійності, безпеки, сервісного обслуговування та практично не мають недоліків. Сказане безпосередньо стосується й сервісу дорожнього руху, який має цифрове відображення у кіберпросторі для подальшого моделювання всіх процесів на хмарі з метою запропонувати кожному водієві якісні умови пересування з економією часу і коштів.

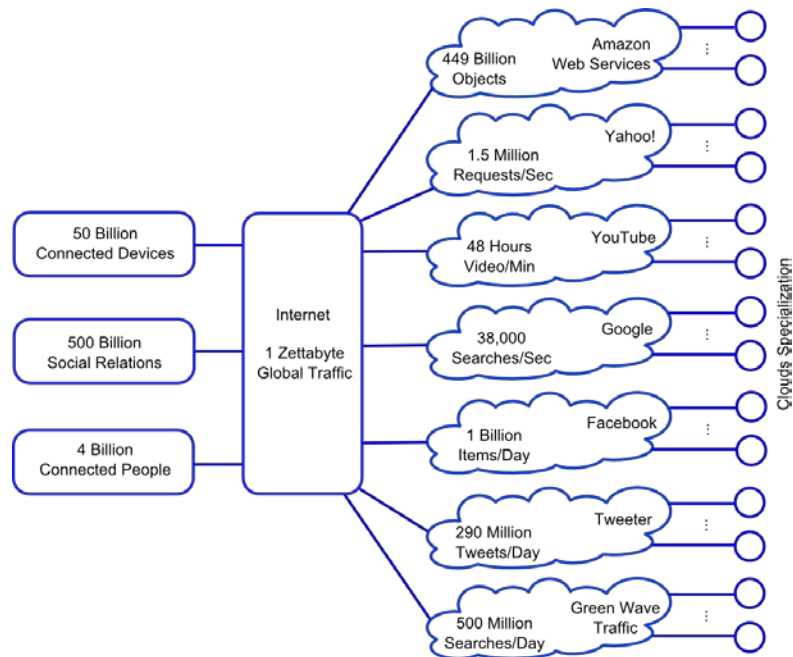


Рис. 1.10 – Віртуалізація реального світу

Кіберфізична система online хмарного управління транспортом орієнтована на надання сервісів для 9 мільйонів водіїв України і 7 тисяч транспортних компаній. Аналогів таких систем у світі поки що не існує. Але є окремі компоненти для створення е-інфраструктури: електронні карти, супутникові системи локації і навігації, спеціалізовані бази даних в хмарах, засоби моніторингу, збору та захисту інформації, автомобільні комп'ютери і водійські гаджети з прийомопередавачами, централізовано керовані світлофори, стільниковий зв'язок, як частина необхідної інфраструктури для реалізації проекту. Існує розуміння з боку держави, громадян і поліції про необхідність створення і використання інтелектуальної інфраструктури і хмарного сервісу для якісного і безпечного дорожнього руху.

1. Теоретичні розробки інтелектуальних моделей, методів і програмно-апаратних засобів аналізу кіберпростору, пов'язані з дискретної оптимізацією, пошуку, розпізнавання та прийняття рішень [97-100].

2. Досвід розробки і застосування вбудованих і RFID цифрових систем для моніторингу дорожнього руху [101-105].

3. Досвід розробки і впровадження програмних продуктів і хмарних сервісів для оптимізації маршрутів транспортних засобів українських підприємств з метою мінімізації матеріальних і часових витрат, підвищення якості обслуговування пасажирів [106-111].

4. Розробки розподіленої системи управління дорожнім рухом в умовах великих міст і мегаполісів на основі високонадійної обчислювальної техніки [107-109].

5. Існуючі системи моніторингу дорожнього руху в США, Канаді та Японії – OnStar і NEXCO Central. Система OnStar орієнтована на моніторинг окремих машин, NEXCO Central здійснює глобальний моніторинг трафіку на основних і найбільш жвавих магістралях країни. OnStar сервіси доступні власникам автомобілів: Acura, Audi, Isuzu, Subaru, Volkswagen. На даний момент налічується близько 4 000 000 користувачів даного сервісу. Вартість одного пристрою моніторингу близько 200 \$. Використовується CDMA канал зв'язку, що надається переважно Verizon Wireless в США і Bell Mobility в Канаді. Для визначення місця розташування використовується GPS. Є можливість голосового зв'язку з операторами. Інформація з сенсорів, в основному це датчики ударів і спрацьовування подушок безпеки, автоматично передається в call-центри. Це дозволяє негайно сповіщати про місцезнаходження аварії рятувальні та правоохоронні органи. Крім цього, всі машини, обладнані цією системою, мають GPS передавач, який дозволяє відстежити викрадений автомобіль. Існує також можливість отримання інформації про швидкість, витрату палива, напрямку руху і стиль водіння автомобіля. Дана інформація використовується страховими компаніями для розрахунку вартості індивідуальних страхових полісів. Нові моделі автомобілів обладнуються системою віддаленої зупинки двигуна. Автомобіль можна завести тільки після введення спеціального секретного коду. Вартість сервісів – Safe & Sound – 18.95 \$ в місяць. Тариф включає автоматичне оповіщення про аварії, моніторинг викраденого автомобіля, аварійні сервіси. Directions & Connections – 28.90 \$ в місяць. Додатково є можливість

моніторингу напрямку руху і стилю водіння автомобіліста. Система NEXCO Central розроблена Japan Highway Public Corporation. Принцип роботи полягає в глобальному моніторингу дорожнього руху на головних автострадах країни. Система покриває близько 2000 км доріг. Дата-центр обробляє дані, одержувані від датчиків з хвилинним інтервалом за допомогою глобальної IP мережі. На дорогах встановлено 744 точок зняття і передачі інформації по телефонних каналах про дорожню ситуацію.

6. Глобальні інфраструктурні проекти кілька століть поспіль об'єднують країни. Трансконтинентальні залізниці і автомагістралі, підводні тунелі і гігантські мости, космічні та авіаційні проекти служать найсильнішим каталізатором для розвитку економік багатьох держав. Розвиток мобільних і хмарних технологій зробили можливим безмежне масштабування обчислювальних потужностей і обсягів збережених даних. Наступний шар кіберпростору – Internet of Things (Car), що передбачає спілкування об'єктів між собою без людини, автоматизований збір, обробку та аналіз big data, що генеруються розумними сенсорами. Стає можливим побудова високоточних моделей поведінки навколишнього середовища і життєдіяльності міст, які стають розумними, що дозволить значно підвищити рівень життя, комфорту та безпеки водія за рахунок постійного моніторингу показів інтелектуальних датчиків у реальному часі.

7. Розвиток і впровадження концепції Internet of Things вимагає вирішення складних інженерних і наукових проблем в області комп'ютерної інженерії (<http://eai.eu/>, <http://iot.ieee.org/>), якими займаються світові науково-дослідні спільноти (IEEE, EAI) і компанії (Intel, NASA, IBM, Apple, Google, Samsung, Dell, AT & T, CISCO, THALES, WorldSensing, Aguila technologies, Connit, SigFox, Guglielmo, DQuid, BitCarrier). Ідея Internet of things заснована на використанні бездротових гетерогенних мереж, що працюють в різних частотних діапазонах за стандартами ZigBEE, WiFi, LTE (3G, 4G, 5G), Bluetooth, які забезпечують дальність дії, швидкість передачі даних і низьке енергоспоживання. Функціональність інфраструктури Internet of things – збір і

аналіз даних використовує структури: big data, open data, cloud computing. Перші орієнтовані на аналіз великих обсягів неструктурованих даних. Другі – це концепція форматування інформації для використання комп'ютерами без втручання людини. Хмарні обчислення є масштабованою платформою для обробки і зберігання big data, що надходять з розумних сенсорів в реальному часі. Невід'ємною частиною IoT є міжмашинна взаємодія M2M – набір технологій і підходів, що забезпечують обмін інформацією між механізмами для створення розумних міст (<http://www.android.com/auto/>, <https://www.apple.com/ios/carplay/>, <http://java.dzone.com/articles/car-wars-con-cars>). Тут фігурують компанії, що надають платформи і готові апаратні рішення: Connit, Aguila Technologies, WorldSensing. Проект WorldSensing FastPrk – це сервіс розумної парковки, що дозволяє водієві за допомогою мобільного пристрою знаходити вільне місце, а міській владі ефективніше використовувати паркувальні місця. Сенсор парковки працює в частотному діапазоні Sub-GHz з радіусом 500 м.

8. Вже сьогодні автомобіль немислимий без сервісів Інтернет, орієнтованих на суміщення приємної поїздки, комфортного відпочинку і реалізації бізнес-операцій всередині машини, яка виконує роль улюбленого і єдиного місця з повним набором функціональних сервісів ділової людини (голосові: зв'язок, повідомлення та пошта, банкінг та навігація, а також музика, ігри, відео) [112-128]. Все це стало можливим завдяки підключенню автомобіля до мережі Інтернет шляхом синхронізації мобільного телефону з монітором (комп'ютером) транспортного засобу, що дає можливість аутентифікувати водія у реальному світі і віртуальному кіберпросторі. Тут лідирують компанії Apple і Google. Вони анонсували платформи зв'язку, навігації та розваг, вбудовані до автомобілю (Apple iOS CarPlay, Google Android Auto), які використовують мікропроцесорну операційну систему Blackberry QNX. Для інсталяції системи необхідно підключити смартфон Android або iOS кабелем USB до комп'ютера автомобіля. Поява на ринку автомобільних ОС дозволяє зробити висновок, що через 3-4 роки всі машини

будуть підключені до глобальних сервісів мережі Інтернет, спрямованих на підвищення якості життя людини в процесі дорожнього руху. Така домовленість є між 31 провідними автомобільними компаніями світу. Практично все, що було задекларовано в проєкті iCTC [97-99] три роки тому, поступово реалізується в сервісах Apple CarPlay, крім одного – віртуального вуличного світлофора на лобовому склі-моніторі автомобіля [129-131]. Світлофорів немає на морі і у повітрі, не буде їх і на землі через 5-10 років.

9. Ринкова та науково-технічна привабливість проєкту Smart Cloud Traffic Control підтверджується також оновленою структурою всесвітнього товариства інженерів IEEE, яке включає 38 спільнот, 9 комітетів і 7 рад, в тому числі, цікаві для проєкту: IEEE Geoscience and Remote Sensing Society, IEEE Intelligent Transportation Systems Society, IEEE Professional Communication Society, IEEE Systems, Man, and Cybernetics Society, IEEE Vehicular Technology Society, IEEE Cloud Computing Community, IEEE Electric Vehicles Community, IEEE Biometrics Council, IEEE Sensors Council.

1.6 Постановка мети і завдань дослідження

Мета дослідження – підвищення якості та безпеки дорожнього руху за рахунок створення кіберфізичної моделі комп'ютингової online взаємодії водія з хмарними сервісами керування автомобілем на основі цифрового моніторингу дорожньої інфраструктури і транспортних потоків, при використанні розумних сенсорів, засобів телекомунікації та навігації.

Задачі дослідження:

1) Розробка моделі транспортного комп'ютингу – кіберфізичної взаємодії автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

2) Створення метрики і критеріїв оцінювання якості інфраструктури для online аналізу кіберфізичного простору, пошуку квазі-оптимального маршруту і зменшення часу його виконання.

3) Розробка методу аналізу кіберфізичної інфраструктури дорожнього руху на основі алгоритму Дейкстри для online пошуку квазіоптимального маршруту в умовах виникнення колізій.

4) Створення архітектури розумного хмарного світлофора на основі використання логічних операцій, що дає можливість істотно збільшити пропускну здатність перехрестя доріг.

5) Практична реалізація моделей і методів цифрового моніторингу і хмарного керування автомобілем в рамках створення кіберфізичної системи дорожнього руху та подальше їх тестування.

Сутність дослідження – створення кіберфізичної інфраструктури комфортного безпечного дорожнього руху за рахунок поступового перенесення дорожніх знаків, світлофорів у кіберпростір і точного позиціонування транспорту з метою хмарного online керування автомобілем на основі розумної суперпозиції замовленого маршруту з оперативним цифровим моніторингом дорожньої обстановки, яка відображається на сенсорному дисплеї автомобіля.

Впровадження хмарного сервісу управління транспортом призведе до збереження екології планети і до зменшення: 1) часу проходження маршрутів, 2) споживання енергетичних ресурсів і матеріальних витрат на створення світлофорів, дорожніх знаків, автомобільних номерів, 3) витрат на дорожню поліцію і експлуатацію інфраструктури, 4) числа аварій і крадіжок автомобілів.

Об'єкт дослідження – процеси цифрового моніторингу та кіберуправління розумним автомобілем у масштабованій дорожній кіберфізичній інфраструктурі, на якій позиціонуються маршрути руху всіх транспортних засобів при виконанні замовлених маршрутів.

Предмет дослідження – кіберфізична система комп'ютирової взаємодії хмарних сервісів управління транспортом та інфраструктурою з розумними засобами цифрового моніторингу дорожньої обстановки з метою online оптимального управління автомобілем для виконання замовленого маршруту.

Функція мети, що формує якість інфраструктури, визначається інтегральним критерієм ефективності хмарної системи моніторингу та управління дорожнім рухом, що складається з трьох суперечливих параметрів: рівень помилкових рішень L , що призводять до колізій; Y – параметр якості обслуговування, що зменшує до нуля час вимушених простоїв транспортних засобів T . Критерій також враховує рівень додаткових витрат або структурну складність H^a на створення і експлуатацію кіберфізичної інфраструктури H у порівнянні з базовою H^s , що реалізують механізми спостереження і керування транспортними потоками:

$$E = F(L, T, H) = \min\left[\frac{1}{3}(L + T + H)\right],$$

$$Y = (1 - P)^n;$$

$$L = 1 - Y^{(1-k)} = 1 - (1 - P)^{n(1-k)};$$

$$T = \frac{(1-k) \times H^s}{H^s + H^a}; \quad H = \frac{H^a}{H^s + H^a}.$$

Тут k і P є прохідність інфраструктури та ймовірність виникнення колізій відповідно, n – число прихованих помилок інфраструктури. Час пропуску заданого транспортного потоку через фрагмент інфраструктури визначається її прохідністю k , помноженою на структурну складність дорожньої функціональності H^s , віднесеної до загальної складності інфраструктури $H^s + H^a$. Рівень технічної надмірності H знаходиться у функціональній залежності від складності механізмів прохідності, віднесеної до загальної складності заданої інфраструктури (дорожня функціональність + інфраструктурні механізми прохідності). При цьому компонент структурної

надлишковості прохідності, а саме, спостережуваність інфраструктури забезпечує задану глибину діагностування пробок або транспортних колізій на дорожній структурі. У той час як керованість інфраструктури покликана усувати колізії інтелектом хмарних сервісів і світлофорів у межах заданого часового інтервалу.

Локальна функція мети хмарного сервісу для водія транспортного засобу визначається максимальним значенням функціоналу ($Q=1$), який складений з трьох суперечливих і безрозмірних параметрів метрики, наведеної до інтервалу $(0,1)$: час $T(X)$ – гроші $M(X)$ – якість $Y(X)$:

$$Q = \frac{1}{3}[T(X) + M(X) + Y(X)],$$

$$T(X) = \frac{1}{2} \left[\frac{V^r}{V} + \frac{T^m(X)}{T^r(X)} \right];$$

$$M(X) = \frac{M}{M + M^\#(X) + M^p + M^r};$$

$$Y(X) = [1 - P(X)]^n.$$

Тут кожен параметр істотно залежить від зеленого сигналу світлофора (X) на маршруті, який доставляє середнє арифметичне значення функціоналу за відсутністю інших колізій. Інші параметри позначають: $[V, V^r, T^r(X), T^m(X), M, M^\#(X), M^p, M^r, P(X), n]$ – модельну і реальну швидкість автомобіля на маршруті; реальний і модельний час очікування на перехресті; мінімальні грошові витрати на проходження маршруту без колізій; витрати, пов'язані з простоем на перехресті; штрафи за порушення правил дорожнього руху; непередбачені ремонти; кількість перехресть (X) на маршруті руху транспортного засобу.

Таким чином, за відсутністю світлофорів і транспортних колізій якість (життя водія) виконання маршруту автомобілем дорівнює одиниці. За їх наявністю використовується запропонована метрика з метою зменшення часу простою транспортного засобу на перехресті.

Основні результати розділу опубліковані в роботах [142, 144, 153].

2 АНАЛІЗ ЯКОСТІ ТОПОЛОГІЧНИХ СТРУКТУР ДОРОЖНЬОГО РУХУ

Пропонуються критерії оцінювання якості дорожньої інфраструктури, наведені до аналізу топології на графах, які визначають середню довжину шляху між двома точками в обмеженому просторі. На прикладах графових структур оцінюються різні топології з метою їх можливого поліпшення. Представлені метрики і структури для вибору кращого рішення (маршрут руху, управління світлофором).

Мета розділу – розробка метрики, критеріїв оцінювання маршрутів руху і дорожніх інфраструктур для реалізації хмарних сервісів управління дорожнім рухом, що надаються водієві у режимі реального часу. Завдання дослідження: 1) Створення критеріїв оцінювання якості інфраструктури дорожнього руху. 2) Розробка аналітичних моделей прохідності (спостереження та управління) дорожньої інфраструктури. 3) Пропозиція метрики для вимірювання е-інфраструктури дорожнього руху. 4) Аналіз різних топологій і зв'язків компонентів інфраструктури. 5) Створення спрощених критеріїв оцінювання якості топології дорожньої інфраструктури. 6) Створення наведеної метрики вимірювання топології графа дорожньої інфраструктури.

2.1 Критерії якості інфраструктури дорожнього руху

Критерії якості топології доріг пов'язані зі статичним аналізом графової структури, що має E дуг і n вершин. Особливість першого з них полягає в обчисленні абсолютного і не наведеного до одиничного інтервалу значення, яке формується вартістю з'єднань, помноженої на якість транзакцій між усіма парами вершин:

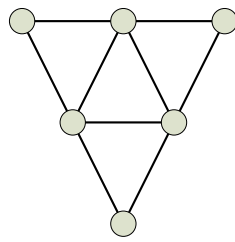
$$Q_1 = \frac{E}{n} \times \sum_{i=1}^n \min_j (p_{ij}) .$$

Застосування цієї формули до оцінювання трьох графових структур (рис. 2.1), що мають 6 вершин і різні топології з'єднання, представлені нижче:

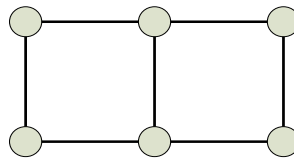
	a	b	c	d	e	f
a	.	1	1	.	.	.
b	1	.	1	1	.	1
c	.	1	.	.	.	1
d	1	1	.	.	1	1
e	.	.	.	1	.	1
f	.	1	1	1	1	.

	a	b	c	d	e	f
a	.	1	.	1	.	.
b	1	.	1	.	1	.
c	.	1	.	.	.	1
d	1	.	.	.	1	.
e	.	1	.	1	.	1
f	.	.	1	.	1	.

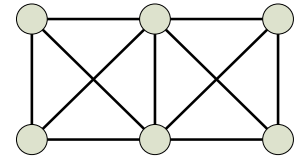
	a	b	c	d	e	f
a	.	1	.	1	1	.
b	1	.	1	1	1	1
c	.	1	.	.	1	1
d	1	1	.	.	1	.
e	1	1	1	1	.	1
f	.	1	1	.	1	.



а



б



в

Рис. 2.1 – Структури сполук графових вершин

Тут три графа мають 9, 7 і 11 дуг відповідно. Підрахунок критерію відповідно до останньої формулою дає наступні результати:

$$Q_1(G_1) = \frac{E}{n} \times \sum_{i=1}^n \min(p_{ij}) = \frac{9}{6} \times (9 \times 1 + 6 \times 2) = 31,5;$$

$$Q_1(G_2) = \frac{7}{6} \times (7 \times 1 + 6 \times 2 + 2 \times 3) = 29,2;$$

$$Q_1(G_3) = \frac{11}{6} \times (11 \times 1 + 4 \times 2) = 34,8.$$

Модифікація оцінки ефективності топології пов'язана з формуванням інтервальної оцінки $(0,0 - 1,0)$ шляхом приведення реальних витрат (число дуг

E) до максимально можливої кількості $V = \frac{n^2 - n}{2}$ парних з'єднань в графі $\frac{E}{V}$,

що забезпечує якість комунікаційних властивостей $\frac{V}{\sum_{i=1}^n \min(p_{ij})}$:

$$Q_2 = \frac{E}{\frac{n^2 - n}{2}} \times \frac{\frac{n^2 - n}{2}}{\sum_{i=1}^2 \sum_j \min(p_{ij})} = \frac{E}{\frac{n^2 - n}{2}} = \frac{E}{\sum_{i=1}^2 \sum_j \min(p_{ij})}.$$

Оцінка має одиничне значення, якщо чисельник і знаменник дорівнюють $V = \frac{n^2 - n}{2}$. В цьому випадку графова структура має всі можливі парні сполуки між вершинами графа, які визначаються половиною декартова квадрата потужності безлічі вершин мінус n вершин. Значення від'ємника формується відсутністю ідемпотентна замикань всіх вершин графа. При цьому кожна пара вершин має довжину шляху, що дорівнює одиниці. Якщо структура представлена неорієнтованим графом, то потужність всіх пар вершин подвоюється $V = n^2 - n$. Перерахунок критеріїв ефективності міжз'єднань всіх пар вершин, представлених на останньому рисунку, дає наступний результат:

$$Q_2(G_1) = \frac{E}{\sum_{i=1}^{V=12} \sum_j \min(p_{ij})} = \frac{9}{9 \times 1 + 6 \times 2} = 0,428;$$

$$Q_2(G_2) = \frac{7}{7 \times 1 + 6 \times 2 + 2 \times 3} = 0,28;$$

$$Q_2(G_3) = \frac{11}{11 \times 1 + 4 \times 2} = 0,578.$$

Збільшення числа зв'язків (чисельника) призводить до зменшення суми довжин шляхів між усіма парами вершин (знаменника). У межі для сильного графа значення дроби дорівнює одиниці. Найкраща якість інфраструктури. При цьому платою за якість комунікацій є прихована від експерта потужність з'єднань, приведена до максимально можливої кількості ребер:

$$H_2(G_1) = \frac{E}{\frac{n^2 - n}{2}} = \frac{9}{15} = 0,60;$$

$$H_2(G_2) = \frac{7}{15} = 0,46;$$

$$H_2(G_3) = \frac{11}{15} = 0,73.$$

Таким чином, мультиплікативність двох критеріїв: складності графа і якості парних комунікацій створюють структурну невизначеність, яка приховує вартість з'єднань і якість комунікацій. Дану ситуацію можна вирішити шляхом формування структурної оцінки, яка містить критерій складності комунікаційної інфраструктури, що формує інтервальні витрати часу – середню досяжність між кожною парою вершин графової структури, приведену до максимально можливої довжини з'єднань. Це демонструє вартість якості інфраструктури, що має цільову функцію, яка мінімізує середню досяжність (довжину шляху або часу) між парою вершин графової структури:

$$Q_3 = \frac{E}{V} \times \left[\frac{1}{n-1} \times \frac{1}{V} \times \sum_{i=1}^V \min_j(p_{ij}) \right] = \frac{E}{V^2(n-1)} \times \sum_{i=1}^V \min_j(p_{ij})$$

$$Q_3(G_1) = \frac{9}{15} \times \left(\frac{21}{15 \times 5} \right) = 0,168;$$

$$Q_3(G_2) = \frac{7}{15} \times \left(\frac{25}{15 \times 5} \right) = 0,155;$$

$$Q_3(G_3) = \frac{11}{15} \times \left(\frac{19}{15 \times 5} \right) = 0,185.$$

В останній оцінці існує проблема отримання мінімального значення добутку, якій менше меншого множника. Для того, щоб інтервальна оцінка була більш виваженою і значущою в діапазоні від 0 до 1, її необхідно зробити адитивною по відношенню до середнього значення двох компонентів, що складають критерій:

$$Q_4 = \frac{1}{2} \times \left[\frac{E}{V} + V \times \left(\sum_{i=1}^V \min(p_{ij}) \right)^{-1} \right],$$

$$Q_4(G_1) = \frac{1}{2} \times \left[\frac{9}{15} + \frac{15}{21} \right] = \frac{1}{2} \times [0,6 + 0,71] = 0,655;$$

$$Q_4(G_2) = \frac{1}{2} \times \left[\frac{7}{15} + \frac{15}{25} \right] = \frac{1}{2} \times [0,467 + 0,60] = 0,504;$$

$$Q_4(G_3) = \frac{1}{2} \times \left[\frac{11}{15} + \frac{15}{19} \right] = \frac{1}{2} \times [0,73 + 0,79] = 0,751;$$

Тут перший компонент – інфраструктурна складність (вартість проекту) – наближається до одиниці за мірою збільшення числа ребер в графі, максимальна кількість яких дорівнює $V = (n^2 - n) / 2$. Другий компонент – інфраструктурна якість – наближається до одиничного значення лише в тому випадку, якщо нові зв'язки забезпечують зменшення довжини шляхів між парами вершин графа. Оцінка може мати одиничну складність проекту або максимальне число з'єднань, але при цьому не досягти одиничної якості, якщо зв'язку невірно розставлені. Цим пояснюється обов'язкова наявність двох компонентів структурного критерію, які одночасно здатні оцінити складність і якість інфраструктури. Оцінка буде максимальною і дорівнює 1 при побудові сильного графа, де кожна пара вершин стає суміжній:

$$Q_4(G_1) = \frac{1}{2} \times \left[\frac{15}{15} + \frac{15}{15} \right] = 1,0;$$

$$Q_4(G_2) = \frac{1}{2} \times \left[\frac{15}{15} + \frac{15}{15} \right] = 1,0;$$

$$Q_4(G_3) = \frac{1}{2} \times \left[\frac{15}{15} + \frac{15}{15} \right] = 1,0;$$

Цікаво побудувати критерій відношення складності до якості, який буде характеризувати властивість – якою ціною досягається якість інфраструктури:

$$Q_5 = \frac{\frac{E}{V}}{V \times \left(\sum_{i=1}^V \min_j(p_{ij}) \right)^{-1}} = \frac{E}{V^2} \times \sum_{i=1}^V \min_j(p_{ij})$$

$$\begin{aligned} Q_5(G_1) &= \left[\frac{15}{225} \times 15 \right] = 1,0; & Q_5(G_1) &= \left[\frac{9}{225} \times 21 \right] = 0,84; \\ Q_5(G_2) &= \left[\frac{15}{225} \times 15 \right] = 1,0; & Q_5(G_2) &= \left[\frac{7}{225} \times 25 \right] = 0,778; \\ Q_5(G_3) &= \left[\frac{15}{225} \times 15 \right] = 1,0; & Q_5(G_3) &= \left[\frac{11}{225} \times 19 \right] = 0,93; \end{aligned}$$

Критерій знаходиться в інтервалі від 0 до 1 і показує ефективність (яка наближається до одиниці) введення нових зв'язків для зменшення середньої довжини шляху між парою вершин інфраструктури. В чисельник дробу виходить твір складності на якість, віднесене до максимального значення компонентів даного твору, який має місце бути для сильного графа. Як правило, підвищення числа зв'язків призводить до зменшення суми довжин шляхів між усіма парами вершин інфраструктури. Останній критерій

$$Q_5 = \frac{E}{V^2} \times \sum_{i=1}^V \min_j(p_{ij})$$

цікавий не тільки інтегральним значенням ефективності, але і структурними компонентами, які являють собою оцінки складності і якості інфраструктурного рішення для оптимального вибору в умовах обмежень. Для практичного використання будь-якого з п'яти критеріїв ефективності необхідно мати швидкодіючий паралельний алгоритм (Дейкстра) обчислення довжин шляхів між усіма парами вершин графа. Необхідно також шукати більш прості оцінки ефективності графових структур, де замість суми всіх пар мінімальних шляхів буде фігурувати обчислювально нескладний компонент, що корелюється зі згаданою сумою.

2.2 Функції прохідності дорожньої інфраструктури

Інший цікавий напрямок метрики вимірювання і оцінювання інфраструктури системи моніторингу і управління пов'язано з поняттям прохідності або тестопригодності (trafficability), що включає два критерії: керованість і спостережуваність, які дають можливість оцінювати як систему в цілому, так і якість руху автомобіля за заданим маршрутом в режимі on-line (в реальному масштабі часу враховувати транспортні потоки). Існуючі стандарти тестопригодного проектування комп'ютерних систем і мереж (IEEE 11.49, IEEE 1500) можна адаптувати для проектування, тестування і верифікації інфраструктури дорожнього руху з метою її подальшої ефективної експлуатації.

Досить істотна надмірність інфраструктури дорожнього руху передбачає її ефективне використання з метою підвищення її прохідності. Для цього використовується (орієнтований) граф транспортних потоків (ГТП), який надає користувачеві інформацію про взаємозв'язки доріг і перехресть. Для інтегрального оцінювання прохідності Q вводяться наступні критерії:

$$Q = \frac{Z(S)}{Z(S) + Z(F) + Z(B) + Z(A)} \times \frac{1}{n} \sum_{i=1}^n (U_i \times N_i);$$

$$U_i = \frac{1}{T} \sum_{j=1}^{x_i} T_j^i \times \frac{1}{d_i^x \vee t_i^x}; N_i = \frac{1}{T} \sum_{j=1}^{y_i} T_j^i \times \frac{1}{d_i^y \vee t_i^y}.$$

де n – кількість вершин графа. Керованість і спостережуваність є метрикою оцінювання прохідності інфраструктури дорожнього руху. Керованість вершини (перехрестя) має функціональну залежність від структурної глибини d_i^x знаходження вершини щодо пункту відправлення або довжини кон'юнктивного терма – t_i^x . Спостережуваність вершини має аналогічну залежність $d_i^y \vee t_i^y$ щодо пункту прибуття. Для підрахунку прохідності можна використовувати один з параметрів $d_i^x, t_i^x (d_i^y, t_i^y)$. Оцінки управління та

спостереження залежать також від процентного відношення числа вхідних дуг

$\frac{1}{T} \sum_{j=1}^{x_i} T_j^i$ (вихідних $\frac{1}{T} \sum_{j=1}^{y_i} T_j^i$) розглядуваної вершини до загальної кількості дуг T ,

де X_i – кількість дуг, що формують доступ до вхідних вершині; (Y_i) – число дуг, що визначають вершину як джерело. Прохідність Q залежить від керованості U , спостережливості (N), а також від вартості реалізації (Z) компонентів інфраструктури, тут визначена метрика структурного покриття (F), транспортні потоки (B), механізм спостереження (A), функціональність (S). Керованість (спостережуваність) є функція від числа дуг, що входять до вершини (що виходять з вершини) графа, а також від структурної глибини вершини (перехрестя) – відстані від пункту відправлення (прибуття). Наведений критерій прохідності може бути також використаний і для оцінки якості граф-схеми або інфраструктури управління дорожнього руху. Тут розглядаються вершини, навантажені вхідними умовами, а також позиція вершини по відношенню до початку або закінчення запланованого маршруту. Кількість умов або вхідних дуг в кожній вершині, об'єднане операцією Or , впливає на прохідність графа в частині керованості. Аналогічно обчислюється спостережуваність, на яку впливає структурна глибина і потужність вихідних (дуг) умов, створювана операціями And , Or . Таким чином, прохідність даної вершини орієнтованого графа може бути представлена логічною функцією, заданою у вигляді кон'юнктивної нормальної форми (КНФ). При цьому керованість і спостережуваність буде визначатися оцінкою за Квайном обчислювальної складності КНФ. У загальному випадку логічні функції спостереження та управління вершини графа задаються кон'юнкцією діз'юнктивних умов (перший рядок):

$$1) U_r^f = \bigwedge_{i=1}^{n_r^x} \left(\bigvee_{j=1}^{x_i} T_{rij}^x \right); N_r^f = \bigwedge_{i=1}^{n_r^y} \left(\bigvee_{j=1}^{y_i} T_{rij}^y \right);$$

$$2) U_r^f = \bigvee_{i=1}^{x_r} \left(\bigwedge_{j=1}^{n_i^x} T_{ij}^x \right); N_r^f = \bigvee_{i=1}^{y_r} \left(\bigwedge_{j=1}^{n_i^y} T_{ij}^y \right).$$

Тут функція керованості U_i^f (спостережуваності N_i^f) визначається кон'юнкцією всіх вершин-попередників n_i^x (наступників n_i^y), Де кожна з них має X_i дуг, що входять (виходять Y_i), з'єднаних знаками диз'юнкції. Потужність диз'юнктивних термів відповідає кількості вхідних до вершини дуг, а число кон'юнкцій є структурна глибина розташування розглянутого компонента в графі. Далі кон'юнктивна форма перетвориться до виду ДНФ – другий рядок в попередньому виразі, де число термів для функції керованості (спостережуваності) $x_i(y_i)$ дорівнює всім можливим шляхам формування стану розглянутої вершини, а довжина терма керованості (спостережливості) $n_i^x(n_i^y)$ є умова досяжності вершини – структурна глибина від входів (виходів).

Цікавим видається нестандартне рішення, коли критерії спостережуваності та управління поточної вершини транзакційного графа обчислюються на підставі моделі у вигляді побудованих логічних функцій спостережуваності та управління (U_i, N_i) і інтегральної оцінки прохідності (Q) при використанні апарату – алгебраїчної форми подання графа. Формули підрахунку, які передбачають облікові оцінки, мають такий вигляд:

$$U_i = \frac{1}{t_{\max}^x \times n_t^x} \times \sum_{i=1}^{n_t^x} \sum_{j=1}^{k_i^x} (t_{\max}^x - |t_{ij}^x| + 1);$$

$$N_i = \frac{1}{t_{\max}^y \times n_t^y} \times \sum_{i=1}^{n_t^y} \sum_{j=1}^{k_i^y} (t_{\max}^y - |t_{ij}^y| + 1);$$

$$Q = \frac{1}{n} \sum_{i=1}^n (U_i \times N_i),$$

де $t_{\max}^x, n_t^x, k_i^x, |t_{ij}^x|$ – кон'юнктивний терм максимальної довжини для визначення критерію керованості; кількість термів в логічній функції керованості; кількість букв (змінних або дуг) в поточному термі функції;

потужність розглядуваного терма. Аналогічні позначення використовуються і при підрахунку критерію спостережуваності – $t_{\max}^y, n_t^y, k_i^y, |t_{ij}^y|$ кожної вершини графа.

Для фрагмента графової структури, представленої на рис. 2.2,

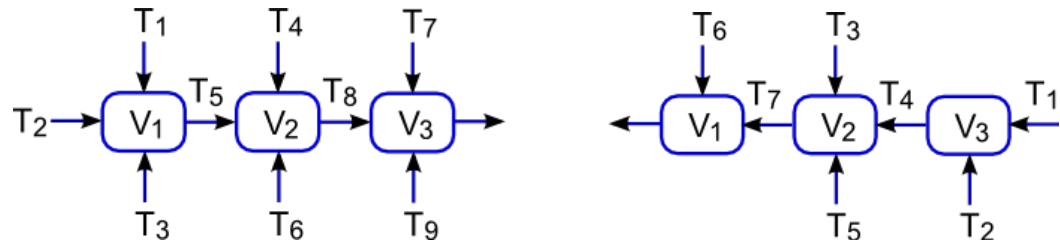


Рис. 2.2 – Графи для підрахунку спостереження та управління

перетворення кон'юнктивної форми у діз'юнктивну модель щодо вершин $V_1 - V_3$ за заданими вище правилами формує такі логічні функції керованості:

$$U^f(V_1) = T_1 \vee T_2 \vee T_3;$$

$$U^f(V_2) = (T_1 \vee T_2 \vee T_3)T_5 \vee T_4 \vee T_6 = T_1T_5 \vee T_2T_5 \vee T_3T_5 \vee T_4 \vee T_6;$$

$$U^f(V_3) = (T_1 \vee T_2 \vee T_3)T_5T_8 \vee (T_4 \vee T_6)T_8 \vee (T_7 \vee T_9) = \\ = T_1T_5T_8 \vee T_2T_5T_8 \vee T_3T_5T_8 \vee T_4T_8 \vee T_6T_8 \vee T_7 \vee T_9.$$

Дотримуючись виразу, який визначає процедури підрахунку оцінок, можна визначити керованість компонента V_3 :

$$U(V_3) = \frac{1}{t_{\max}^x \times n_t^x} \times \sum_{i=1}^{n_t^x} \sum_{j=1}^{k_i^x} (t_{\max}^x - |t_{ij}^x| + 1) = \frac{1}{3 \times 7} \times (1 + 1 + 1 + 2 + 2 + 3 + 3) = 0,61.$$

Для другого фрагмента графа, представлено на рис. 2.2, перетворення кон'юнктивної форми у діз'юнктивну структуру дозволяє визначити логічну функцію і чисельне значення спостережуваності для вершини V_1 :

$$N^f(V_1) = T_6 \vee T_7(T_3 \vee T_5 \vee T_4(T_1 \vee T_2)) = T_6 \vee T_7T_3 \vee T_7T_5 \vee T_7T_4T_1 \vee T_7T_4T_2;$$

$$N^f(V_2) = T_3 \vee T_5 \vee T_4(T_1 \vee T_2) = T_3 \vee T_5 \vee T_4T_1 \vee T_4T_2; V_3 = T_1 \vee T_2.$$

$$N(V_1) = \frac{1}{t_{\max}^y \times n_t^y} \times \sum_{i=1}^{n_t^y} \sum_{j=1}^{k_i^y} (t_{\max}^y - |t_{ij}^y| + 1) = \frac{1}{3 \times 5} \times (3 + 2 + 2 + 1 + 1) = \frac{9}{15} = 0,6.$$

Для випадку, коли дуги в графі мають вагові коефіцієнти $b_{ij}^x(b_{ij}^y)$, що показують чисельне значення транспортних потоків, задіяних у смугах руху між вершинами (мультидуги), формули підрахунку прохідності суттєво ускладнюються:

$$U_i = \frac{\sum_{i=1}^{n_t^x} \prod_{j=1}^{k_i^x} b_{ij}^x (t_{\max}^x - |t_{ij}^x| + 1)}{t_{\max}^x \times (\sum_{i=1}^{n_t^x} \prod_{j=1}^{k_i^x} b_{ij}^x)}; N_i = \frac{\sum_{i=1}^{n_t^y} \prod_{j=1}^{k_i^y} b_{ij}^y (t_{\max}^y - |t_{ij}^y| + 1)}{t_{\max}^y \times (\sum_{i=1}^{n_t^y} \prod_{j=1}^{k_i^y} b_{ij}^y)}$$

Синтезовані логічні функції моделі дорожнього руху задають всі можливі шляхи управління, як в часі, так і у просторі, що можна вважати новою аналітичною формою опису прохідності інфраструктури. За ДНФ можна визначити критерії керованості (спостережуваності) для всіх компонентів граф-моделі. Тут розглядаються два варіанти (сценарії) обрахунку інфраструктури. 1) Чи враховується тільки графова модель, де вага кожної дуги дорівнює 1, незалежно від величини транспортного потоку. 2) Всі дуги графа відзначаються реальною кількістю транспортних транзакцій, що мають місце бути між двома вершинами графа. Оцінки прохідності описаних варіантів можуть застосувати перший сценарій, або мати більш складну і точну модель транспортних транзакцій, розподілених у часі, на множині графових компонентів.

Результат обчислення керованості (спостережуваності) для всіх вершин графа дає можливість отримати графік, який дозволяє визначити критичні

вершини для установки необхідних точок управління (світлофорів) та спостереження (моніторів). Після визначення спостереження та управління вершин транзакційного графа виконується підрахунок узагальненого критерію прохідності аналізованої інфраструктури. Вона характеризує якість проектного варіанту, що є досить суттєвим при порівнянні декількох альтернативних рішень.

2.3 Метрика e-інфраструктури дорожнього руху

Для ідентифікації або розпізнавання компонентів, об'єктів, процесів або явищ в кіберпросторі необхідно мати ефективну метрику вимірювання відстаней в дискретно булевом (векторно-логічному) просторі, на основі якої будуються швидкодіючі механізми, що визначають похідну, ступінь подібності або відмінності між ними.

Дискретний векторно-логічний простір (кіберпростір) – сукупність взаємодіючих за відповідною метрикою інформаційних процесів і явищ, що описуються векторами логічних змінних і використовують як носій пам'яті.

Метрика – спосіб вимірювання відстані у просторі між компонентами процесів або явищ, описаних векторами логічних змінних. Відстань у кіберпросторі – це хог-відношення між парою векторів, які позначають компоненти процесу або явища, що відрізняє його від кодової відстані за Хеммінгом. Відстань, похідна (булева), ступінь зміни, відмінності або близькості є ізоморфні поняття, пов'язані з визначенням відношення двох компонентів процесу або явища. Поняття відстані між компонентами у кіберпросторі є міра їх відмінності. Процедури порівняння, вимірювання, оцінювання, розпізнавання, тестування, діагностування, ідентифікування є спосіб визначення ставлення при наявності не менше ніж одного об'єкта.

Компонент простору представлений k -мірним (двійковим) вектором $a = (a_1, \dots, a_j, \dots, a_k), a_j \in \{0,1\}, k = 1,2,3, \dots$, де кожна його координата визначена у

двійковому алфавіті, 0 – «неправда», 1 – «істина». Нуль-вектор є k -мірний кортеж, всі координати якого дорівнюють нулю: $\mathbf{a}_j = 0, j = \overline{1, k}$.

Метрика β кіберпростору визначається єдиною рівністю, яка формує нуль-вектор для хог-суми відстаней \mathbf{d}_i між ненульовим і кінцевим числом точок (об'єктів), замкнених в цикл:

$$\beta = \bigoplus_{i=1}^n \mathbf{d}_i = 0,$$

де n – кількість (ціле число) відстаней між компонентами (векторами) простору, складовими цикл $D = (\mathbf{d}_1, \dots, \mathbf{d}_1, \dots, \mathbf{d}_n)$, \mathbf{d}_i – є вектор відстані, відповідний ребру циклу, що з'єднує два компонента (вектора) \mathbf{a} , \mathbf{b} простору, який далі позначається без індексу як $\mathbf{d}(\mathbf{a}, \mathbf{b})$. Відстань між двома об'єктами (векторами) \mathbf{a} і \mathbf{b} є похідний вектор: $d(\mathbf{a}, \mathbf{b}) = (\mathbf{a}_j \oplus \mathbf{b}_j)_1^k$. Векторному значенн. відстані відповідає норма – скалярні відстань за Хеммінгом між двома векторами – як число одиниць вектора $\mathbf{d}(\mathbf{a}, \mathbf{b})$. Інакше: метрика β векторного логічного двійкового простору є рівна нуль-вектору хог-сума відстаней між кінцевим числом точок (вершин) графа, що утворюють цикл. Сума n -мірних двійкових векторів, які задають координати точок циклічної фігури, дорівнює нуль-вектору. На основі введеної метрики можна дати більш формальне визначення кіберпростору, який є векторно-логічним, нормованим β -метрикою, де хог-сума відстаней між кінцевим числом точок циклу дорівнює нуль-вектору. Визначення метрики ставить на перше місце не елементи множини, але відношення, що дозволяє скоротити систему аксіом (тотожності, симетрії і транзитивного трикутного замикання) з трьох до одного і поширити її дію на як завгодно складні конструкції n -мірного логічного простору. Класичне завдання метрики для визначення взаємодії однієї, двох і трьох точок у векторному логічному просторі є окремим випадком β -метрики при $i = 1, 2, 3$:

$$M = \begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a, b) = d(b, a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a, b) \oplus d(b, c) = d(a, c). \end{cases}$$

Векторно-логічний транзитивний трикутник має повну аналогію чисельному вимірі відстані в метричному М-просторі, яке задається системою аксіом, що визначає взаємодію однієї, двох і трьох точок в будь-якому просторі:

$$M = \begin{cases} d(a, b) = 0 \leftrightarrow a = b; \\ d(a, b) = d(b, a); \\ d(a, b) + d(b, c) \geq d(a, c). \end{cases}$$

Специфіка аксіоми трикутника метричного простору полягає у чисельному (скалярному) порівнянні відстаней трьох об'єктів, коли інтервальна невизначеність відповіді – дві сторони трикутника можуть бути більше або рівні третій – малоприсадна для визначення точної довжини останньої сторони. Векторно-логічний простір усуває даний недолік, повністю виключає ступінь невизначеності у бінарному відношенні детермінованих станів процесів або явищ. В цьому випадку чисельна невизначеність третьої сторони трикутника у векторному логічному просторі набуває форму точного двійкового вектора, який характеризує відстань між двома об'єктами і обчислюється на основі знання відстаней двох інших сторін трикутника: $d(a, b) \oplus d(b, c) = d(a, c) \rightarrow d(a, b) \oplus d(b, c) \oplus d(a, c) = 0$.

Метрика β багатозначного векторно-логічного кіберпростору є вектор, що дорівнює значенню Δ за всіма координатами, отриманий шляхом застосування симетричної різниці відстаней між кінцевим числом точок, що утворюють цикл: $\beta = \bigtriangleup_{i=1}^n d_i = \emptyset$, де кожна координата вектора, що відповідає об'єкту, визначена в алфавіті, який становить булеан на універсумі примітивів потужністю p : $\mathbf{a}_j = \{\alpha_1, \dots, \alpha_r, \dots, \alpha_m\}$, $\mathbf{m} = 2^p$:

$$a \Delta b = \{0 \Delta 0 = \square, 0 \Delta 1 = x, 0 \Delta x = 1, 0 \Delta \square = 0, 1 \Delta 1 = \square, \\ 1 \Delta x = 0, 1 \Delta \square = 1, x \Delta x = \square, x \Delta \square = x, \square \Delta \square = \square\}.$$

Рівність порожньому вектору симетричної різниці по координатній теоретико-множинній взаємодії підкреслює рівнозначність компонентів (відстаней), що беруть участь у формуванні рівняння, де єдина координатна операція $d_{i,j} \Delta d_{i+1,j}$ визначена у чотиризначній моделі Кантора $A = \{0, 1, x, \emptyset\}$, $x = \{0, 1\}$. Замкнутість будь-якого теоретико-множинного алфавіту можна перевіряти за допомогою Δ -операції (багатозначний аналог хог-операції), коли Δ -сума всіх символів повинна дорівнювати пустій множині: $0 \Delta 1 \Delta x \Delta \emptyset = \emptyset$. Якщо кодувати дану процедуру двійковими кодами-векторами, то хог-сума буде дорівнювати нуль вектору: $01 \oplus 10 \oplus 11 \oplus 00 = 00$. Для практичного використання введеної метрики кіберпростору далі пропонується доказовий перехід від чисельної характеристики бінарного відношення об'єктів, що об'єднує три скалярні оцінки їх взаємодії, до чисто векторно-логічного критерію якості відношень двох об'єктів.

Нехай вхідний вектор $m = (m_1, \dots, m_j, \dots, m_k)$, $m_j \in \{0, 1, x\}$ і аналізований об'єкт $A = (A_1, \dots, A_j, \dots, A_k)$, $A_j \in \{0, 1, x\}$, який також представлений вектором, мають однакову розмірність k . Ступінь приналежності m -вектора до A позначається як $\mu(m \in A)$.

Визначення. Інтегральна теоретико-множинна метрика для оцінювання якості запиту (послуги) є функція взаємодії багатозначних за координатами векторів $m \Delta A$, яка визначається середньою сумою трьох параметрів: кодова відстань $d(m, A)$, функція приналежності $\mu(m \in A)$ і функція приналежності $\mu(A \in m)$:

$$Q = \frac{1}{3} [d(m, A) + \mu(m \in A) + \mu(A \in m)],$$

$$d(m, A) = \frac{1}{n} [n - \text{card}[(i : m_i \cap A_i = \emptyset, i = 1, \dots, k)]]; \mu(m \in A) = 2^{c-a}; \mu(A \in m) = 2^{c-b};$$

$$a = \text{card}[(i : A_i = x, i = 1, \dots, k)]; b = \text{card}[(i : m_i = x, i = 1, \dots, k)]; c = \text{card}[(i : m_i \cap A_i = x, i = 1, \dots, k)];$$

Перетин (об'єднання) векторів – є векторна операція, заснована на відповідних координатних теоретико-множинних операціях. Операції координатного перетину і об'єднання визначені в алфавіті Кантора $A = \{0,1, x = \{0,1\}, \emptyset\}$. Нормування параметрів дозволяє оцінити рівень взаємодії векторів в чисельному інтервалі $[0,1]$.

Процедура обчислення векторного критерію якості залежить від значності алфавіту:

$$Q' = \begin{cases} m \oplus A \leftarrow \{m_i, A_i\} \in \{0,1\}; \\ m \Delta A \leftarrow \{m_i, A_i\} \in \{0,1, x\}. \end{cases}$$

Критерій якості Q однозначно визначає три форми взаємодії двох будь-яких об'єктів у n -вимірному векторному логічному просторі: відстань і дві функції приналежності. Процес-модель пошуку рішення з мінімальним числом одиничних координат з більш, ніж двох альтернатив, представлена на рис. 2.3. Вона включає наступні операції: 1) Спочатку до вектору-результату Q , у якому буде збережено краще рішення, заносяться одиничні значення в усі координати (найгірше рішення) і одночасно здійснюється операція slc зсуву вліво з ущільненням одиниць поточного вектора Q_i . 2) Виконується порівняння двох векторів: Q і чергової оцінки Q_i зі списку рішень. 3) Реалізується векторна операція and ($Q \wedge Q_i$), а результат порівнюється з вектором Q , що дає можливість змінити його, якщо вектор Q_i має менше число одиничних значень. 4) Процедура пошуку оцінки кращого рішення повторюється n раз.

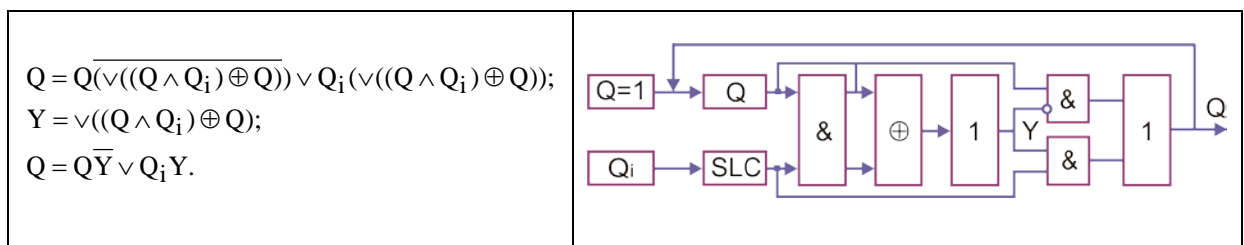


Рис. 2.3 – Процес-модель вибору рішення

Тут og-оператор редукції (після хог елемента) формує бінарне однобітове рішення на основі логічної операції og над n розрядами критерію якості. Таким

чином, векторний логічний критерій якості взаємодії об'єктів у кіберпросторі дозволяє отримувати оцінку рішення з високою швидкістю логічних паралельних операцій, що особливо істотно для критичних хмарних систем управління трафіком у реальному часі.

2.4 Топології зв'язків компонентів системи

Топологія зв'язків або відношень між компонентами різної природи формує різноманіття системоутворюючих структур. При цьому оптимізація числа компонентів або кількості зв'язків стає критично важливою для функціонування системи, що має на меті і обмеження. Далі просувається ідея, що трикутні примітиви створюють ефективні транзакційні структури. Тріада компонентів домінує в фізичній та соціальній природі, як одна з найпоширеніших структур, що має властивості: 1) Транзитивність трикутного замикання є основою будь-якої метрики вимірювання процесів і явищ. 2) Трикутник – найпримітивніша геометрична фігура на площині, а тетраедр, складений з трикутних площин, в просторі. 3) Всі геометричні фігури на площині можуть бути отримані за допомогою суперпозиції трикутних примітивів, а в просторі – шляхом використання тетраедрів. 4) Всі вершини в трикутнику суміжні, що означає – від кожної вершини існує мінімальний шлях до будь-якої іншої, чого немає ні в одному іншому геометричному примітиві. 5) Трикутник має мінімальну вартість міжз'єднань, що утворюють площину, яку можна прийняти за одиницю. 6) Комунікації, складені з трикутних структур, є оптимальними за критерієм: середня довжина шляху між будь-якими двома точками топології, віднесена до вартості її реалізації, чого немає ні в одному іншому геометричному примітиві. 7) Трикутник має мінімальну вартість міжз'єднань, що утворюють площину, яку можна прийняти за одиницю. 8) Комунікації, складені з трикутних структур, є оптимальними за критерієм: середня довжина шляху між будь-якими двома точками топології, віднесена до вартості її реалізації. чого немає ні в одному іншому геометричному примітиві. 9) Трикутник має мінімальну вартість міжз'єднань,

що утворюють площину, яку можна прийняти за одиницю. 7) Комунікації, складені з трикутних структур, є оптимальними за критерієм: середня довжина шляху між будь-якими двома точками топології, віднесена до вартості її реалізації.

На основі введеної раніше β -метрики можна формувати квазіоптимальний кіберпростір планети. Воно не прив'язане до геометричної топології поверхні і в цьому сенсі є віртуальним. Але в частині покриття поверхні і простору телекомунікаціями фізична основа Cyber Space повинна бути мажорірованою з метою забезпечення відмовостійкості та надійності функціонування каналів зв'язку при виникненні катаклізмів. Інформаційна структура Cyber Space повинна бути ієрархічною та замкненою як глобально, так і локально, на будь-якому рівні ієрархії. Елементарна комірка структури простору повинна бути трикутною. Це забезпечить суттєве зменшення інформаційних обсягів Cyber Space, у межі – на третину. Що означає підвищення продуктивності всіх приймально-передавачів і сховищ контенту планети на 33%? Дане твердження пов'язано з аксіомою транзитивного замикання кінцевого числа (1,2,3,4, ..., n) точок в інформаційному векторно-логічному просторі:

- 1) $d_1 = 0$;
- 2) $d_1 \oplus d_2 = 0$;
- 3) $d_1 \oplus d_2 \oplus d_3 = 0$;
- 4) $d_1 \oplus d_2 \oplus d_3 \oplus d_4 = 0$;
- 5) $d_1 \oplus d_2 \oplus \dots \oplus d_i \oplus \dots \oplus d_n = 0$.

Розглядаючи всі можливі варіанти транзитивного замикання з позиції мінімізації умов для відновлення інформації, очевидним є факт відновлення третьої сторони трикутника по двом відомим. В цьому випадку достатньо, наприклад, передати по каналу зв'язку дві сторони, щоб відновити третю. Скорочення переданих обсягів в даному випадку дорівнює 33%. У всіх інших

випадках зменшення обсягів інформації буде менш істотним. Наприклад, для замкнутого чотирикутника необхідно передати три відстані або сторони, щоб відновити четверту, виграш – 25%. Можлива площинна інтерпретація Cyber Space має вигляд, представлений на рис. 2.4.

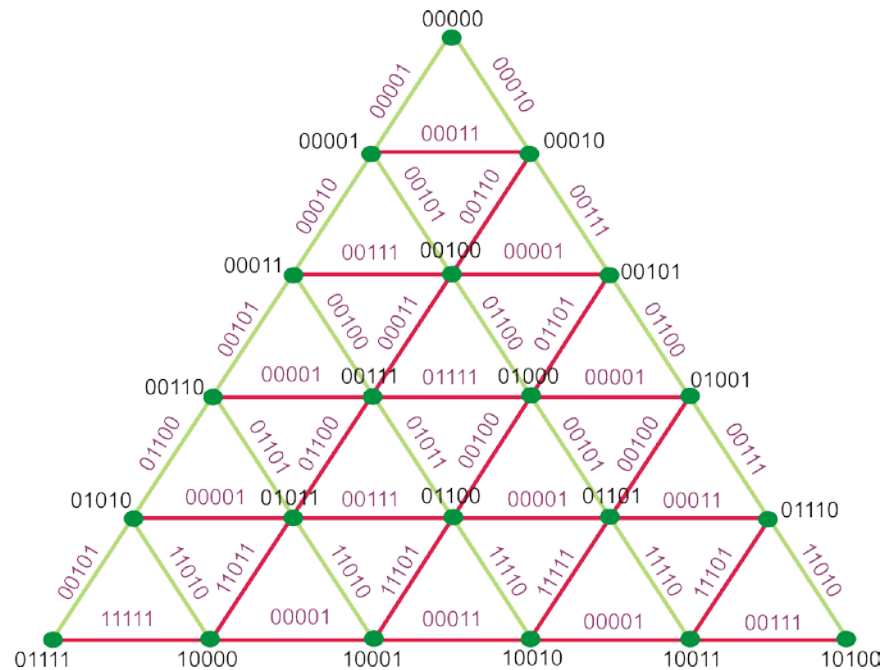


Рис. 2.4 – Трикутний кіберпростір

Щоб відновити цю структуру, необхідно знати вже не 2/3 обсягу інформації, яка складена кожними двома сторонами трикутників. Досить вже мати сторони, задані зеленим кольором, що становить 44% від обсягу кодів-відстаней. Решта 56% компонентів трикутників можуть бути визначені за допомогою аксіоми:

$$d_1 \oplus d_2 \oplus d_3 = 0 \rightarrow d_3 = d_1 \oplus d_2.$$

У загальному випадку функціональна залежність відношення відновлюваних сторін замкнутого у трикутник простору до загальної кількості n шарів трикутної структури визначається як:

$$\eta = \frac{\sum_{i=1}^n (i+1)}{\sum_{i=1}^n (3 \times i)} = \frac{1}{3} \left(\frac{2}{n+1} + 1 \right) = \frac{1}{3} \cdot \frac{2+n+1}{n+1} = \frac{1}{3} \cdot \frac{n+3}{n+1} = \frac{n+3}{3(n+1)}.$$

Графік відношення кількості переданої інформації до її повного обсягу відстаней в трикутному кібер-просторі представлений на рис. 2.5. Граничне значення графіка наближається до 33%. Це означає – для великих обсягів трикутного простору досить передати третю частину всіх кодів-відстаней, щоб відновити повний простір кодів.

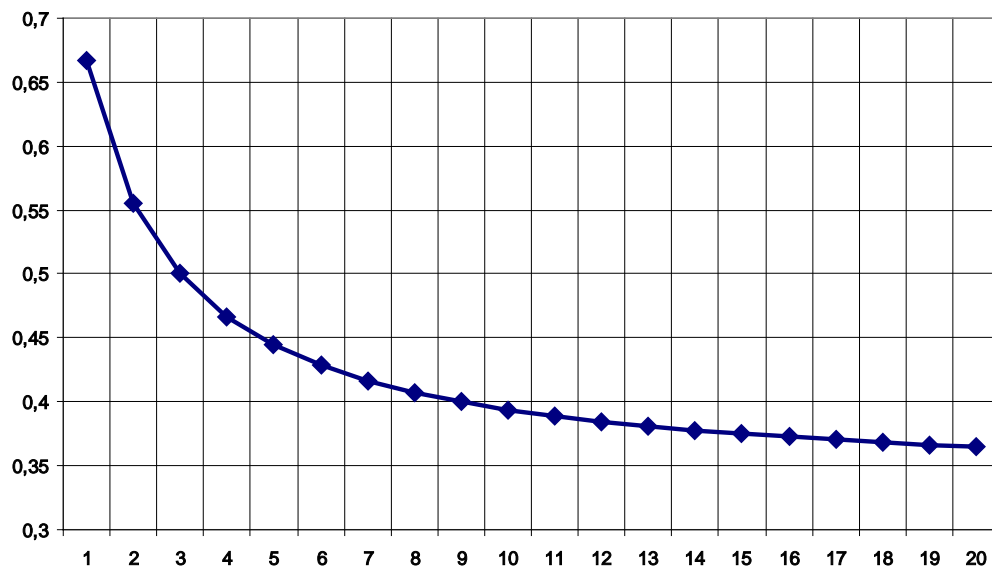


Рис. 2.5 – Функція числа відстаней для відновлення простору

Трикутна метрика простору є самою економічною. Вона створює короткі відстані та шляхи між об'єктами завдяки наявності транзитивного замикання. Сусідство об'єкта з шістьма точками робить метрику більш кращою у порівнянні з топологією «Манхеттен» в частині прокладання шляху між двома точками. Оптимальна структура мультипроцесорної системи повинна бути складена з трикутників. Тут оптимальність визначається відношенням середньої довжини шляху між будь-якими двома точками до загальної кількості сторін, формує вартість топології. Довжина шляху у трикутній топології завжди буде не гірше, ніж в «Манхеттені», але загальне число зв'язків на одну діагональ більше в кожному чотирикутнику. Це дає можливість прямого доступу до 6 сусідів, що істотно для мультипроцесора. «Манхеттен» з діагональними зв'язками має ту ж довжину шляху, що і трикутний простір, маючи при цьому на один зв'язок більше. Тут задіяна друга

діагональ чотирикутника, що забезпечує сусідство з вісьмома вершинами простору. Оскільки трикутник – найпримітивніша фігура, яка формує площину, то вага однієї сторони, яку не слід зберігати для ідентифікації простору, буде максимальним у трикутнику. Система трикутників дає вже 66% структурних компонентів, які можна не описувати при формуванні простору для подальшого відновлення. Всі інші примітивні площинні фігури дають менший вигаш при формуванні Cyber Space. Дуже важливо для підвищення надійності цифрових систем вміти відновлювати до 66 відсотків несправних компонентів без втрати інформації.

2.5 Спрощений критерій якості топології

Середня довжина шляху між компонентами топології (цифрової системи) є основний параметр, що впливає на швидкодію виконання транзакцій між вузлами структури. При розгляді варіантів реалізації топології комп'ютерної системи необхідно визначити інтегральну характеристику у вигляді суми всіх відстаней між кожною парою компонентів або вершин відповідного графа, наведеної до числа ребер або дуг. Тут інтерес представляють примітивні фігури: чотирикутник, трикутник і тетраедр. Останній має унікальну властивість – кожна вершина тетраедра має три сусідніх, в той час як трикутник має унікальні дві суміжні вершини на площині. Для оцінки вартості інформаційної транзакції між двома компонентами системи в графі, що містить n вершин, вводиться критерій якості структури, $\min P_i$ між усіма парами вершин $\frac{1}{2}(n^2 - n)$ до кількості ребер топології k :

$$L = \frac{\sum_{i=1}^{\frac{1}{2}(n^2-n)} \min P_i}{\frac{1}{2}(n^2-n)} \times \frac{\frac{1}{2}(n^2-n)}{k} = \frac{\sum_{i=1}^{\frac{1}{2}(n^2-n)} \min P_i}{k} = \frac{1}{k} \sum_{i=1}^{\frac{1}{2}(n^2-n)} \min P_i.$$

Тут аргумент k – кількість ребер в графі є витратною частиною або вартісною характеристикою топологічної структури, тому вона фігурує в знаменнику. Але збільшення даного параметра може істотно зменшити

чисельник дробу, в якому записана інтегральна довжина мінімальних шляхів між усіма парами вершин графа, яка наближається до мінімуму. Таким чином, чим менше інтегральна вартість тразакції L , тим вище ефективність топологічної структури.

Приклад 1. Визначити критерії якості трьох структурних реалізацій, що мають 4 вершини-компонента (a, b, c, d): $\mathbf{G}_1 = \{ab, bc, ad, cd\}$, $\mathbf{G}_2 = \{ab, ac, ad, cd, bc, bd\}$, $\mathbf{G}_3 = \{ab, bc, ac, ad, cd\}$:

$$\begin{array}{c}
 \mathbf{G}_1 = \\
 \begin{array}{c|cccc}
 & a & b & c & d \\
 a & . & 1 & 1 & . \\
 b & 1 & . & . & 1 \\
 c & 1 & . & . & 1 \\
 d & . & 1 & 1 & . \\
 \hline
 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{G}_2 = \\
 \begin{array}{c|cccc}
 & a & b & c & d \\
 a & . & 1 & 1 & 1 \\
 b & 1 & . & 1 & 1 \\
 c & 1 & 1 & . & 1 \\
 d & 1 & 1 & 1 & . \\
 \hline
 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{G}_3 = \\
 \begin{array}{c|cccc}
 & a & b & c & d \\
 a & . & 1 & 1 & . \\
 b & 1 & . & 1 & 1 \\
 c & 1 & 1 & . & 1 \\
 d & . & 1 & 1 & . \\
 \hline
 \end{array}
 \end{array}$$

Використовуючи наведену формулу, досить просто підрахувати відношення суми довжин мінімальних шляхів для кожної з 6 пар: $\{ab, ac, ad, cd, bc, bd\}$, з'єднаних $k = (4,6,5)$ ребрами топології:

$$L_1 = \frac{1+2+1+1+2+1}{4} = \frac{8}{4} = 2,0;$$

$$L_2 = \frac{1+1+1+1+1+1}{6} = \frac{6}{6} = 1,0;$$

$$L_3 = \frac{1+1+1+1+2+1}{5} = \frac{7}{5} = 1,4.$$

Важливою характеристикою L -критерію є його чутливість, коли незначна модифікація графа – додавання одного ребра – може привести до істотної зміни оцінки в бік зменшення. Ринкова привабливість аналізу ефективності структур актуальна для транспортної інфраструктури міст при проектуванні і перебудові комунікацій.

Приклад 2. Показано застосування критерію якості до оцінювання трьох графових структур, що мають 6 вершин і різні топології з'єднання, представлені на рис. 2.1:

		a	b	c	d	e	f			a	b	c	d	e	f			a	b	c	d	e	f
$G_1 = c$.	1	1	1	.	1	.	.			.	1	.	1	1	.
		1	.	1	1	.	1			1	.	1	.	1	.			1	.	1	1	1	1
		.	1	.	.	.	1	$G_2 = c$.	1	.	.	.	1	$G_3 = c$.	1	.	.	1	1
		1	1	.	.	1	1			1	.	.	.	1	.			1	1	.	.	1	.
		.	.	.	1	.	1			.	1	.	1	.	1			1	1	1	1	.	1
		.	1	1	1	1	.			.	.	1	.	1	.			.	1	1	.	1	.

Тут три графа мають 9, 7 і 11 дуг відповідно. Підрахунок критерію за L-формулою дає наступні результати:

$$L(G_1) = \frac{9 \times 1 + 6 \times 2}{9} = \frac{21}{9} = 2,33;$$

$$L(G_2) = \frac{7 \times 1 + 6 \times 2 + 2 \times 3}{7} = \frac{25}{7} = 3,57;$$

$$L(G_3) = \frac{11 \times 1 + 4 \times 2}{11} = \frac{19}{11} = 1,73.$$

Висновки: 1) Отриманий критерій якості топології досить просто реалізувати у хмарному сервісі, доступному для широкого кола користувачів, що бажають визначити ефективність старої або інноваційної структури комп'ютерної системи, університету, соціальної мережі, компанії або транспортної інфраструктури. 2) Критерій показує очевидні переваги трикутної топології взаємодії системних компонентів для обслуговування транзакційних процесів у кіберфізичному просторі. 3) Трикутна метрика простору є сама економічна. Вона створює найкоротші відстані та шляхи між координатами, завдяки наявності транзитивного замикання. Трикутна топологія міст виграє у метрики Манхеттена по скороченню маршрутів руху від 33 до 66 відсотків. 4) Впровадження даної топології в структурну

організацію цифрових систем на 2D, 3D кристалах забезпечить скорочення довжини з'єднувальних ліній і підвищення швидкодії не менше, ніж на 33 відсотки. 5) Дана топологія забезпечує підвищення надійності обчислювальних систем і мереж до 66 відсотків, завдяки трикутній організації зв'язків структурних компонентів, коли будь-який відмовивший компонент має не менше двох сусідів. 6) Об'ємна організація системних компонентів за трикутною тетраедною топологією надає ще більші переваги щодо оптимізації транзакційних маршрутів для кіберфізичних об'єктів, включаючи авіатранспорт, глобальний комп'ютинг і 3D-SoC. 7) Трикутна інфраструктура забезпечує підвищення надійності цифрових систем завдяки можливості відновлення до 66 відсотків несправних зв'язків компонентів без втрати інформації. Інакше, 66% несправних зв'язків у трикутній топології системи залишає її в працездатному стані. 8) Недоліком трикутної топології можна вважати незвичайність для людини трикутної системи координат, що звик до декартової метриці. Тут мова йде про ідентифікацію точки на площині трьома координатами. Декартова метрика передбачає використання двох координат для позиціонування об'єкта.

Приклад 3. Показано застосування критерію якості до оцінювання двох графових структур, що мають 21 вершину, які складають топології "Манхеттен" і "трикутник", рис. 2.6 і 2.7. Кожному рисунку ставиться у відповідність матриця смежностей (табл. 2.1, 2.3), яка визначає сусідство вершин в графі, а також матриця відстаней (табл. 2.2, 2.4), що задає довжину шляхів між кожною парою вершин. Щоб оцінити якість топології кожного графа з позиції довжини шляху, необхідно скласти всі мінімальні шляхи між парами вершин і розділити їх на кількість пар, яке визначається числом сполучень з 21 вершини по 2.

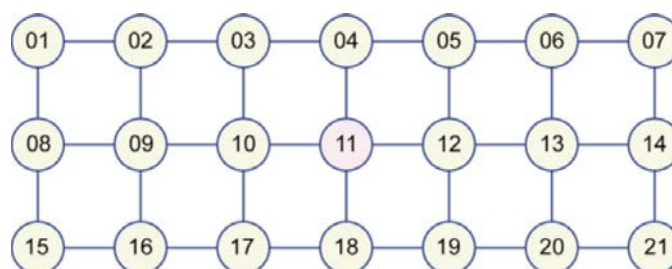


Рис. 2.6 – Прямокутний граф топології "Манхеттен"

Таблиця 2.1 – Матриця суміжностей для прямокутного графа

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2		0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3			0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4				0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5					0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6						0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7							0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
8								0	1	0	0	0	0	0	1	0	0	0	0	0	0
9									0	1	0	0	0	0	0	1	0	0	0	0	0
10										0	1	0	0	0	0	0	1	0	0	0	0
11											0	1	0	0	0	0	0	1	0	0	0
12												0	1	0	0	0	0	0	1	0	0
13													0	1	0	0	0	0	0	1	0
14														0	0	0	0	0	0	0	1
15															0	1	0	0	0	0	0
16																0	1	0	0	0	0
17																	0	1	0	0	0
18																		0	1	0	0
19																			0	1	0
20																				0	1
21																					0

Таблиця 2.2 – Таблиця відстаней для прямокутного графа

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	2	3	4	5	6	1	2	3	4	5	6	7	2	3	4	5	6	7	8
2		0	1	2	3	4	5	2	1	2	3	4	5	6	3	4	5	6	7	8	9
3			0	1	2	3	4	3	2	1	2	3	3	5	4	3	2	3	4	5	6
4				0	1	2	3	4	3	2	1	2	3	4	5	4	3	2	3	4	5
5					0	1	2	5	4	3	2	1	2	3	6	5	4	3	2	3	4
6						0	1	6	5	4	3	2	1	2	7	6	5	4	3	2	3
7							0	7	6	5	4	3	2	1	8	7	6	5	4	3	2
8								0	1	2	3	4	5	6	1	2	3	4	5	6	7
9									0	1	2	3	4	5	2	1	2	3	4	5	6
10										0	1	2	3	4	3	2	1	2	5	4	5
11											0	1	2	3	4	3	2	1	2	3	4
12												0	1	2	5	4	3	2	1	2	3
13													0	1	6	5	4	3	2	1	2
14														0	7	6	5	4	3	2	1
15															0	1	2	3	4	5	6
16																0	1	2	3	4	5
17																	0	1	2	3	4
18																		0	1	2	3
19																			0	1	2
20																				0	1
21																					0

Для прямокутної топології графа критерій якості визначається наступною оцінкою: $L = \text{sum} / k$, $k = 32$, $\text{Sum} = 714$; $L = 714/32 = 22,3125$.

Таблица 2.3 – Матрица суміжностей для трикутного графа

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2		0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3			0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4				0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5					0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
6						0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
7							0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
8								0	1	0	0	1	1	0	0	0	0	0	0	0	0
9									0	1	0	0	1	1	0	0	0	0	0	0	0
10										0	0	0	0	1	1	0	0	0	0	0	0
11											0	1	0	0	0	1	1	0	0	0	0
12												0	1	0	0	0	1	1	0	0	0
13													0	1	0	0	0	1	1	0	0
14														0	1	0	0	0	1	1	0
15															0	0	0	0	0	1	1
16																0	1	0	0	0	0
17																	0	1	0	0	0
18																		0	1	0	0
19																			0	1	0
20																				0	1
21																					0

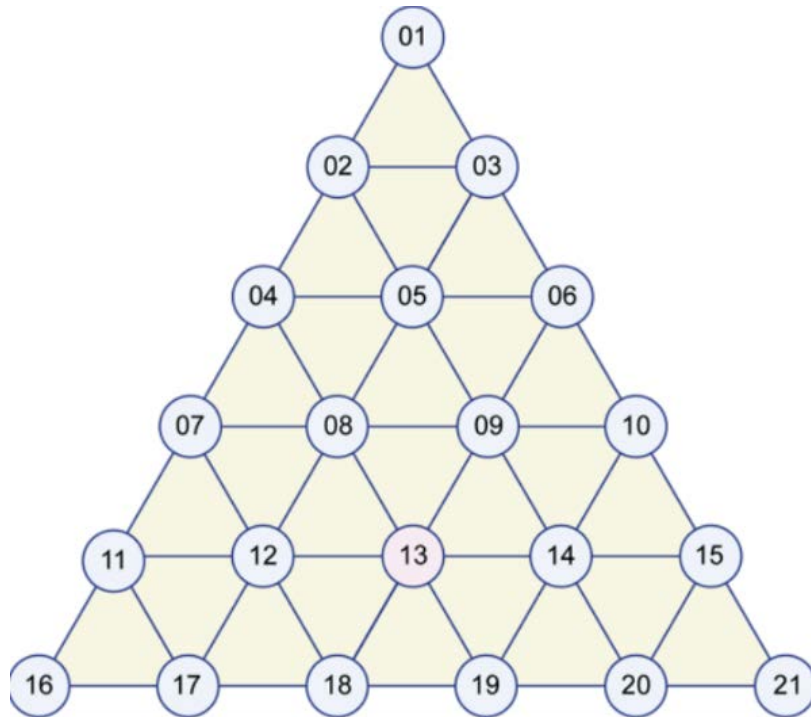


Рис. 2.7 – Трикутний граф

Таблиця 2.4 – Таблиця відстані для трикутного графа

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	1	2	2	2	3	3	3	3	4	4	4	4	4	5	5	5	5	5	5
2		0	1	1	1	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	5
3			0	0	1	1	3	2	2	2	4	3	3	3	4	5	4	4	4	4	4
4				0	1	2	1	1	2	3	2	2	2	3	4	3	3	3	3	4	5
5					0	1	2	1	1	2	3	2	2	2	3	4	3	3	3	3	4
6						0	3	2	1	1	4	3	2	2	2	5	4	3	3	3	3
7							0	1	2	3	1	1	2	3	4	2	2	2	3	4	5
8								0	1	2	2	1	1	2	3	3	2	2	2	3	4
9									0	1	3	2	1	1	2	4	3	2	2	2	3
10										0	4	3	2	1	1	5	4	3	2	2	2
11											0	1	2	3	4	1	1	2	3	4	5
12												0	1	2	3	2	1	1	2	3	4
13													0	1	2	3	2	1	1	2	3
14														0	1	4	3	2	1	1	2
15															0	5	4	3	2	1	1
16																0	1	2	3	4	5
17																	0	1	2	3	4
18																		0	1	2	3
19																			0	1	2
20																				0	1
21																					0

Для трикутної топології графа критерій якості визначається наступною оцінкою: $L = \text{sun} / k$, $k = 45$, $\text{Sum} = 545$ $L = 545/45 = 12,1$. Це означає, що дорожня інфраструктура, побудована на трикутних з'єднаннях пунктів відправлення і прибуття в два рази ефективніше, ніж загальноприйнята топологія "Манхеттен", що має оцінку $L = 22,3125$.

2.6 Наведена метрика топології графа

Раніше згадувалося, що середня довжина шляху між компонентами топології є основний параметр, що впливає на час виконання маршруту між вузлами інфраструктури. Для топології, яка має 4 компоненти (a, b, c, d), на площині досить просто підрахувати відношення кількості пар вершин до суми довжин мінімальних шляхів на кожній з n пар: ab, ac, ad, bc, bd, cd, з'єднаних k ребрами топології:

$$L = \frac{k}{\frac{1}{2}(n^2-n)} \times \frac{\frac{1}{2}(n^2-n)}{\sum_{i=1}^n \min P_i}$$

Перший співмножник визначає приведену до повного графу (кожна вершина безпосередньо пов'язана з іншими) вартість міжз'єднань структури, а другий – формує зворотну оцінку середньої довжини шляху для кожної пари вершин графа. Обидва співмножника визначені в інтервалі (0,1), де краще рішення задається оцінкою $L=1$. Тут $\min P_i$ визначається шляхом використання модифікованого методу Дейкстри, наведеного в наступному розділі. Обчислення оцінок для трьох топологій системних зв'язків дають наступний результат: 1) чотирикутник: $L = 4 / (1 + 2 + 1 + 1 + 2 + 1) = 0,50$; 2) трикутна структура, яка містить два примітиви: $L = 5 / (1 + 1 + 1 + 1 + 2 + 1) = 0,71$; 3) чотирикутник з двома діагоналями: $L = 6 / (1 + 1 + 1 + 1 + 1 + 1) = 1,0$.

Далі можна визначити інтервальний $[0,1]$ критерій структурної ефективності, як число n всіх можливих пар (сполучень по два) на фіксованій множині вершин графа, віднесене до суми мінімальних відстаней між ними:

$$Q = \frac{\sum_{i=1}^n \frac{1}{\min p_i}}{\sum_{i=1}^n \min p_i} = \frac{n}{\sum_{i=1}^n \min p_i}.$$

В цьому випадку наведені вище оцінки матимуть вигляд:

$$Q_1 = \left[\frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}{2} \right] = \frac{6}{8} = 0,75$$

$$Q_2 = \frac{1}{7} \left[\frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} \right] = \frac{6}{6} = 1,0$$

$$Q_3 = \left[\frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{2} \right] = \frac{6}{7} = 0,86$$

Платою за підвищення ефективності топології служить структурна складність проекту, яка визначається кількістю дуг n .

Таким чином, ефективність дорожньої інфраструктури є відношення кількості (ребер) парних з'єднань n компонентів до суми мінімальних шляхів (досяжностей p) кожної пари:

$$Q_t = \frac{\frac{1}{2}(n^2 - n)}{\sum_{i=1}^n \min(p_{ij})}$$

Дана оцінка буде дорівнювати одиниці, якщо відповідний інфраструктурі граф буде повним, коли дві будь-які вершини є суміжними. Однак реалізація такого графа має високу вартість, що призводить до необхідності ввести оцінку складності інфраструктури як відношення реальної кількості E ребер графа, що містить n вершин до повного графу:

$$Q_s = \frac{E}{\frac{1}{2}(n^2 - n)}$$

Нехай є три структури, що містять 4 вершини, представлені у формі матриці суміжностей:

$$G_1 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & . & 1 & 1 & . \\ b & 1 & . & . & 1 \\ c & 1 & . & . & 1 \\ d & . & 1 & 1 & . \end{array} \quad G_2 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & . & 1 & 1 & 1 \\ b & 1 & . & 1 & 1 \\ c & 1 & 1 & . & 1 \\ d & 1 & 1 & 1 & . \end{array} \quad G_3 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & . & 1 & 1 & . \\ b & 1 & . & 1 & 1 \\ c & 1 & 1 & . & 1 \\ d & . & 1 & 1 & . \end{array}$$

Обчислення двох оцінок, введених вище, дає наступний результат:

$$\begin{aligned} Q_t(G_1) &= \frac{6}{8} = 0,75 & Q_s(G_1) &= \frac{4}{6} = 0,66 \\ Q_t(G_2) &= \frac{6}{6} = 1,0 & Q_s(G_2) &= \frac{6}{6} = 1,0 \\ Q_t(G_3) &= \frac{6}{7} = 0,86 & Q_s(G_3) &= \frac{5}{6} = 0,83 \end{aligned}$$

Якщо мультиплікувати обидві оцінки, то вийде узагальнений критерій ефективності інфраструктури для довільного числа компонентів або вершин, який залежить від кількості з'єднань E , віднесених до суми мінімальних шляхів всіх пар вершин:

$$Q = \frac{\frac{1}{2}(n^2 - n)}{\sum_{i=1}^n \min(p_{ij})} \times \frac{E}{\frac{1}{2}(n^2 - n)} = \frac{E}{\sum_{i=1}^n \min(p_{ij})}.$$

Перерахунок критерію ефективності для трьох варіантів структури представлений наступними оцінками:

$$Q(G_1) = \frac{4}{8} = 0,5;$$

$$Q(G_2) = \frac{6}{6} = 1,0;$$

$$Q(G_3) = \frac{5}{7} = 0,71.$$

Подальша модифікація критерію спрямована на отримання зваженої оцінки, яка враховує вартість шляху (середньої довжини маршруту між двома вершинами), наведеної до витрат – фактичного числа ребер графа апаратної реалізації обчислювальної структури:

$$Q = \frac{E}{\frac{1}{2}(n^2 - n)} \times \left[\frac{1}{n \times E} \times \sum_{i=1}^n \min(p_{ij}) \right] = \frac{2 \times \sum_{i=1}^n \min(p_{ij})}{n \times (n^2 - n)} = \frac{2 \times \sum_{i=1}^n \min(p_{ij})}{n^2(n-1)}.$$

Тут перший співмножник $\frac{E}{\frac{1}{2}(n^2 - n)}$ – витратна частина або число дуг,

що формують граф, віднесене до максимально можливого їх числа повного

графа, а другий співмножник $\frac{1}{n \times E} \times \sum_{i=1}^n \min(p_{ij})$ є середня довжина шляху між

двома вершинами, приведена до максимально можливого маршруту, складеного з усіх дуг повного графа. Таке приведення (поділ на число дуг повного графа) дає можливість нормувати другий співмножник, який без приведення завжди не менш одиниці, до одиничного інтервалу. Однак незручність використання даного критерію пов'язано з характером функцій,

що задаються першим і другим співмножники, які, по суті, працюють в протифазі. Якщо вартість графа збільшується, то середня довжина шляху зменшується, що призводить до зменшення критерію. Для усунення даного недоліку і з метою отримання оцінки, коли її значення, чим ближче до 1, тим краще, необхідна її модифікація щодо другого співмножника:

$$Q = \frac{E}{\frac{1}{2}(n^2 - n)} \times \left[1 - \frac{1}{n \times E} \times \sum_{i=1}^n \min_j(p_{ij}) \right] = \frac{2 \times E}{(n^2 - n)} \times \left[1 - \frac{1}{n \times E} \times \sum_{i=1}^n \min_j(p_{ij}) \right].$$

Несуттєве ускладнення оцінки призводить до характеру функціональної залежності, коли довжина з'єднання (другий співмножник) прямо пропорційно залежить не тільки від числа дуг у відповідному графі (перший співмножник), але і від яких саме дуг, що роблять шлях між двома вершинами коротше. Важливою характеристикою даної оцінки є її чутливість, коли незначна модифікація графа – додавання одного ребра – може привести до істотної зміни оцінки в сторону збільшення. Ринкова привабливість аналізу ефективності структур актуальна для інфраструктури міст в умовах існування транспортних заторів:

$$Q_1 = \frac{E}{\sum_{i=1}^n \min_j(p_{ij})};$$

$$Q_2 = \frac{2 \times \sum_{i=1}^n \min_j(p_{ij})}{n^2(n-1)};$$

$$Q_3 = \frac{2 \times E}{(n^2 - n)} \times \left[1 - \frac{1}{n \times E} \times \sum_{i=1}^n \min_j(p_{ij}) \right].$$

Ще одна модифікація критерію пов'язана зі спрощенням обробки графової структури, що має E дуг і n вершин. Його особливість полягає в

обчисленні абсолютного і не наведеного до інтервалу значення, який формується вартістю з'єднань, помноженої на якість транзакцій між усіма парами вершин:

$$Q_4 = \frac{E}{n} \times \sum_{i=1}^n \min(p_{ij}).$$

Застосування цієї формули до оцінювання трьох графових структур, що мають 6 вершин і різні топології з'єднання представлені нижче (див. рис. 2.1):

	a	b	c	d	e	f		a	b	c	d	e	f		a	b	c	d	e	f			
$G_1 =$	a	.	1	1	.	.	$G_2 =$	a	.	1	.	1	.	$G_3 =$	a	.	1	.	1	1	.		
	b	1	.	1	1	.		b	1	.	1	.	1		.	b	1	.	1	1	1	1	
	c	.	1	.	.	.		1	c	.	1	.	.		.	1	c	.	1	.	.	1	1
	d	1	1	.	.	1		1	d	1	.	.	.		1	.	d	1	1	.	.	1	.
	e	.	.	.	1	.		1	e	.	1	.	1		.	1	e	1	1	1	1	.	1
	f	.	1	1	1	1		.	f	.	.	1	.		1	.	f	.	1	1	.	1	.

Тут три графа мають 9, 7 і 11 дуг відповідно. Підрахунок критерію відповідно до останньої формули дає наступні результати:

$$Q_4(G_1) = \frac{E}{n} \times \sum_{i=1}^n \min(p_{ij}) = \frac{9}{6} \times (9 \times 1 + 6 \times 2) = 31,5;$$

$$Q_4(G_2) = \frac{7}{6} \times (7 \times 1 + 6 \times 2 + 2 \times 3) = 29,2;$$

$$Q_4(G_3) = \frac{11}{6} \times (11 \times 1 + 4 \times 2) = 34,8.$$

Модифікація оцінки ефективності топології пов'язана з приведенням реальних витрат (число дуг E) до максимально можливої кількості $V = \frac{n^2 - n}{2}$

парних з'єднань в графі $\frac{E}{V}$, що забезпечує якість комунікаційних

властивостей $\frac{V}{\sum_{i=1}^V \min(p_{ij})}$:

$$Q_4 = \frac{E}{\frac{n^2 - n}{2}} \times \frac{\frac{n^2 - n}{2}}{\sum_{i=1}^2 \sum_j \min(p_{ij})} = \frac{E}{\sum_{i=1}^2 \sum_j \min(p_{ij})} = \frac{E}{\sum_{i=1}^V \sum_j \min(p_{ij})}.$$

Оцінка дорівнює одиниці, якщо чисельник і знаменник рівні $V = \frac{n^2 - n}{2}$.

У даному випадку графова структура має всі можливі парні сполуки між вершинами графа, які визначаються половиною декартова квадрата потужності множини вершин за вирахуванням n вершин. Віднімання визначається вершинами графа, що не мають ідемпотентних замикань. При цьому кожна пара вершин має довжину шляху, що дорівнює одиниці. Перерахунок критеріїв ефективності, дає наступний результат:

$$Q_5(G_1) = \frac{E}{\sum_{i=1}^{V=12} \min(p_{ij})} = \frac{9}{9 \times 1 + 6 \times 2} = 0,428;$$

$$Q_5(G_2) = \frac{7}{7 \times 1 + 6 \times 2 + 2 \times 3} = 0,28;$$

$$Q_5(G_3) = \frac{11}{11 \times 1 + 4 \times 2} = 0,578.$$

При цьому платою за якість комунікацій є потужність з'єднань, приведена до максимально можливої кількості ребер:

$$H_5(G_1) = \frac{E}{\frac{n^2 - n}{2}} = \frac{9}{15} = 0,60;$$

$$H_5(G_2) = \frac{7}{15} = 0,46;$$

$$H_5(G_3) = \frac{11}{15} = 0,73.$$

Доцільно мати дві оцінки: інтегральний критерій якості комунікацій, який неявно визначає витрати часу на середню досяжність між кожною парою вершин графових структури, а також наведену до максимально можливого числа потужність з'єднань, яка демонструє вартість якості інфраструктури системи, що має цільову функцію, що мінімізувала середню досяжність (довжину шляху або часу) між парою вершин графової структури. Залежно від числа з'єднань перший критерій має тенденцію до зростання від 0 до 1, другий також збільшується за мірою збільшення числа ребер в графі. Тому мультиплікування двох критеріїв не дає нової властивості при оцінюванні інфраструктури досяжностей кожної пари вершин. Висновки: 1) Необхідно використовувати обидва критерії для оцінювання структурного проекту. 2) Слід модифікувати алгоритм Дейкстра для обчислення середнього значення досяжностей між парою вершин в графі, що і представлено нижче.

2.7 Висновки до розділу 2

Наукова новизна розділу формується наступними пунктами:

1) Розроблено метрику і критерії практичного оцінювання маршрутів руху і дорожніх інфраструктур для реалізації хмарних сервісів управління дорожнім рухом, що надаються водієві у режимі реального часу.

2) Запропоновано критерії оцінювання якості інфраструктури дорожнього руху, які враховують середнє значення довжини шляху в інфраструктурі, а також вартість створення дорожньої структури.

3) Розроблено аналітичні моделі прохідності дорожньої інфраструктури, які враховують поняття спостереження та управління для формування алгоритмів управління на основі використання світлофорів.

4) Запропоновано метрики для вимірювання е-інфраструктури дорожнього руху, які дають можливість створювати ефективні маршрути руху на основі використання трикутних примітивів. Представлені приклади різних топологій і зв'язків компонентів інфраструктури.

5) Створено систему спрощених і наведених критеріїв оцінювання якості топології дорожньої інфраструктури, яка орієнтована на прокладання хмарних маршрутів в реальному масштабі часу.

Основні результати розділу 2 опубліковані в роботах [142, 143, 145, 148, 150, 152].

3 МОДИФІКАЦІЯ АЛГОРИТМУ ДЕЙКСТРИ ДЛЯ ПОШУКУ МАРШРУТУ

Розглядається алгоритм побудови оптимального маршруту на дорожній інфраструктурі між двома координатами, які ототожнюються з пунктом відправлення та прибуття. Можливі обмеження пов'язані з дорожніми колізіями, аваріями і ремонтом доріг, що також має бути враховано при виборі оптимального маршруту. Передбачається, що реалізація алгоритму буде представлена як online хмарний сервіс для водіїв транспортних засобів в рамках створення кіберфізичної системи інтелектуального хмарного управління транспортом. Алгоритм використовується для обчислення всіх можливих шляхів або відстаней між кожною парою компонентів або вершин відповідної топології.

3.1 Вступ

Підстави для представлених нижче досліджень відображені у наступних публікаціях і розробках:

- теоретичні розробки інтелектуальних моделей, методів і програмно-апаратних засобів аналізу кіберпростору, пов'язані з дискретною оптимізацією, пошуком, розпізнаванням та прийняттям рішень [97-100];
- досвід розробки та застосування вбудованих і RFID цифрових систем для моніторингу дорожнього руху [101-105];
- досвід розробки та впровадження програмних продуктів і хмарних сервісів для оптимізації маршрутів транспортних засобів українських підприємств з метою мінімізації матеріальних та часових витрат і підвищення якості обслуговування пасажирів [106-111];
- розробки розподіленої системи управління дорожнім рухом в умовах великих міст і мегаполісів на основі високонадійної обчислювальної техніки

[107-109], прототипування розумних кіберфізичних систем управління трафіком [132, 133];

– методи оцінювання обчислювальних структур і пошуку найкоротших шляхів між парою вершин [135-138], відомі модифікації алгоритму Дейкстра [134-139].

Мета дослідження – синтез оптимального маршруту руху транспортного засобу шляхом використання модифікованого алгоритму Дейкстри на топологічній інфраструктурі дорожнього руху, що приводиться до графу, з урахуванням існування дорожніх колізій. Завдання: 1) Визначення всіх найкоротших шляхів і відповідних відстаней з вершини a до всіх інших вершин інфраструктури. 2) Для графа з одиничними ваговими коефіцієнтами, заданого матрицею суміжності, знайти всі найкоротші відстані між усіма парами вершин. 3) Для різних типів графів, заданих матрицями суміжності, знайти найкоротші відстані між усіма парами вершин і оцінити інфраструктури за розробленими критеріями якості. 4) Виконати порівняння різних архітектур і графових структур з метою визначення ефективних рішень для скорочення часу руху за замовленим маршрутом.

Відоме удосконалення алгоритму Дейкстри відшукування ланцюгів найменшої довжини з обраної вершини графа (пункту відправлення) до всіх інших його вершин (можливих пунктів прибуття) пов'язане зі скороченням кількості операцій (додавань і порівнянь) і збереженням отриманої інформації на одному з етапів для наступних обчислень. Це досягається процедурою розміщення міток алгоритму Дейкстри, яка зменшує складність алгоритму до $O(n^2)$. Відомо, що при визначенні найкоротших шляхів між усіма парами вершин використовується алгоритм Флойда-Уоршелла, складність якого становить $O(n^3)$.

3.2 Метод пошуку найкоротших шляхів у дорожній інфраструктурі

Завдання 3.1. Для графа G_1 , представленого на рис. 3.10, визначити всі найкоротші шляхи і відповідні їм відстані з вершини a до всіх інших вершин.

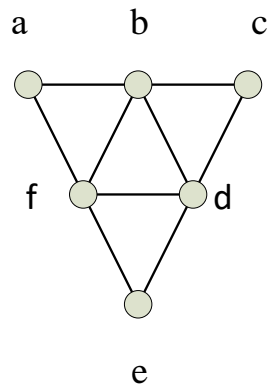


Рис. 3.10 – Граф G_1 – конфігурація мережі

Розв'язок. Розглянутий на рис. 3.10 граф є неорієнтованим. Він має дві рівноправні між собою групи вершин: 1) a, c, e ; 2) b, d, f . «Рівноправність» розуміється в сенсі інваріантності уявлення найкоротших шляхів і їх дерев при пошуку з вершин a, c, e , що породжує першу групу шляхів, або з вершин b, d, f – другу групу шляхів. Отже, для даної задачі замість алгоритму Флойда можна двічі послідовно застосувати алгоритм Дейкстри, щоб знайти найкоротші шляхи між усіма парами вершин.

Отже, завдання розподіляється на дві підзадачі: 1) знайти найкоротші відстані і вказати всі найкоротші шляхи від вершини a до всіх інших вершин; 2) знайти найкоротші відстані і вказати всі найкоротші шляхи від вершини b до всіх інших вершин.

Матриця суміжності графа G_1 має вигляд:

$$G_1 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & . & . & . & 1 \\ b & 1 & . & 1 & 1 & . & 1 \\ c & . & 1 & . & 1 & . & . \\ d & . & 1 & 1 & . & 1 & 1 \\ e & . & . & . & 1 & . & 1 \\ f & 1 & 1 & . & 1 & 1 & . \end{array} \quad (3.1)$$

Підзадача 3.1.1. Знайти найкоротші відстані і вказати всі відповідні їм найкоротші шляхи від вершини a до всіх інших вершин графа G_1 (див. рис. 3.10).

В процесі реалізації алгоритму Дейкстри заповнюється таблиця, кількість рядків і стовпців якої визначається потужністю множини вершин графа, тобто 6×6 . У заголовках рядків таблиці вказуються вершини, до яких належить знайти найкоротшу відстань (табл. 3.1).

Таблиця 3.1 – Обчислення найкоротших відстаней в графі G_1 з вершини a

	1	2	3	4	5
	u = a r = 0	u = b r = 1	u = f r = 1	u = c r = 2	u = d r = 2
b	a, 1				
c	a, ∞	b, 2	b, 2		
d	a, ∞	b, 2	b, 2	b, 2	
e	a, ∞	a, ∞	f, 2	f, 2	f, 2
f	a, 1	a, 1			

Коментарі до заповнення табл. 3.1:

1) У заголовку першого стовпчика як пункт відправлення вказується початкова вершина $u = a$, пройдена відстань приймається рівною нулю: $r = 0$. Кожне ребро ідентифікується стовпчиковою (початок) і рядковою (кінцевою) вершинами і читається «стовпець – рядок». При заповненні першого стовпчика враховуються такі правила: початкова вершина a не змінюється і повторюється в кожному осередку; відстань збігається з довжиною ребра, яке

з'єднує початкову (стовпчикову) вершину з вершиною з рядка, якщо таке ребро не існує, то відстані присвоюється тимчасова мітка ∞ .

Після заповнення першого стовпчика з усіх кінцевих числових міток вибирається мінімальна – 1 (якщо таких декілька, то вибирається будь-яка з них). Їй відповідає постійно позначена вершина в рядку – b. Це означає, що найкоротша відстань до вершини b вже знайдена і дорівнює 1. Постійно позначена вершина приймається як новий відправний пункт $u=b$ і разом з мінімальною відстанню $r=1$ заноситься до заголовка чергового стовпчика.

2) При заповненні другого стовпчика таблиці числові мітки модифікуються згідно з правилом: до довжини ребра (b, j) додається пройдена відстань 1 і результат порівнюється з попередньою числовою міткою в рядку; якщо нова сумарна відстань менше попередньої, то вона приймається як поточна, при цьому вершина змінюється на вершину з поточного стовпчика. Так, ребро (b, c) має довжину 1, тоді пройдена відстань $r = 1 + 1 = 2$ має кінцеве значення, отже, вона приймається як поточна замість ∞ . При цьому вершина a змінюється на вершину b з поточного стовпчика. Ребро (b, f) має довжину 1, тоді пройдена відстань $r = 1 + 1 = 2 > 1$, тому в рядку f зберігається попередня числова мітка, позначення вершини при цьому також не змінюється.

З усіх кінцевих числових міток другого стовпчика вибирається мінімальна – 1. Їй відповідає вершина f. Тепер вершина $u = f$ є постійно поміченою, вона обирається як новий відправний пункт і запам'ятовується у заголовку третього стовпця одночасно з пройденою відстанню $r = 1$.

3) Третій стовпець заповнюється аналогічно попередньому. Тут модифікується мітка в рядку e. Всі числові мітки виявляються однаковими, тому обирається будь-яка. Нехай це буде $u = c$, яка тепер є постійно поміченою, розглядається як новий відправний пункт і записується до заголовку четвертого стовпчика разом з пройденою відстанню $r = 2$.

4) У четвертому стовпці мітки не змінюються, тому вибирається вершина $u=d$. Вона тепер постійно позначена, приймається як новий

відправний пункт і зберігається в заголовку у наступній колонці поряд з пройденою відстанню $r = 2$.

5) В п'ятому стовпці числа мітка в рядку e не змінюється, оскільки ребро (d, e) має довжину 1, і сумарна пройдена відстань більше попередньої: $1+2=3 > 2$. Таким чином, залишається єдина постійно позначена вершина. Отже, процес пошуку мінімальних відстаней завершується.

Примітки: 1) Якщо в деякому стовпці зустрічаються дві або більше вершини з однаковими мінімальними числовими позначками, то вибирається будь-яка з них. Це означає, що можливо існують дві різні ланцюги однакової довжини. 2) Якщо при обчисленні поточних числових міток нова сумарна відстань збігається з попередньою, то зберігається стара числова мітка. 3) Якщо граф орієнтований, то при обчисленні числових міток враховується напрямок ребра: ребро з вершини x_i до вершини x_j існує, якщо в графі є дуга (x_i, x_j) , Тоді можна врахувати її довжину. Але ребро (x_j, x_i) , при цьому може бути відсутнім, тоді відповідна позначка приймається рівною нескінченності. 4) Постійно помічені вершини в заголовках стовпців не повторюються. 5) Відстані в заголовках стовпців не зменшуються, тобто розташовуються в порядку \leq . 6) Поточні числові мітки в рядку не зростають (\geq).

Таким чином, з табл. 3.1 можна отримати інформацію про всі найкоротші ланцюги і відповідні їм довжини.

Наприклад, потрібно знайти найкоротший ланцюг з вершини a до вершини e . Послідовність вершин у ланцюзі виписується з кінця: останній заповнений осередок в рядку e містить інформацію про довжину найкоротшого ланцюга $r = 2$ і передостанній вершині в цьому ланцюзі – f . Інформація про попередню вершину знаходиться в останній комірці рядка f , в даному випадку – вершина a , яка є початком маршруту:

$$a \xrightarrow{1} f \xrightarrow{1} e \Rightarrow \text{dist} = 1 + 1 = 2.$$

Дані для всіх найкоротших ланцюгів представлені у табл. 3.2.

Таблиця 3.2 – Найкоротші ланцюга з вершини a для графа G1

Ланцюг	Довжина
$a \xrightarrow{1} b$	$r(a, b) = 1$
$a \xrightarrow{1} b \xrightarrow{1} c$	$r(a, c) = 1 + 1 = 2$
$a \xrightarrow{1} b \xrightarrow{1} d$	$r(a, d) = 1 + 1 = 2$
$a \xrightarrow{1} f \xrightarrow{1} e$	$r(a, e) = 1 + 1 = 2$
$a \xrightarrow{1} f$	$r(a, f) = 1$

Граф, який ілюструє дерево найкоротших ланцюгів з вершини a, зображений на рис. 3.11.

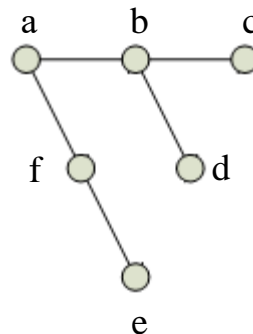


Рис. 3.11 – Дерево найкоротших ланцюгів з вершини a для графа G1

Оскільки всі ребра в графі (див. рис. 3.10) мають вагу 1, з табл. 3.1 видно, що відстані при обчисленні можуть збільшуватися тільки на 1. Тому фактично, як тільки нескінченна мітка змінилася на кінцеву числову мітку, вона згодом вже не модифікується, тобто не може зменшитися. Це означає, що відповідна найкоротша відстань між вершинами вже визначена. Тоді кількість складань і порівнянь в алгоритмі Дейкстри скорочується на $2 * 5 = 10$ операцій (табл. 3.3).

Таблиця 3.3 – Модифікована (скорочена) таблиця 3.1 для обчислення найкоротших відстаней в графі G_1 з вершини a

	1	2	3
$u = a$ $r = 0$	$u = b$ $r = 1$	$u = f$ $r = 1$	
b	a, 1		
c	a, ∞	b, 2	
d	a, ∞	b, 2	
e	a, ∞	a, ∞	f, 2
f	a, 1	a, 1	

Слід зауважити, що найкоротші маршрути з вершин a , c , e і дерева найкоротших шляхів будуть ідентичними.

Підзадача 3.1.2. Для графа на рис. 3.10 відшукати найкоротші відстані і вказати відповідні їм шляхи від вершини b до всіх інших його вершин.

В ході виконання алгоритму Дейкстри складається таблиця, розмір якої $b \times b$ (число рядків і стовпців) визначається кількістю вершин в графі. У заголовках рядків таблиці вказуються вершини, до яких належить знайти найкоротшу відстань (табл. 3.4).

Таблиця 3.4 – Обчислення найкоротших відстаней з вершини b для графа G_1

	1	2	3	4	5
$u = b$ $r = 0$	$u = a$ $r = 1$	$u = c$ $r = 1$	$u = d$ $r = 1$	$u = f$ $r = 1$	
a	b, 1				
c	b, 1	b, 1			
d	b, 1	b, 1	b, 1		
e	b, ∞	b, ∞	b, ∞	d, 2	d (f), 2
f	b, 1	b, 1	b, 1	b, 1	

Заповнення табл. 3.4 аналогічно попередньому (див. коментарі до табл. 3.1).

У табл. 3.5 представлені обчислення з урахуванням модифікації алгоритму Дейкстри. Тут видно, що на початковому етапі після розміщення міток в першому стовпці визначені найкоротші відстані від пункту

відправлення – вершини b – до інших вершин графа (пунктів прибуття), за винятком вершини e . Всі вони збігаються з довжиною ребра 1, а самі найкоротші ланцюга ідентичні з ребер, які з'єднують початковий пункт (вершину b) з кінцевими пунктами прибуття – вершинами a , c , d , f . Залишається знайти найкоротша відстань і ланцюг від початкового пункту b до пункту прибуття e . Поетапне виконання алгоритму показує, що з усіх кінцевих числових міток в першому стовпці можна вибрати будь-яку, оскільки всі вони дорівнюють 1. Для даного прикладу можна бачити, що вибір вершин f (або d) дозволить завершити пошук ще за один прохід (див. табл. 3.5) з тим же результатом, що і в табл. 3.4.

Таблиця 3.5 – Модифікована (скорочена) таблиця 3.4 найкоротших відстаней з вершини b для графа $G1$

	1	2
	u = b r = 0	u = f r = 1
a	b, 1	b, 1
c	b, 1	b, 1
d	b, 1	b, 1
e	b, ∞	f, 2
f	b, 1	

Дані для всіх найкоротших ланцюгів представлені в табл. 3.6.

Таблиця 3.6 – Найкоротші ланцюга з вершини b для графа $G1$

Ланцюг	Довжина
$b \xrightarrow{1} a$	$r(b, a) = 1$
$b \xrightarrow{1} c$	$r(b, c) = 1$
$b \xrightarrow{1} d$	$r(b, d) = 1$
$b \xrightarrow{1} d \xrightarrow{1} e$	$r(b, e) = 1 + 1 = 2$
$b \xrightarrow{1} f$	$r(b, f) = 1$

Граф, який ілюструє дерево найкоротших ланцюгів з вершини b , наведено на рис. 3.12.

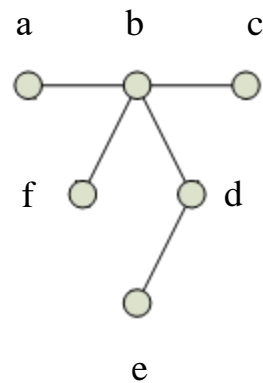


Рис. 3.12 – Граф (дерево) найкоротших ланцюгів з вершини b для графа $G1$

Слід зауважити, що найкоротші ланцюга з вершин b , d , f і дерева найкоротших шляхів будуть ідентичними, відповідно.

Матриця найкоротших відстаней між усіма парами вершин графа $G1$ представлена нижче:

$$\text{Dist}_1 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & 2 & 2 & 2 & 1 \\ b & 1 & . & 1 & 1 & 2 & 1 \\ c & 2 & 1 & . & 1 & 2 & 2 \\ d & 2 & 1 & 1 & . & 1 & 1 \\ e & 2 & 2 & 2 & 1 & . & 1 \\ f & 1 & 1 & 2 & 1 & 1 & . \end{array} \quad (3.2)$$

3.3 Метод пошуку найкоротших відстаней між парами вершин

Задача 3.2. Для графа $G2$ з одиничними ваговими коефіцієнтами, представленого на діаграмі (рис. 3.13, а) з відповідною матрицею суміжності (рис. 3.13, б), знайти всі найкоротші відстані між усіма парами вершин.

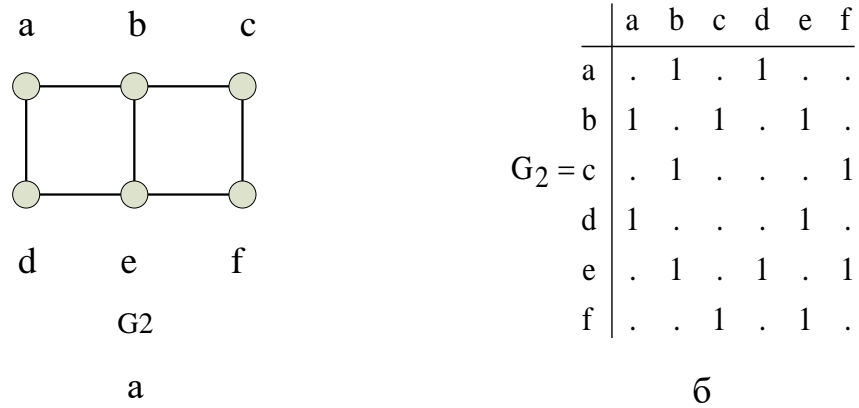


Рис. 3.13 – Граф G2 (а) і його матриця суміжності (б)

Підзадача 3.2.1. Знайти відстані і вказати всі найкоротші шляхи від вершини а до решти вершин графа G2.

В процесі реалізації алгоритму Дейкстри заповнюється таблиця, кількість рядків і стовпців якої визначається потужністю безлічі вершин графа (6x6). У заголовках рядків таблиці вказуються вершини, до яких належить знайти найкоротша відстань (табл. 3.7).

Таблиця 3.7 – Обчислення найкоротших відстаней з вершини а для графа G2

	1	2	3	4	5
	u = a	u = b	u = d	u = c	u = e
	r = 0	r = 1	r = 1	r = 2	r = 2
b	a, 1				
c	a, ∞	b, 2	b, 2		
d	a, 1	a, 1			
e	a, ∞	b, 2	b, 2	b, 2	
f	a, ∞	a, ∞	a, ∞	c, 3	c, 3

У табл. 3.8 наведені обчислення з урахуванням модифікації алгоритму. Видно, що кількість операцій в даному випадку зменшується несуттєво: на одну ітерацію – порівняння.

Таблиця 3.8 – Обчислення найкоротших відстаней з вершини а для графа G2 з урахуванням модифікованого алгоритму

	1	2	3	4
$u = a$ $r = 0$		$u = b$ $r = 1$	$u = d$ $r = 1$	$u = c$ $r = 2$
b	a, 1			
c	a, ∞	b, 2	b, 2	
d	a, 1	a, 1		
e	a, ∞	b, 2	b, 2	b, 2
f	a, ∞	a, ∞	a, ∞	c, 3

Дані для всіх найкоротших ланцюгів представлені в табл. 3.9.

Таблиця 3.9 – Найкоротші ланцюга, відновлені по таблиці 3.8

Ланцюг	Довжина
$a \xrightarrow{1} b$	$r(a, b) = 1$
$a \xrightarrow{1} b \xrightarrow{1} c$	$r(a, c) = 1 + 1 = 2$
$a \xrightarrow{1} d$	$r(a, d) = 1$
$a \xrightarrow{1} b \xrightarrow{1} e$	$r(a, e) = 1 + 1 = 2$
$a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{1} f$	$r(a, f) = 1 + 1 + 1 = 3$

Граф, який ілюструє дерево найкоротших ланцюгів з вершини а, представлений на рис. 3.14.

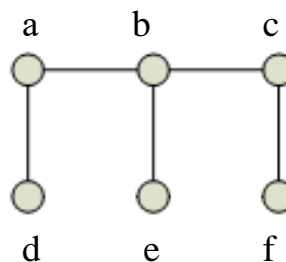


Рис. 3.14 – Дерево найкоротших ланцюгів з вершини а для графа G2

Слід зауважити, що найкоротші маршрути з вершин c , d , f і дерева найкоротших шляхів будуть ідентичними тим, що побудовані з вершини a .

Підзадача 3.2.2. Для графа G_2 , представленого на рис. 3.13, необхідно знайти найкоротші відстані і вказати всі найкоротші шляхи від вершини b до всіх інших вершин. У заголовках рядків таблиці вказуються вершини – пункти прибуття, до яких належить знайти найкоротші відстані і маршрути (табл. 3.10).

Таблиця 3.10 – Обчислення найкоротших відстаней з вершини b графа G_2

	1	2	3	4	5
	$u = b$ $r = 0$	$u = a$ $r = 1$	$u = c$ $r = 1$	$u = e$ $r = 1$	$u = d$ $r = 2$
a	$b, 1$				
c	$b, 1$	$b, 1$			
d	b, ∞	$a, 2$	$a, 2$	$a, 2$	
e	$b, 1$	$b, 1$	$b, 1$		
f	b, ∞	b, ∞	$c, 2$	$c, 2$	$c, 2$

Обчислення з урахуванням модифікації алгоритму наведені в табл. 3.11.

Таблиця 3.11 – Скорочена таблиця 3.10 з урахуванням модифікації алгоритму

	1	2	3
	$u = b$ $r = 0$	$u = a$ $r = 1$	$u = c$ $r = 1$
a	$b, 1$		
c	$b, 1$	$b, 1$	
d	b, ∞	$a, 2$	$a, 2$
e	$b, 1$	$b, 1$	$b, 1$
f	b, ∞	b, ∞	$c, 2$

Дані для всіх найкоротших ланцюгів представлені в табл. 3.12.

Таблиця 3.12 – Найкоротші ланцюга з вершини b для графа G_2

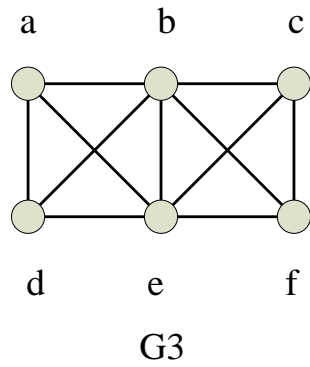
Ланцюг	Довжина
$b \xrightarrow{1} a$	$r(b,a) = 1$
$b \xrightarrow{1} c$	$r(b,c) = 1$
$b \xrightarrow{1} a \xrightarrow{1} d$	$r(b,d) = 1+1 = 2$
$b \xrightarrow{1} e$	$r(b,e) = 1$
$b \xrightarrow{1} c \xrightarrow{1} f$	$r(b,f) = 1+1 = 2$

Дерево найкоротших ланцюгів з вершини b буде таким самим, як і на рис. 3.14. Матриця найкоротших відстаней між усіма парами вершин графа G_2 має вигляд:

$$\text{Dist}_2 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & 2 & 1 & 2 & 3 \\ b & 1 & . & 1 & 2 & 1 & 2 \\ c & 2 & 1 & . & 3 & 2 & 1 \\ d & 1 & 2 & 3 & . & 1 & 2 \\ e & 2 & 1 & 2 & 1 & . & 1 \\ f & 3 & 2 & 1 & 2 & 1 & . \end{array} \quad (3.3)$$

3.4 Пошук мінімальних шляхів для графа з одиничними дугами

Задача 3.3. Для графа з одиничними ваговими коефіцієнтами, представленого на діаграмі (рис. 3.15, а) з відповідною матрицею суміжності (рис. 3.15, б), необхідно знайти найкоротші відстані між усіма парами вершин.



$$G_3 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & . & 1 & 1 & . \\ b & 1 & . & 1 & 1 & 1 & 1 \\ c & . & 1 & . & . & 1 & 1 \\ d & 1 & 1 & . & . & 1 & . \\ e & 1 & 1 & 1 & 1 & . & 1 \\ f & . & 1 & 1 & . & 1 & . \end{array}$$

а

б

Рис. 3.15 – Граф G₃ (а) і його матриця суміжності (б)

Підзадача 3.3.1. Знайти відстані і вказати всі найкоротші шляхи від вершини а до решти вершин графа G₃. В процесі реалізації алгоритму Дейкстри заповнюється таблиця, кількість рядків і стовпців якої визначається потужністю множини вершин графа, тобто 6х6. У заголовках рядків таблиці вказуються вершини, до яких належить знайти найкоротшу відстань (таблиця 3.13).

Таблиця 3.13 – Обчислення найкоротших відстаней з вершини а для графа G₃

	1	2	3	4	5
	u = a r = 0	u = b r = 1	u = d r = 1	u = e r = 1	u = f r = 1
b	a, 1				
c	a, ∞	b, 2	b, 2	b, 2	b, 2
d	a, 1	a, 1	a, 1		
e	a, 1	a, 1	a, 1		
f	a, ∞	b, 2	b, 2	b, 2	

Модифіковані обчислення наведені в табл. 3.14.

Таблиця 3.14 – Скорочена таблиця найкоротших відстаней за модифікованим алгоритмом

X	1	2
	u = a r = 0	u = b r = 1
b	a, 1	/
c	a, ∞	b, 2
d	a, 1	a, 1
e	a, 1	a, 1
f	a, ∞	b, 2

Дані для всіх найкоротших ланцюгів в графі з вершини a наведені в табл. 3.15.

Таблиця 3.15 – Найкоротші ланцюги з вершини a графа G3

Ланцюг	Довжина
$a \xrightarrow{1} b$	$r(a, b) = 1$
$a \xrightarrow{1} b \xrightarrow{1} c$	$r(a, c) = 1 + 1 = 2$
$a \xrightarrow{1} d$	$r(a, d) = 1$
$a \xrightarrow{1} e$	$r(a, e) = 1$
$a \xrightarrow{1} b \xrightarrow{1} f$	$r(a, f) = 1 + 1 = 2$

Граф, який ілюструє дерево найкоротших ланцюгів з вершини a, що відповідають найкоротшим маршрутам із заданого пункту відправлення в усі можливі пункти прибуття, представлений на рис. 3.16.

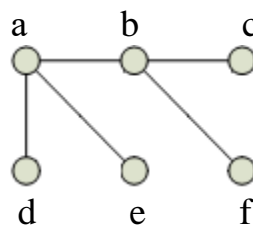


Рис. 3.16 – Дерево найкоротших ланцюгів з вершини a

Слід зауважити, що найкоротші маршрути з вершин c , d , f і дерева найкоротших шляхів будуть ідентичними тим, що побудовані з вершини a .

Підзадача 3.3.2. Для графа на рис. 3.15, а знайти найкоротші відстані і всі відповідні шляхи від вершини b до всіх інших вершин. У заголовках рядків таблиці вказуються вершини (пункти прибуття), до яких належить знайти найкоротші відстані (табл. 3.16).

Таблиця 3.16 – Обчислення найкоротших відстаней з вершини b графа G_3

X	1
	u = b r = 0
a	b, 1
c	b, 1
d	b, 1
e	b, 1
f	b, 1

Stop

З табл. 3.16 видно, що алгоритм Дейкстри зупинений після першого проходу, оскільки всі найкоротші відстані визначені на початковому етапі розміщення міток (1-й стовпець). Дані для всіх найкоротших ланцюгів з вершини b представлені в табл. 3.17.

Таблиця 3.17 – Найкоротші ланцюга з вершини b графа G_3

Ланцюг	Довжина
$b \xrightarrow{1} a$	$r(b, a) = 1$
$b \xrightarrow{1} c$	$r(b, c) = 1$
$b \xrightarrow{1} d$	$r(b, d) = 1$
$b \xrightarrow{1} e$	$r(b, e) = 1$
$b \xrightarrow{1} f$	$r(b, f) = 1$

Дерево найкоротших ланцюгів з вершини b має топологію типу «зірка» і представлено на рис. 3.17.

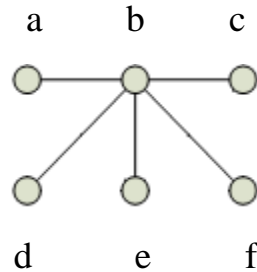


Рис. 3.17 – Дерево найкоротших ланцюгів з вершини b

Матриця найкоротших відстаней між усіма парами вершин графа G_3 має вигляд:

$$\text{Dist}_3 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & 2 & 1 & 1 & 2 \\ b & 1 & . & 1 & 1 & 1 & 1 \\ c & 2 & 1 & . & 2 & 1 & 1 \\ d & 1 & 1 & 2 & . & 1 & 2 \\ e & 1 & 1 & 1 & 1 & . & 1 \\ f & 2 & 1 & 1 & 2 & 1 & . \end{array} \quad (3.4)$$

Дерево найкоротших ланцюгів для вершини b буде ілюструватися графом, аналогічним графу на рис. 3.17, все найкоротші відстані також будуть дорівнювати 1.

Опис модифікованого алгоритму Дейкстри.

Кожній вершині з множини V ставиться у відповідність мітка, яка визначає мінімальну відоме відстань від цієї вершини до початкової вершини a (пункту відправлення). Алгоритм виконується крок за кроком. На кожному кроці він «відвідує» одну вершину і намагається зменшити числові мітки. Реалізація алгоритму завершується, коли всі вершини відвідані.

Ініціалізація. Числова мітка самої вершини a покладається рівною 0, іншим вершинам присвоюється тимчасова мітка – нескінченність. Це означає, що відстані від a до інших вершин поки невідомі. Всі вершини графа позначаються як невідвідані.

Крок алгоритму. Якщо всі вершини відвідані, алгоритм завершує роботу. В іншому випадку, з ще не відвіданих вершин вибирається вершина u з мінімальною міткою. При цьому розглядаються всі можливі маршрути, де u є передостаннім пунктом. Вершини, в які ведуть ребра з u , називаються сусідніми по відношенню до u . Для кожного сусіда вершини u , крім позначених як відвідані, розглядається нова довжина шляху, що дорівнює сумі значень поточної мітки вершини u і довжини ребра, що з'єднує u з цим сусідом.

У традиційному алгоритмі Дейкстри далі враховується такий крок: якщо отримане значення довжини менше значення мітки сусіда, то мітка сусіда замінюється отриманим значенням довжини.

Для графів з ребрами одиничної довжини (ваги) сума відстаней кожен раз може збільшуватися тільки на 1. Тому в згаданому пункті тільки нескінченні мітки сусіда можуть змінюватися на кінцеві числові мітки, які згодом не змінюються, тобто зменшуватися вже не можуть. Відповідні відстані є числами натурального ряду. З цієї причини порівняння доцільно проводити тільки з метою визначення кінцевих числових міток для тих вершин, які таких поки не мають, тобто їх тимчасові мітки рівні нескінченності. Якщо не існує ребра, яке з'єднує постійно позначену вершину з вершиною, що має нескінчену мітку, то як черговий пункт вибирається постійно позначена вершина з мінімальною міткою у поточному стовпці (як і раніше), що дозволяє реалізувати спробу знайти мінімальний маршрут через іншу вершину.

Після розгляду всіх сусідів вершина u позначається як відвідана і крок алгоритму повторюється.

LABEL – масив для зберігання поточних відміток.

PERM – масив для вказівки постійно помічених вершин (вершини стають постійно поміченими, коли вони виявляються рівними u_i для будь-якого i).

Якщо $PERM(v) = 1$, то v – постійно позначена вершина і її мітка дорівнює $d(s, v)$. Спочатку $PERM(s)=1$ і $PERM(v)=0$ при $v \neq s$. PRED – масив покажчиків на вершини, з яких здійснено перехід до вершин з постійною міткою. Якщо вершина v позначена постійною міткою, то послідовність $v, PRED(v), PRED(PRED(v)), \dots, s$ – вершини, які складають найкоротший орієнтований шлях з s до v .

1. Початок. Покласти $LABEL(s) = 0, PERM(s) = 1, PRED(s) = s; \forall v \neq s$ покласти $LABEL(v) = \infty, PERM(v) = 0, PRED(v) = v$.

2. Нехай $i = 0, u = s$ (u – остання з вершин з незмінною міткою. Тепер це вершина s).

3. Обчислення LABEL і зміна елементів масиву PRED. Покласти $i=i+1$. Виконати для кожної вершини v з нескінченною міткою наступні дії (у традиційному алгоритмі Дейкстри цей пункт виконувався для всіх вершин v , крім вершин з незмінною міткою, а в модифікованому алгоритмі він застосовується тільки до вершин з тимчасовими мітками $LABEL(v) = \infty$, Оскільки інші мітки модифікуватися не будуть):

3.1. Покласти $M = \min \{LABEL(v), LABEL(u) + w(u, v)\}$, де $w(u, v)=1$ – довжина ребра, що з'єднує вершини u і v , якщо таке існує, інакше (т.е. коли не існує ребра (u, v)) – як постійно помічена обирається вершина з мінімальною кінцевою числовою міткою, а якщо таких кілька, то вибирається одна з них);

3.2. Якщо $M < LABEL(v)$, то покласти $LABEL(v) = M, PRED(v) = u$.

4. Виділення вершини u_i . Серед всіх вершин, що не позначені незмінною міткою, знайти вершину w з найменшою міткою (якщо таких вершин кілька, то вибір можна зробити довільно). Покласти $PERM(w) = 1, u_i = w$ (w – останньою вершиною з незмінною міткою).

5. Якщо $i < n-1$, то повернення до кроку 3, інакше – кінець (всі найкоротші шляхи знайдені).

Мітки вершин являють собою довжини найкоротших шляхів; v , PRED(v), PRED(PRED(v)), ..., s – є вершини найкоротшого орієнтованого st шляху.

Таким чином, наведена модифікація алгоритму Дейкстри для неорієнтованих зважених графів довжиною в один символ ребра, що дозволяє скоротити кількість складань і порівнянь за рахунок виключення з цього процесу вже знайдених на попередньому етапі кінцевих числових міток, які в подальшому не можуть зменшуватися, а залишаються константами, але можливості перетворення тільки нескінченних міток сусіда в кінцеві числові мітки.

3.5 Визначення мінімальних шляхів у трикутній топології

Задача 3.4. Для графа G_4 з 21 вершини, пов'язаних трикутної топологією (рис. 3.18), визначити всі найкоротші шляхи і відповідні їм відстані з вершини v_1 (на рис. 3.18 – коротко 01) до всіх інших вершин.

Рішення. Розглянутий на рис. 3.18 граф G_4 є неорієнтованим. Застосування алгоритму Дейкстри дозволяє знайти рішення поставленого завдання.

Матриця суміжності графа G_4 має вигляд:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
8	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0
9	0	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0
12	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0
13	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0
14	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0
15	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1
16	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

(3.5)

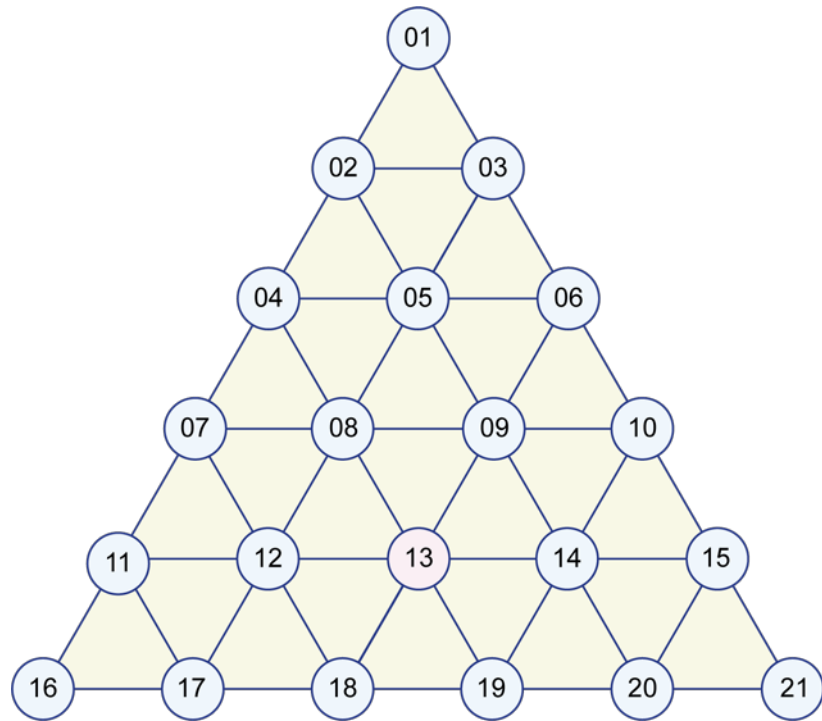


Рис. 3.18 – Граф G4 – конфігурація мережі

В процесі реалізації алгоритму Дейкстри заповнюється таблиця, кількість рядків і стовпців якої визначається потужністю множини вершин графа, тобто 21×21 . У заголовках рядків таблиці вказуються вершини, до яких належить знайти найкоротшу відстань (табл. 3.18).

Таблиця 3.18 – Обчислення найкоротших відстаней з вершини v_1 графа G_4

G_4	$u=v_1$ $r=0$	$u=v_2$ $r=1$	$u=v_3$ $r=1$	$u=v_4$ $r=2$	$u=v_5$ $r=2$	$u=v_6$ $r=2$	$u=v_7$ $r=3$	$u=v_8$ $r=3$	$u=v_9$ $r=3$	$u=v_{10}$ $r=3$	$u=v_{11}$ $r=4$	$u=v_{12}$ $r=4$	$u=v_{13}$ $r=4$	$u=v_{14}$ $r=4$	$u=v_{15}$ $r=5$
v_2	$v_{1,1}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—
v_3	$v_{1,1}$	$v_{1,1}$	—	—	—	—	—	—	—	—	—	—	—	—	—
v_4	$v_{1,\infty}$	$v_{2,2}$	$v_{2,2}$	—	—	—	—	—	—	—	—	—	—	—	—
v_5	$v_{1,\infty}$	$v_{2,2}$	$v_{2,2}$	$v_{2,2}$	—	—	—	—	—	—	—	—	—	—	—
v_6	$v_{1,\infty}$	$v_{1,\infty}$	$v_{3,2}$	$v_{3,2}$	$v_{3,2}$	—	—	—	—	—	—	—	—	—	—
v_7	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{4,3}$	$v_{4,3}$	$v_{4,3}$	—	—	—	—	—	—	—	—	—
v_8	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{4,3}$	$v_{4,3}$	$v_{4,3}$	$v_{4,3}$	—	—	—	—	—	—	—	—
v_9	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{5,3}$	$v_{5,3}$	$v_{5,3}$	$v_{5,3}$	—	—	—	—	—	—	—
v_{10}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{6,3}$	$v_{6,3}$	$v_{6,3}$	$v_{6,3}$	—	—	—	—	—	—
v_{11}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{7,4}$	$v_{7,4}$	$v_{7,4}$	$v_{7,4}$	—	—	—	—	—
v_{12}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{7,4}$	$v_{7,4}$	$v_{7,4}$	$v_{7,4}$	$v_{7,4}$	—	—	—	—
v_{13}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{8,4}$	$v_{8,4}$	$v_{8,4}$	$v_{8,4}$	$v_{8,4}$	—	—	—
v_{14}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{9,4}$	$v_{9,4}$	$v_{9,4}$	$v_{9,4}$	$v_{9,4}$	—	—
v_{15}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{10,4}$	$v_{10,4}$	$v_{10,4}$	$v_{10,4}$	$v_{10,4}$	—
v_{16}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$
v_{17}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$
v_{18}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{12,5}$	$v_{12,5}$	$v_{12,5}$	$v_{12,5}$
v_{19}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{13,5}$	$v_{13,5}$	$v_{13,5}$
v_{20}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{14,5}$	$v_{14,5}$
v_{21}	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{1,\infty}$	$v_{15,5}$

Коментарі до заповнення табл. 3.18:

1) У заголовку першого стовпчика як пункт відправлення вказується початкова вершина $u = v_1$, пройдена відстань приймається рівною нулю: $r = 0$. Кожне ребро ідентифікується стовпчиковою (початок) і рядковою (кінець) вершинами і читається «стовпець-рядок». При заповненні першого стовпчика враховуються такі правила: початкова вершина v_1 не змінюється і повторюється в кожному осередку; відстань збігається з довжиною ребра

(дорівнює 1), що з'єднує початкову (стовпчикову) вершину з вершиною з рядка. За відсутністю такого ребра відстані присвоюється тимчасова мітка ∞ .

Після заповнення першого стовпчика з усіх кінцевих числових міток вибирається мінімальна – 1 (якщо таких декілька, то вибирається будь-яка з них). Їй відповідає постійно позначена вершина у рядку – v_2 . Це означає, що найкоротша відстань до вершини v_2 вже знайдена і дорівнює 1. Постійно позначена вершина приймається як новий відправний пункт $u = v_2$ і разом з мінімальним відстанню $r = 1$ заноситься до заголовку чергового стовпчика.

2) При заповненні другого стовпчика таблиці числові мітки модифікуються згідно з правилом: до довжини ребра (v_2, j) додається пройдена відстань 1 і результат порівнюється з попередньою числовою міткою в рядку; якщо нова сумарна відстань менше попередньої, то вона приймається як поточна, при цьому вершина змінюється на вершину з поточного стовпчика. Так, ребро (v_2, v_4) має довжину 1, тоді пройдена відстань $r = 1 + 1 = 2$ має кінцеве значення, отже, воно приймається як поточна замість ∞ . При цьому вершина v_1 змінюється на вершину v_2 з поточного стовпчика.

Ребро (v_2, v_3) має довжину 1, тоді пройдена відстань $r = 1 + 1 = 2 > 1$, тому в рядку v_3 зберігається попередня числова мітка, позначення вершини при цьому також не змінюється. *Слід зауважити, що алгоритм Дейкстри в цьому випадку зазнає такої модифікації: якщо кінцева числова мітка вже визначена на поточному етапі, далі вона вже не змінюється (оскільки може тільки збільшуватися), тобто мінімальна відстань (довжина ланцюга) вже виявляється визначеною цим значенням числової мітки. Таким чином, виконувати додавання і порівняння для вершин з кінцевими числовими позначками на наступних етапах не має сенсу, їх слід розглядати по черзі (в порядку зростання індексів) і вибирати на кожній наступній ітерації як постійно помічені, поки всі вони не будуть вичерпані.*

З усіх кінцевих числових міток другого стовпчика вибирається мінімальна – 1. Їй відповідає вершина v_3 . Тепер вершина $u = v_3$ є постійно

поміченою, вона вибирається як новий відправний пункт і запам'ятовується у заголовку третього стовпця одночасно з пройденою відстанню $r = 1$.

3) Третій стовпець заповнюється аналогічно попередньому. Тут модифікується мітка в рядку v_6 . Всі числові мітки виявляються однаковими, тому вибирається перша за порядком, а саме v_4 . Тепер $u=v_4$ є постійно поміченою, розглядається як новий відправний пункт і записується до заголовку четвертого стовпчика разом з пройденою відстанню $r = 2$.

4) У четвертому стовпці не змінюються постійні мітки з рядків v_5 і v_6 , рівні 2. При цьому модифікуються мітки в рядках v_7 і v_8 . Тому вибирається вершина $u=v_5$, яка тепер постійно позначена, приймається як новий відправний пункт і зберігається в заголовку у наступній колонці поряд з пройденою відстанню $r = 2$.

5) В п'ятому стовпці числова мітка 2 в рядку v_6 не змінюється і є мінімальною. При цьому з'являється кінцева мітка 3 в рядку v_9 . Таким чином, як постійно помічена вершина тепер вибирається v_6 і заноситься до заголовку шостого стовпчика разом з пройденою відстанню 2.

6) У шостому стовпці з'являється нова кінцева числова мітка 3 в рядку v_{10} , інші мітки не змінюються. Як постійно помічена вершина тепер вибирається v_7 і заноситься до заголовку сьомого стовпчика разом з пройденою відстанню 3.

7) У сьомому стовпці з'являються дві нові кінцеві числові мітки 4 в рядках v_{11} і v_{12} , інші мітки не змінюються. Як постійно помічену слід вибрати вершину v_8 , яка разом з пройденою відстанню 3 заноситься до заголовку восьмого стовпця.

8) У восьмому стовпці з'являється одна нова кінцева числова мітка 4 в рядку v_{13} , інші мітки не змінюються. Тепер як постійно помічену слід вибрати вершину v_9 , яка разом з пройденою відстанню 3 заноситься до заголовку дев'ятого стовпчика.

9) У дев'ятому стовпці додається ще одна нова кінцева числова мітка 4 в рядку v_{14} , інші мітки не змінюються, тому як постійно помічена вибирається

вершина v_{10} , яка разом з пройденою відстанню 3 заноситься до заголовку десятого стовпчика.

10) У десятому стовпці додається ще одна нова кінцева числова мітка 4 в рядку v_{15} , інші позначки не модифікуються, тому як постійно помічена вибирається вершина v_{11} , яка разом з пройденою відстанню 4 заноситься до заголовку одинадцятого стовпця.

11) У одинадцятому стовпці з'являються відразу дві нові кінцеві числові мітки 5 в рядках v_{16} і v_{17} , інші позначки не модифікуються. Як постійно помічена вибирається вершина v_{12} , яка разом з пройденою відстанню 4 заноситься до заголовку дванадцятого стовпчика.

12) У дванадцятому стовпчику з'являється ще одна числова мітка 5 в рядку v_{18} , інші мітки не змінюються. Як постійно помічена вибирається вершина v_{13} , яка разом з пройденою відстанню 4 заноситься до заголовку тринадцятого стовпчика.

13) У тринадцятому стовпці з'являється ще одна числова мітка 5 в рядку v_{19} , інші мітки не змінюються. Як постійно помічена вибирається вершина v_{14} , яка разом з пройденою відстанню 4 заноситься до заголовку чотирнадцятого стовпчика.

14) У чотирнадцятому стовпці з'являється ще одна числова мітка 5 в рядку v_{20} , інші мітки не змінюються. Як постійно помічена вибирається вершина v_{15} , яка разом з пройденою відстанню 4 заноситься у заголовок п'ятнадцятого стовпчика.

15) У п'ятнадцятому стовпці з'являється кінцева числова мітка 5 в рядку v_{21} , інші мітки не змінюються. Всі рядки мають кінцеві числові мітки. Отже, процес пошуку мінімальних відстаней завершено.

Таким чином, з табл. 3.18 можна отримати інформацію про всі найкоротші ланцюги і відповідні їм довжини.

Наприклад, потрібно знайти найкоротший ланцюг з вершини v_1 до вершину v_{21} . Послідовність вершин у ланцюзі випикується з кінця: остання заповнена осередок в рядку v_{21} містить інформацію про довжину

найкоротшого ланцюга $r_{\min} = 5$ і передостанній вершині в цьому ланцюзі – v_{15} . Інформація про попередню вершину знаходиться в останній комірці рядка v_{15} , в даному випадку – вершина v_{10} , далі – v_6 , v_3 і v_1 , яка є початком маршруту:

$$v_1 \xrightarrow{1} v_3 \xrightarrow{1} v_6 \xrightarrow{1} v_{10} \xrightarrow{1} v_{15} \xrightarrow{1} v_{21}, \text{ dist} = 5.$$

Дані всіх найкоротших ланцюгів представлені в табл. 3.19. Граф, який ілюструє дерево найкоротших ланцюгів з вершини v_1 , представлений на рис. 3.19.

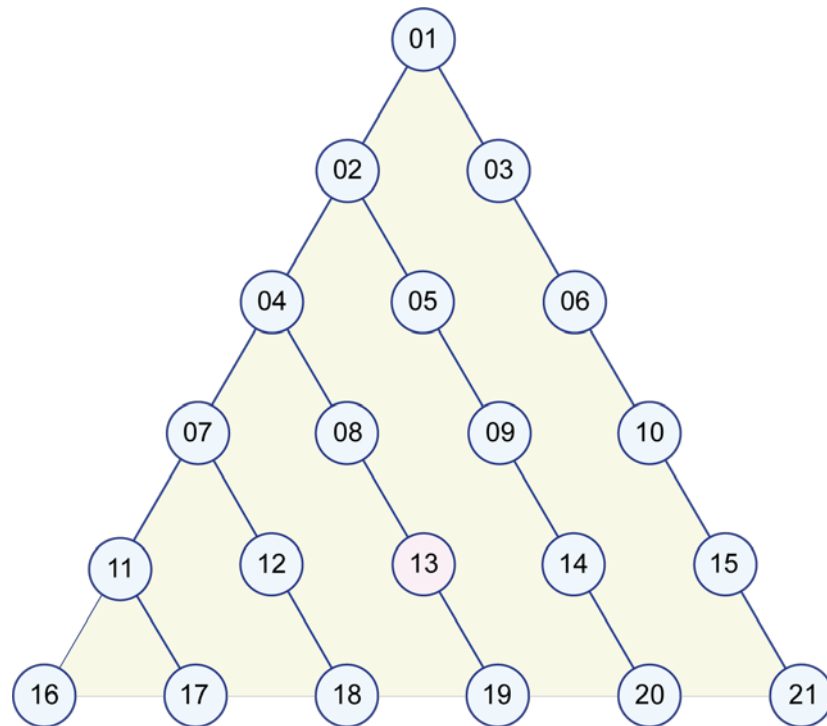


Рис. 3.19 – Дерево найкоротших ланцюгів з вершини v_1

Таблиця 3.19 – Найкоротші ланцюги з вершини v_1 графа G_4

Ланцюг	Довжина
$\overset{1}{v_1} \rightarrow v_2, r_{\min} = 1$	$r(v_1, v_2) = 1$
$\overset{1}{v_1} \rightarrow v_3, r_{\min} = 1$	$r(v_1, v_3) = 1$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow v_4, r_{\min} = 2$	$r(v_1, v_4) = 2$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow v_5, r_{\min} = 2$	$r(v_1, v_5) = 2$
$\overset{1}{v_1} \rightarrow \overset{1}{v_3} \rightarrow v_6, r_{\min} = 2$	$r(v_1, v_6) = 2$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow v_7, r_{\min} = 3$	$r(v_1, v_7) = 3$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow v_8, r_{\min} = 3$	$r(v_1, v_8) = 3$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_5} \rightarrow v_9, r_{\min} = 3$	$r(v_1, v_9) = 3$
$\overset{1}{v_1} \rightarrow \overset{1}{v_3} \rightarrow \overset{1}{v_6} \rightarrow v_{10}, r_{\min} = 3$	$r(v_1, v_{10}) = 3$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_7} \rightarrow v_{11}, r_{\min} = 4$	$r(v_1, v_{11}) = 4$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_7} \rightarrow v_{12}, r_{\min} = 4$	$r(v_1, v_{12}) = 4$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_8} \rightarrow v_{13}, r_{\min} = 4$	$r(v_1, v_{13}) = 4$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_5} \rightarrow \overset{1}{v_9} \rightarrow v_{14}, r_{\min} = 4$	$r(v_1, v_{14}) = 4$
$\overset{1}{v_1} \rightarrow \overset{1}{v_3} \rightarrow \overset{1}{v_6} \rightarrow \overset{1}{v_{10}} \rightarrow v_{15}, r_{\min} = 4$	$r(v_1, v_{15}) = 4$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_7} \rightarrow \overset{1}{v_{11}} \rightarrow v_{16}, r_{\min} = 5$	$r(v_1, v_{16}) = 5$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_7} \rightarrow \overset{1}{v_{11}} \rightarrow v_{17}, r_{\min} = 5$	$r(v_1, v_{17}) = 5$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_7} \rightarrow \overset{1}{v_{12}} \rightarrow v_{18}, r_{\min} = 5$	$r(v_1, v_{18}) = 5$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_4} \rightarrow \overset{1}{v_8} \rightarrow \overset{1}{v_{13}} \rightarrow v_{19}, r_{\min} = 5$	$r(v_1, v_{19}) = 5$
$\overset{1}{v_1} \rightarrow \overset{1}{v_2} \rightarrow \overset{1}{v_5} \rightarrow \overset{1}{v_9} \rightarrow \overset{1}{v_{14}} \rightarrow v_{20}, r_{\min} = 5$	$r(v_1, v_{20}) = 5$
$\overset{1}{v_1} \rightarrow \overset{1}{v_3} \rightarrow \overset{1}{v_6} \rightarrow \overset{1}{v_{10}} \rightarrow \overset{1}{v_{15}} \rightarrow v_{21}, r_{\min} = 5$	$r(v_1, v_{21}) = 5$

Оскільки всі ребра в графі (див. рис. 3.18) мають вагу 1, з табл. 3.18 видно, що відстані при обчисленні можуть збільшуватися тільки на 1. Тому фактично, як тільки нескінченна мітка змінилася на кінцеву числову мітку, вона згодом вже не модифікується, тобто не може зменшитися. Це означає, що відповідна найкоротша відстань між вершинами вже визначена. Тоді кількість складань і порівнянь в алгоритмі Дейкстри можна скоротити за рахунок вилучення порівнянь вже з отриманими кінцевими числовими позначками та послідовного вибору постійно помічених вершин зі списку тих, що мають мінімальні числові мітки, до їх вичерпання з подальшим переходом до міток на одиницю більших. Час реалізації алгоритму скорочується на 30%.

3.6 Пошук мінімальних шляхів у прямокутній топології

Задача 3.5. Визначити всі найкоротші шляхи і відстані з вершини v_1 (на рис. 3.20 – коротко 01) до всіх інших вершин прямокутної топології графа G_5 , що містить 21 вершину.

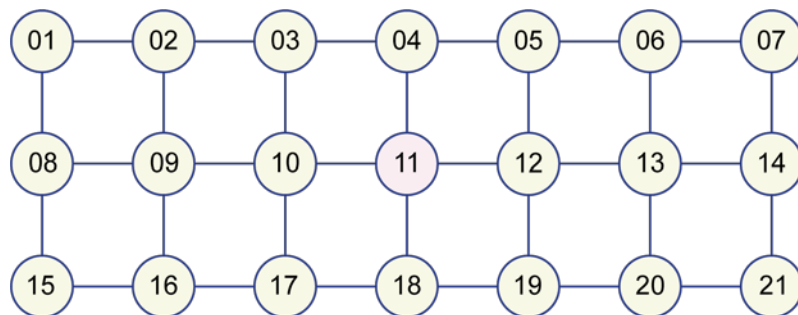


Рис. 3.20 – Граф G_5 – конфігурація мережі

Розв'язок. Розглянутий на рис. 3.20 граф G_5 є неорієнтованим. Застосування алгоритму Дейкстри дозволяє знайти рішення поставленого завдання. Матриця суміжності графа G_5 має вигляд:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
9	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
10	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0
11	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
12	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
13	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
14	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
16	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0
18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

(3.6)

В процесі реалізації алгоритму Дейкстри заповнюється таблиця, кількість рядків і стовпців якої визначається потужністю множини вершин графа, тобто 21×21 . У заголовках рядків таблиці вказуються вершини, до яких належить знайти найкоротші відстані (табл. 3.20).

Коментарі до заповнення табл. 3.20:

1) У заголовку першого стовпчика як пункт відправлення вказується початкова вершина $u = v_1$, пройдена відстань приймається рівною нулю: $r = 0$. Кожне ребро ідентифікується стовпчиковою (початок) і рядковою (кінець) вершинами і читається «стовпець-рядок». При заповненні першого стовпчика враховуються такі правила: початкова вершина v_1 не змінюється і повторюється в кожному осередку; відстань збігається з довжиною ребра (дорівнює 1), що з'єднує початкову (стовпчикову) вершину з вершиною з рядка. За відсутністю такого ребра відстані присвоюється тимчасова мітка ∞ .

Після заповнення першого стовпчика з усіх кінцевих числових міток вибирається мінімальна – 1 (якщо таких декілька, то вибирається будь-яка з них). Їй відповідає постійно позначена вершина в рядку – v_2 . Це означає, що найкоротшу відстань до вершини v_2 вже знайдено і вона дорівнює 1. Постійно позначена вершина приймається як новий відправний пункт $u = v_2$ і разом з мінімальною пройденою відстанню $r = 1$ заноситься до заголовку чергового стовпчика.

2) При заповненні другого стовпчика таблиці числові мітки модифікуються згідно з правилом: до довжини ребра (v_2, j) додається пройдена відстань 1 і результат порівнюється з попередньою числовою міткою в рядку; якщо нова сумарна відстань менше попередньої, то вона приймається як поточна, при цьому вершина змінюється на вершину з поточного стовпчика. Так, ребро (v_2, v_3) має довжину 1, тоді пройдена відстань $r = 1 + 1 = 2$ має кінцеве значення, отже, воно приймається як поточне замість тимчасової мітки ∞ . При цьому вершина v_1 змінюється на вершину v_2 з поточного стовпчика.

Ребро (v_2, v_8) має довжину 1, тоді пройдена відстань $r = 1 + 1 = 2 > 1$, тому в рядку v_8 зберігається попередня числова мітка, позначення вершини при цьому також не змінюється. Слід зауважити, що алгоритм Дейкстри в цьому випадку зазнає таку ж модифікацію, що і в задачі 3.4, а саме: якщо кінцева числова мітка вже визначена на поточному етапі, далі вона вже не

змінюється (оскільки може тільки збільшуватися), тобто мінімальна відстань (довжина ланцюга) вже виявляється визначено цим значенням числової мітки. Таким чином, виконувати додавання і порівняння для вершин з кінцевими числовими позначками на наступних етапах не має сенсу, їх слід розглядати по черзі (в порядку зростання індексів) і вибирати на кожній наступній ітерації як постійно помічені, поки всі вони не будуть вичерпані.

З усіх кінцевих числових міток другого стовпчика (для вершин v_3 , v_8) вибирається мінімальна – 1. Їй відповідає вершина v_8 . Тепер вершина $u = v_8$ є постійно поміченою, вона вибирається як новий відправний пункт і запам'ятовується у заголовку третього стовпця одночасно з пройденою відстанню $r = 1$.

3) Третій стовпець заповнюється аналогічно попередньому. Тут модифікується мітка в рядку v_{15} . Всі отримані тут кінцеві числові мітки виявляються однаковими (дорівнюють 2), тому вибирається перша за порядком, а саме v_3 . Тепер $u = v_3$ є постійно поміченою, розглядається як новий відправний пункт і записується до заголовку четвертого стовпчика разом з пройденою відстанню $r = 2$.

4) У четвертому стовпці не змінюються постійні мітки з рядків v_9 і v_{15} , рівні 2. При цьому модифікуються мітки в рядках v_4 і v_{10} . Далі вибирається перша за порядком вершина $u = v_9$ з мінімальною міткою у стовпці, яка тепер постійно позначена, приймається як новий відправний пункт і зберігається в заголовку у наступній колонці поряд з пройденою відстанню $r = 2$.

5) В п'ятому стовпці числова мітка 2 в рядку v_{15} не змінюється і є мінімальною. При цьому з'являється кінцева мітка 3 в рядку v_{16} . Таким чином, як постійно помічена вершина тепер вибирається v_{15} і заноситься до заголовку шостого стовпчика разом з пройденою відстанню $r = 2$.

6) У шостому стовпці нові кінцеві числові позначки не з'являються, при цьому як постійно помічена вершина тепер вибирається v_4 і заноситься до заголовку сьомого стовпчика разом з пройденою відстанню $r = 3$.

7) У сьомому стовпці з'являються дві нові кінцеві числові мітки 4 в рядках v_5 і v_{11} , інші мітки не змінюються. Як постійно помічену слід вибрати вершину v_{10} , яка разом з пройденою відстанню $r = 3$ заноситься до заголовку восьмого стовпця.

8) У восьмому стовпці з'являється одна нова кінцева числова мітка 4 в рядку v_{17} , інші мітки не змінюються. Тепер як постійно помічену слід вибрати вершину v_{16} , яка разом з пройденою відстанню $r = 3$ заноситься до заголовку дев'ятого стовпчика.

9) У дев'ятому стовпці нова кінцева числова мітка не з'являється, а інші мітки не змінюються, тому як постійно помічена вибирається вершина v_5 , яка разом з пройденою відстанню $r = 4$ заноситься до заголовку десятого стовпчика.

10) У десятому стовпці додаються ще дві нові кінцеві числові мітки 5 в рядках v_6 і v_{12} , інші позначки не модифікуються, як постійно помічена вибирається вершина v_{11} , яка разом з мінімальною числовою міткою – пройденою відстанню $r = 4$ заноситься до заголовку одинадцятого стовпця.

11) У одинадцятому стовпці з'являється одна нова кінцева числова мітка 5 в рядку v_{18} , інші позначки не модифікуються. Як постійно помічена вибирається вершина v_{17} , яка разом з мінімальною пройденою відстанню $r=4$ заноситься до заголовка дванадцятого стовпчика.

12) У дванадцятому стовпці нові кінцеві числові мітки не з'являються. Як постійно помічена вибирається вершина v_6 , яка разом з пройденою відстанню $r = 5$ заноситься до заголовку тринадцятого стовпчика.

13) У тринадцятому стовпці з'являються дві числові мітки 6 в рядках v_7 і v_{13} , інші мітки не змінюються. Як постійно помічена вибирається вершина v_{12} , яка разом з пройденою відстанню $r=5$ заноситься до заголовку чотирнадцятого стовпчика.

14) У чотирнадцятому стовпці з'являється ще одна числова мітка 6 в рядку v_{19} , інші мітки не змінюються. Як постійно помічена вибирається

вершина v_{18} , яка разом з пройденою відстанню $r = 5$ заноситься до заголовку п'ятнадцятого стовпчика.

15) У п'ятнадцятому стовпці мітки не змінюються і нові не з'являються, тому як постійно помічена вибирається перша за списком вершина v_7 з кінцевою числовою міткою $r = 6$, яка заноситься до заголовку шістнадцятого стовпчика.

16) У шістнадцятому стовпці з'являється нова кінцева числова мітка 7, якій відповідає вершина v_{14} . При цьому як постійно помічена вибирається вершина v_{13} і разом з пройденою відстанню $r = 6$ заноситься до заголовку сімнадцятого стовпчика.

17) У сімнадцятому стовпці модифікується мітка з рядка v_{20} , відповідна відстань дорівнює 7. Як постійно помічена вибирається вершина v_{19} і разом з мінімальною відстанню $r = 6$ запам'ятовується у заголовку вісімнадцятого стовпчика.

18) У вісімнадцятому стовпці нові кінцеві числові позначки не з'являються, тому як постійно помічена вибирається вершина v_{14} і разом з пройденою відстанню $r=7$ заноситься до заголовку дев'ятнадцятого стовпчика.

19) У дев'ятнадцятому стовпці модифікується мітка, яка відповідає рядку v_{21} . Тепер рядки мають кінцеві числові мітки, які більш змінюватися не можуть. Отже, процес пошуку мінімальних відстаней завершено.

Таким чином, з табл. 3.20 можна отримати інформацію про всі найкоротші ланцюги і відповідні їм довжини.

Наприклад, потрібно знайти найкоротший ланцюг з вершини v_1 до вершини v_{21} . Послідовність вершин у ланцюзі випикується з кінця: останній заповнений осередок в рядку v_{21} містить інформацію про довжину найкоротшої ланцюга $r_{\min} = 8$ і передостанній вершині в цьому ланцюзі – v_{14} . Інформація про попередню вершину знаходиться в останній комірці рядка v_{14} , в даному випадку – вершина v_7 , далі – v_6, v_5, v_4, v_3, v_2 і v_1 , яка є початком маршруту:

$$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6 \xrightarrow{1} v7 \xrightarrow{1} v14 \xrightarrow{1} v21, r_{\min} = 8.$$

Дані всіх найкоротших ланцюгів представлені в табл. 3.21.

Таблиця 3.21 – Найкоротші ланцюга з вершини $v1$ графа $G5$

ланцюг	довжина
$v1 \xrightarrow{1} v2, r_{\min} = 1$	$r(v1, v2) = 1$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3, r_{\min} = 2$	$r(v1, v3) = 2$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4, r_{\min} = 3$	$r(v1, v4) = 3$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5, r_{\min} = 4$	$r(v1, v5) = 4$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6, r_{\min} = 5$	$r(v1, v6) = 5$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6 \xrightarrow{1} v7, r_{\min} = 6$	$r(v1, v7) = 6$
$v1 \xrightarrow{1} v8, r_{\min} = 1$	$r(v1, v8) = 1$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v9, r_{\min} = 2$	$r(v1, v9) = 2$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v10, r_{\min} = 3$	$r(v1, v10) = 3$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v11, r_{\min} = 4$	$r(v1, v11) = 4$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v12, r_{\min} = 5$	$r(v1, v12) = 5$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6 \xrightarrow{1} v13, r_{\min} = 6$	$r(v1, v13) = 6$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6 \xrightarrow{1} v7 \xrightarrow{1} v14, r_{\min} = 7$	$r(v1, v14) = 7$
$v1 \xrightarrow{1} v8 \xrightarrow{1} v15, r_{\min} = 2$	$r(v1, v15) = 2$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v9 \xrightarrow{1} v16, r_{\min} = 3$	$r(v1, v16) = 3$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v10 \xrightarrow{1} v17, r_{\min} = 4$	$r(v1, v17) = 4$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v11 \xrightarrow{1} v18, r_{\min} = 5$	$r(v1, v18) = 5$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v12 \xrightarrow{1} v19, r_{\min} = 6$	$r(v1, v19) = 6$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6 \xrightarrow{1} v13 \xrightarrow{1} v20, r_{\min} = 7$	$r(v1, v20) = 7$
$v1 \xrightarrow{1} v2 \xrightarrow{1} v3 \xrightarrow{1} v4 \xrightarrow{1} v5 \xrightarrow{1} v6 \xrightarrow{1} v7 \xrightarrow{1} v14 \xrightarrow{1} v21, r_{\min} = 8$	$r(v1, v21) = 8$

Граф, який ілюструє дерево найкоротших ланцюгів з вершини v_1 , зображено на рис. 3.21.

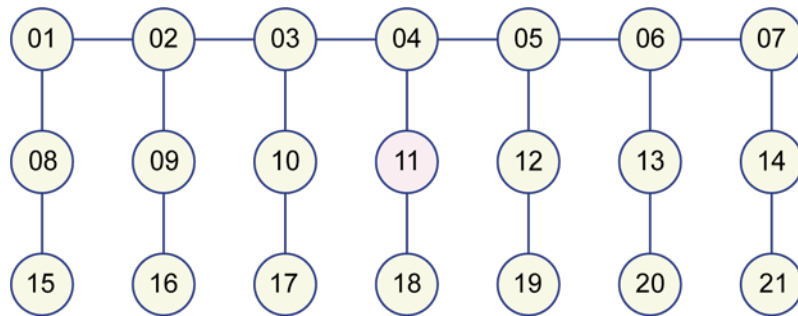


Рис. 3.21 – Дерево найкоротших ланцюгів з вершини v_1 графа G_5

3.7 Висновки до розділу 3

Розглянуто метод побудови оптимального маршруту на дорожній інфраструктурі між двома координатами, який має дорожні колізії, для його подальшої реалізації як online хмарного сервісу для водіїв транспортних засобів.

Виконано покроковий синтез алгоритму побудови оптимального маршруту руху транспортного засобу шляхом використання модифікованого алгоритму Дейкстри на топологічній інфраструктурі дорожнього руху, що приводиться до графу, з урахуванням існування дорожніх колізій.

Вирішені задачі:

1) Визначення всіх найкоротших шляхів і відповідних відстаней з вершини a до всіх інших вершин інфраструктури.

2) Пошуку всіх найкоротших відстаней між усіма парами вершин для графа з одиничними ваговими коефіцієнтами, заданого матрицею суміжності.

3) Оцінювання різних типів графів за розробленими критеріями якості, заданих матрицями суміжності, шляхом обчислення найкоротших відстаней між усіма парами вершин і складності графа.

4) Порівняння різних графових архітектур з метою визначення ефективних рішень при проектуванні нових дорожніх інфраструктур для скорочення часу руху за замовленим маршрутом.

Основні результати розділу опубліковані в роботах [147, 149, 150].

4 КІБЕРФІЗИЧНА СТРУКТУРА ХМАРНОГО ТРАНСПОРТНОГО КОМП'ЮТИНГУ

Пропонується структура кіберфізичної системи, орієнтована на якісні інновації в моніторингу і управлінні транспортом, що усувають забруднення планети, численні аварії та колізії на дорогах за рахунок поступового переміщення дорожніх знаків в кіберпростір, делегування координат кожного автомобіля і хмарного online надання інформації про маршрут руху. Хмарний сервіс призначений для моніторингу та управління дорожнім рухом в реальному масштабі часу на основі використання глобальних систем позиціонування, навігації (GPS, GPRS), інтелектуальних дорожніх контролерів, засобів радіочастотної ідентифікації автомобілів з метою підвищення якості та безпеки пересування транспортних засобів, а також мінімізації часових і матеріальних витрат при русі автомобілів за заданими маршрутами. Розроблені моделі та методи інтегруються в кіберфізичну систему online взаємодії хмари моніторингу та управління з транспортними засобами та інфраструктурою дорожнього руху. Кожен автомобіль має особистий кабінет у кіберпросторі, інваріантний по відношенню до водіїв, які обслуговують транспортний засіб.

Задачі дослідження:

- 1) Визначення інноваційної привабливості для розробки і впровадження кіберфізичної системи хмарного комп'ютиного управління транспортом на основі цифрового моніторингу.
- 2) Створення хмарної кіберфізичної інфраструктури, що відображає і моделює процеси пересування транспортних засобів в часі і просторі, для сервісного обслуговування учасників дорожнього руху в реальному масштабі часу.
- 3) Створення архітектурних рішень для компонентів кіберфізичного комп'ютиного, пов'язаного з цифровим моніторингом і хмарним керуванням транспортом.

4) Верифікація моделей і методів, інтегрованих в структуру прототипа кіберфізичної системи хмарного управління транспортом.

4.1 Інноваційна привабливість "Cloud Traffic Control"

Запропонована кіберфізична система – дорожня інфраструктура і фізичний транспорт з цифровими сенсорами, а також хмарні сервіси моніторингу та управління дорожнім рухом відрізняється від існуючих структурною інтеграцією online взаємодіючих інтерактивних компонентів: 1) Суперпозиція сервісів електронної картографії, радіолокації, радіонавігації і online вибору маршруту руху. 2) Хмарний сервіс сенсорного моніторингу автомобіля і управління дорожнім рухом на основі поступового перетворення реальних знаків дорожнього руху у віртуальні. 3) Створення повного рефлекторного відображення фізичних транспортних процесів і явищ в хмарних сервісах, що включають кабінети автомобілів з їх дорожньою просторово-часовою історією (рис. 4.1).

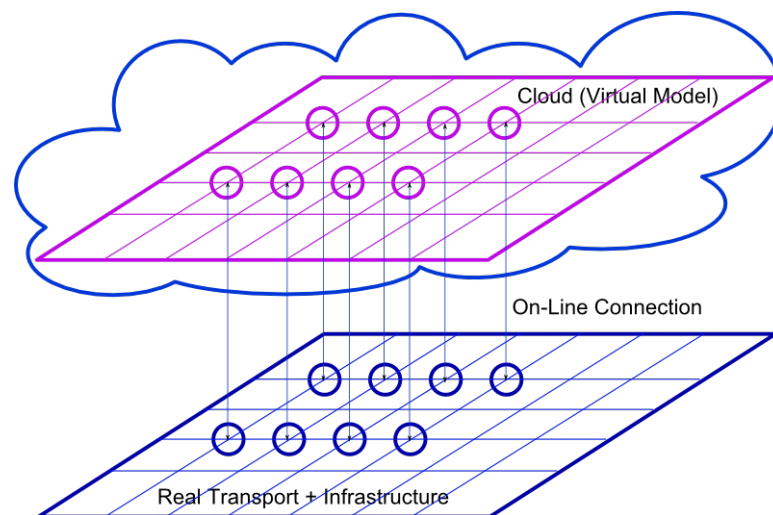


Рис. 4.1 – Відображення інфраструктури і транспорту в хмарі

Структурна модель взаємодії хмар моніторингу та управління з транспортними засобами представлена на рис. 4.2, де автомобілі, перебуваючи в режимі on-line, делегують свої ідентифікатори (персональні дані), параметри руху і поточні координати хмарі, а натомість отримують в реальному масштабі часу сервіси оптимального (за часом, витратами і якістю) маршруту і режиму

руху для досягнення кінцевого пункту. Інтегральний аналіз дорожньої обстановки на основі оперативного збору даних від автомобілів і інфраструктурних моніторів дає можливість в режимі on-line оптимально управляти дорожніми контролерами перемикання світлофорів.

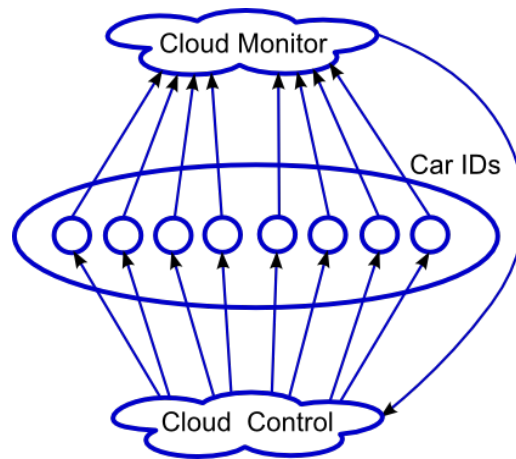


Рис. 4.2 – Структура взаємодії транспорту та хмари

Взаємодія реального світу (автомобіль і інфраструктура) з хмарою формує два типи відношень, що задаються автоматними моделями, рис. 4.3: 1) транспортної інфраструктури з хмарою моніторингу та управління; 2) автомобіля з хмарою для оптимізації та ефективності пересування.

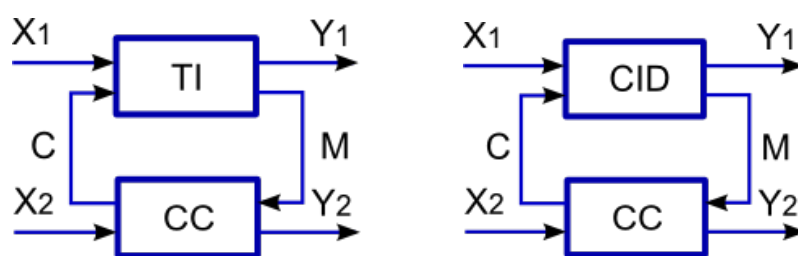


Рис. 4.3 – Взаємодія інфраструктури і транспорту з хмарою моніторингу та управління

Тут представлені сигнали X_1, Y_1, X_2, Y_2, C, M , які визначаються як: 1) вхідні умови або операнди, необхідні для виконання замовлених сервісів; 2) вихідні сповіщальні сигнали, що підтверджують виконання сервісних

операцій; 3) вхідні управляючі дії, що формують запити на виконання сервісів; 4) вихідні змінні, що формують і ідентифікують стан системи управління; 5) сигнали розумного управління автомобілем або інфраструктурою дорожнього руху; 6) сповіщальні сигнали про виконання операційних сервісів. Автоматні моделі системи управління дорожнім рухом та автомобілем у формі взаємодії змінних, що представлені функціями переходів і виходів автомата першого роду:

$$\begin{aligned}
 CC &= \{X, Y, C, M, f, g\}, \\
 \begin{cases} Y(t) = f[(X(t), M(t), Y(t-1))]; \\ C(t) = g[(X(t), M(t), Y(t-1))]. \end{cases}
 \end{aligned}$$

Кожен з двох автоматів взаємодії інфраструктури і транспорту з хмарою має дві вхідні змінні (замовлення сервісів та стану об'єктів управління), а також два вихідних сигнали для моніторингу стану автомата (хмари) і управління хмарними сервісами.

В даному випадку автоматне моделювання процесів управління транспортом реалізує тезу «потоки автомобілів керують світлофорами, але не навпаки». Ідея була описана в інтернет-публікації [<http://www.wired.com/wiredscience/2010/09/traffic-lights-adapt/>] і реалізована в центральній частині Дрездена. Це забезпечило на практиці істотне зниження часу простою на перехрестях (56% для автобусів і трамваїв, 9% для автомобілів, вантажівок і 36% для пішоходів). В даному випадку інтелектуальні світлофори мають функції моніторингу вхідних і вихідних потоків автомобілів на перехрестях, що дає можливість реалізувати квазіоптимальне управління вуличними контролерами. Інший приклад. Населення США витрачає 500 000 років в пересуваннях по дорогах за один рік, що становить 100 мільярдів доларів сумарних витрат. Якщо зменшити час, проведений в автомобілях за рік на 10%, то економія складе кругленьку суму в 10 мільярдів доларів.

Більш детальна картина взаємодії реальних і віртуальних компонентів з хмарою моніторингу та управління транспортом представлена на рис. 4.4 (буферні комп'ютери дорожньої інфраструктури, інтелектуальні вуличні світлофори, Інтернет, розумна пил, Car-ID, супутники навігації, автомобіль, електронна карта, засоби захисту хмарних сервісів, спеціальні послуги державним службам, інтерфейси або гаджети для зв'язку з інтернетом).

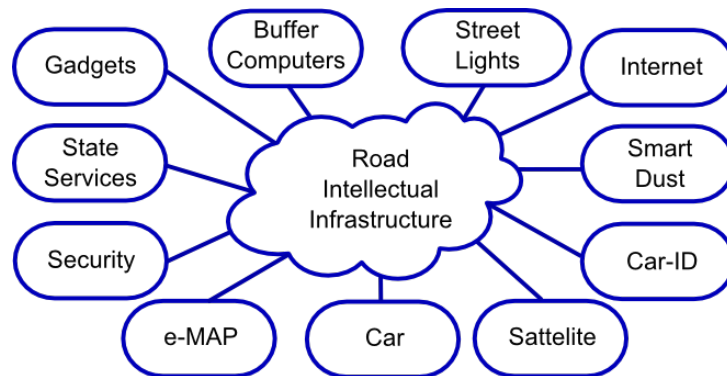


Рис. 4.4 – Інтелектуальна інфраструктура дорожнього руху

Переваги від впровадження і ринкова привабливість хмарних сервісів дорожнього руху, аж ніяк не очевидні для громадян:

1) Для планети – збереження екології за рахунок зменшення забруднення навколишнього середовища, підвищення тривалості і якості життя людини, економія паливно-енергетичних ресурсів шляхом скорочення часу руху завдяки вибору оптимального маршруту, зменшення кількості і складності пробок за рахунок впровадження в інфраструктуру інтелектуальних світлофорів.

2) Для державних структур – міліція, дорожня інспекція – точна ідентифікація автомобілів, моніторинг позиціонування транспортних засобів в часі і просторі, включаючи викрадення, колізії, несанкціоновані маршрути. Істотне зниження аварійності за рахунок безпеки маневрів, зменшення наслідків дорожньо-транспортних пригод, підвищення безпеки і комфорту учасників дорожнього руху; моніторинг реальної швидкості руху всіх транспортних засобів та інформування про зони швидкісного режиму.

Цифровий моніторинг проїзду на заборонні знаки і сигнали світлофорів. Протидія угону транспортного засобу і самовільного покидання місця ДТП, завдяки моніторингу місця розташування кожного автомобіля. Інформування за допомогою тривожної кнопки спеціальних служб про події, що сталися на дорогах або з автомобілем. Автоматична реєстрація динаміки ДТП внутрішніми засобами ідентифікаційного модуля. Реєстрація стилю водіння для страхових компаній.

3) Для транспортних компаній – моніторинг позиціонування і пересування транспортних засобів, квазіоптимальне виконання замовлень з перевезення пасажирів і вантажів з точки зору мінімізації матеріальних і / або часових витрат;

4) Для водія – надання сервісів, пов'язаних з прокладанням квазіоптимальних маршрутів і графіка руху з урахуванням негативних чинників існуючої інфраструктури з метою мінімізації матеріальних і часових витрат в режимі реального часу; зниження аварійності завдяки моніторингу закритих для візуального перегляду ділянок дороги і прорахунку рівня безпеки маневрів. Попередження водія про потенційну небезпеку на замовленому маршруті, отриманої від хмари в процесі руху.

5) Для пасажирів – надання сервісів з моніторингу позиціонування і руху пасажирських транспортних засобів на зупиночних або транспортних терміналах за допомогою використання стаціонарних моніторів або мобільних гаджетів для зв'язку з відповідними хмарними сервісами. Візуалізація на екрані в автомобілі критичних точок маршруту руху транспортного засобу в реальному масштабі часу шляхом використання камер відеоспостереження.

6) Для міста – економія палива, зменшення забруднення навколишнього середовища, скорочення часу руху, завдяки вибору оптимального маршруту, наданого хмарою. Зменшення кількості та складності пробок за рахунок планування руху транспортних засобів, що враховує майбутні маршрути інших учасників. Коригування маршруту руху транспортного засобу в реальному часі при зміні дорожньої ситуації. Інтелектуальне управління

циклами перемикання світлофорів залежно від дорожньої обстановки на перехрестях. Генерування аналітичних звітів і рекомендацій щодо поліпшення інфраструктури доріг, розстановки знаків, світлофорів та централізоване програмування циклів їх перемикання.

Аргументи проти впровадження хмари в масштабах країни: «Порушення права на недоторканність приватного життя, оскільки теоретично хмара здійснює тотальний моніторинг всіх транспортних засобів». Насправді сьогодні існує система законного перехоплення телекомунікацій, реалізована відповідно до міжнародних вимог. Але можливість перехоплення телефонних переговорів будь-якого абонента використовується тільки в ході слідства і з санкції суду. У тому числі є можливість відслідковувати місце знаходження абонента. Даний факт для законослухняних громадян ніякої проблеми не створює. «Додаткові витрати на придбання апаратно-програмних засобів ідентифікації та зв'язку транспорту з хмарою». Вартість цих коштів невисока і порівнянна із середньою сумою штрафу за порушення ПДР. Економічні переваги хмари, пов'язані з економією палива і зменшення часу, поїздки компенсують витрати протягом року.

В рамках системної інтеграції захищених від несанкціонованого доступу, хмарних компонентів (рис. 4.5): хмар моніторингу, управління і збереження історії, блоків радіочастотної ідентифікації транспорту, а також інтелектуальних сенсорів дорожньої інфраструктури, створюються мікросервіси квазіоптимального управління транспортом і дорожнім рухом в режимі реального часу для вирішення соціальних, гуманітарних, економічних та екологічних проблем:

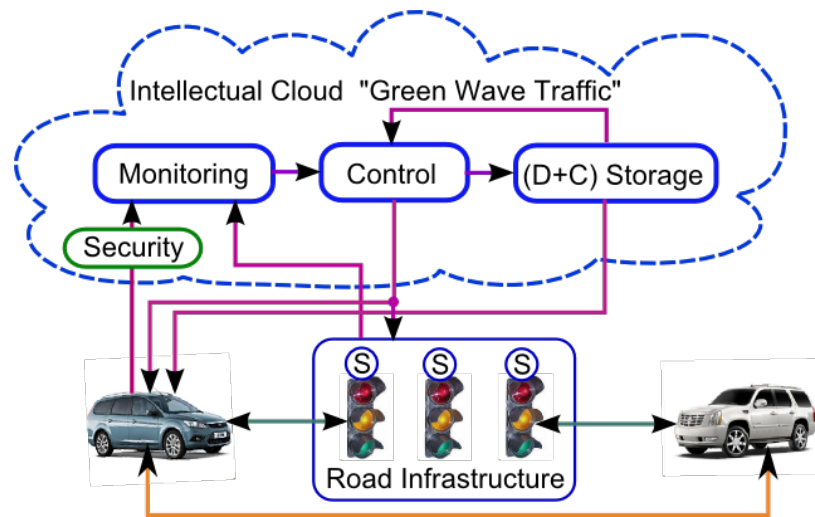


Рис. 4.5 – Інтелектуальне хмара «Зелена хвиля»

1. Спеціального управління перемиканням дорожніх світлофорів в режимі on-line для автоматичного забезпечення безперешкодного руху по заданому маршруту спеціалізованих машин або кортежів (перевезення дітей, важливі державні чиновники, швидка допомога, пожежна служба, військові колони, небезпечні вантажі).

2. Оптимального управління світлофорами в режимі on-line на дорогах і перехрестях за допомогою точного цифрового моніторингу дорожнього руху шляхом використання RFID міток автомобіля, що дає можливість мінімізувати час проходження маршруту усіма учасниками дорожнього руху.

3. Планування оптимального маршруту для досягнення одного або кількох пунктів призначення автомобілем в часі і в просторі, що дає можливість зменшити часові та матеріальні витрати при заданій якості комфорту (час доби, року, дорожнє покриття, ліві повороти, погодні умови, затори, ремонтні роботи) пересування транспортного засобу.

4. Інтелектуальної історії пересування автомобіля, що має віртуальну модель в кіберпросторі – індивідуальну комірку в хмарі, інваріантну по відношенню до водіїв, які обслуговують транспортний засіб, що дає можливість відстежити будь-які пересування транспортного засобу в

минулому; а також прогнозувати бажані маршрути і поїздки в майбутньому без участі водія.

5. Сервіс хмарного моніторингу міток радіочастотної ідентифікації транспортних засобів в режимі on-line, який усуває автомобільні номери з системи обліку, що дає можливість: 1) Виключити безпосередню участь дорожньої поліції в фіксації порушень правил дорожнього руху (перевищення швидкості, проїзд на заборонні сигнали світлофорів, порушення правил маневрування); 2) Заощадити тисячі тонн металу на виготовлення номерів і спростити реєстрацію автомобілів при покупці з декількох днів до декількох хвилин; 3) Автоматизувати оформлення ДТП без участі співробітників дорожньої поліції шляхом цифрового моніторингу цифрової карти події, скопійованого з хмари; 4) Істотно (x2-x5) скоротити чисельний склад дорожньої поліції, оскільки історія переміщень автомобіля і його дорожніх порушень стає абсолютно прозорою для хмари, що дозволить автоматично списувати з рахунків водія вартість порушення відповідно до законодавства цієї країни; 5) Повністю виключити корупцію у відносинах водія з дорожньою поліцією, завдяки неможливості стерти інформацію про порушення в хмарі; 6) Практично ліквідувати криміналітет в області викрадення автомобілів, завдяки вбудованому в машину блоку радіочастотної ідентифікації, що забезпечує цілодобову спостережуваність транспорту в режимі on-line, якщо автомобіль фізично не знищений; 7) Спростити легалізацію водія шляхом ідентифікації водія ліцензії (driver's license) в списку дозволених осіб блоку радіочастотної ідентифікації автомобіля по протоколу "blue tooth", що дозволяє усунути виготовлення паперів та довіреностей на водіння автомобіля іншими особами; 8) У кілька разів зменшити кількість ДТП та істотно підвищити якість життя водіїв і пасажирів завдяки тотальному моніторингу порушень і невідворотності покарань за них; 9) Зменшити на 30% автомобільні викиди вуглекислого газу за рахунок зменшення часу простоїв на перехрестях і вибору оптимальних режимів і маршрутів пересування транспорту; 10) Забезпечити високу ринкову привабливість хмари за рахунок продажу сервісів

компаніям і приватним особам, що гарантує отримання високого прибутку – від сотень мільйонів до десятків мільярдів доларів, що масштабується залежно від площі покриття сервісами: міста, області, країни, всього світу.

Створення цифрової інфраструктури моніторингу та управління дорожнім рухом за допомогою детермінованих керуючих і спостерігаючих вузько випромінювань є майбутнє транспорту без водіїв, аварій, колізій на землі, воді і в повітрі. Для цього необхідні більш точні (до 10 сантиметрів) тривимірні стандарти ідентифікації простору планети з можливістю його моніторингу та управління.

4.2 Структура сервісів хмарного управління транспортом

Деталізується масштабована кіберфізична система цифрового моніторингу і хмарного управління дорожнім рухом в реальному масштабі часу на основі використання глобальних систем позиціонування і навігації, автомобільних комп'ютерів, навігаторів або мобільних гаджетів, розумних світлофорів, технологій Big Data і Internet of Things з метою підвищення якості і безпеки експлуатації транспортних засобів, а також мінімізації часових і матеріальних витрат при русі автомобілів за заданими маршрутами. Основна інноваційна ідея – поступове перенесення світлофорів з поверхні землі в віртуальний кібер простір для хмарного управління транспортом, обладнаним мобільним гаджетом (навігатором) або автомобільним комп'ютером, коли на екран (лобове скло) виводяться: карта з дорожніми знаками, маршрут прямування, координати учасника дорожнього руху та реальні сигнали віртуального розумного світлофора. Пропонується контейнер інноваційних технологій для комплексного вирішення соціальних, гуманітарних, економічних, паливно-енергетичних, страхових, кримінальних і екологічних проблем на основі створення і застосування хмарного сервісу цифрового моніторингу та управління транспортом. Згадані технології та функціональні компоненти інтегруються в просту для реалізації кібер комп'ютерингову модель інтерактивної взаємодії в реальному часі розумної інфраструктури цифрового

моніторингу і хмарного управління з гаджетом водія або комп'ютером автомобіля.

Структура масштабованої розумної кіберфізичної комп'ютерингової системи Cloud Traffic Control (рис. 4.6) включає наступні компоненти:

1) Оцифровані відношення: правила дорожнього руху, дорожнє законодавство, культура учасників дорожнього руху, традиції.

2) Транспортні засоби з цифровими ідентифікаторами, водії і пішоходи з розумними гаджетами.

3) Кіберфізична інфраструктура дорожнього руху, що включає дорожнє полотно і розв'язки, знаки, світлофори, парковки, заправки, пункти харчування та ремонту, електронні карти, засоби позиціонування і навігації, засоби телекомунікації, Edge Gadgets, цифрові ідентифікатори, Embedded MicroSystems and Smart Things.

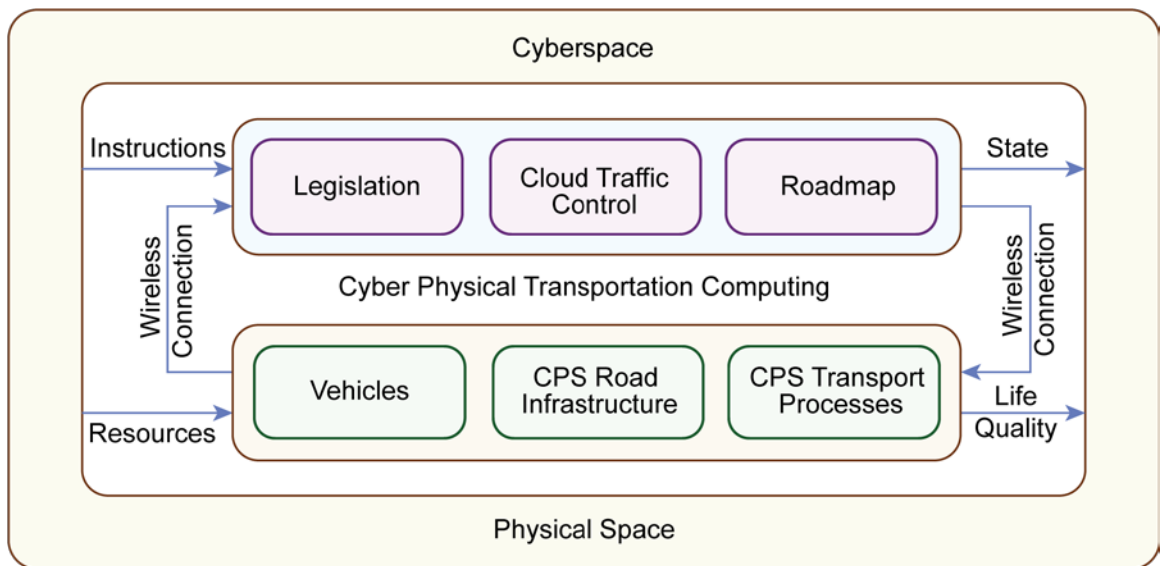


Рис. 4.6 – Cyber Physical Transportation Computing

4) Хмарне планування і управління дорожнім рухом на основі цілодобового точного цифрового online моніторингу позиціонування і маршрутів руху транспортних засобів.

5) RoadMap сталого розвитку транспортної системи планети для підвищення якості життя людини і збереження екосистеми планети на основі: точного до 10 см позиціонування, створення достатньої потужності online 24/7 засобів телекомунікацій, переходу на використання електромобілів, очищення вулиць і доріг від знаків і світлофорів, driver-free car control, що виключає дорожні аварії, викрадення автомобілів, транспортні пробки і дорожню поліцію.

6) Комфортні, безаварійні, оцифровані транспортні процеси, що підлягають глобальній та локальній мінімізації за інтегральною метрикою: витрачання енергоресурсів, часу доставки вантажів і пасажирів, фінансових і матеріальних витрат на проектування і експлуатацію кіберфізичної інфраструктури.

Хмарні кіберфізичні комп'ютерні сервіси (рис. 4.7):

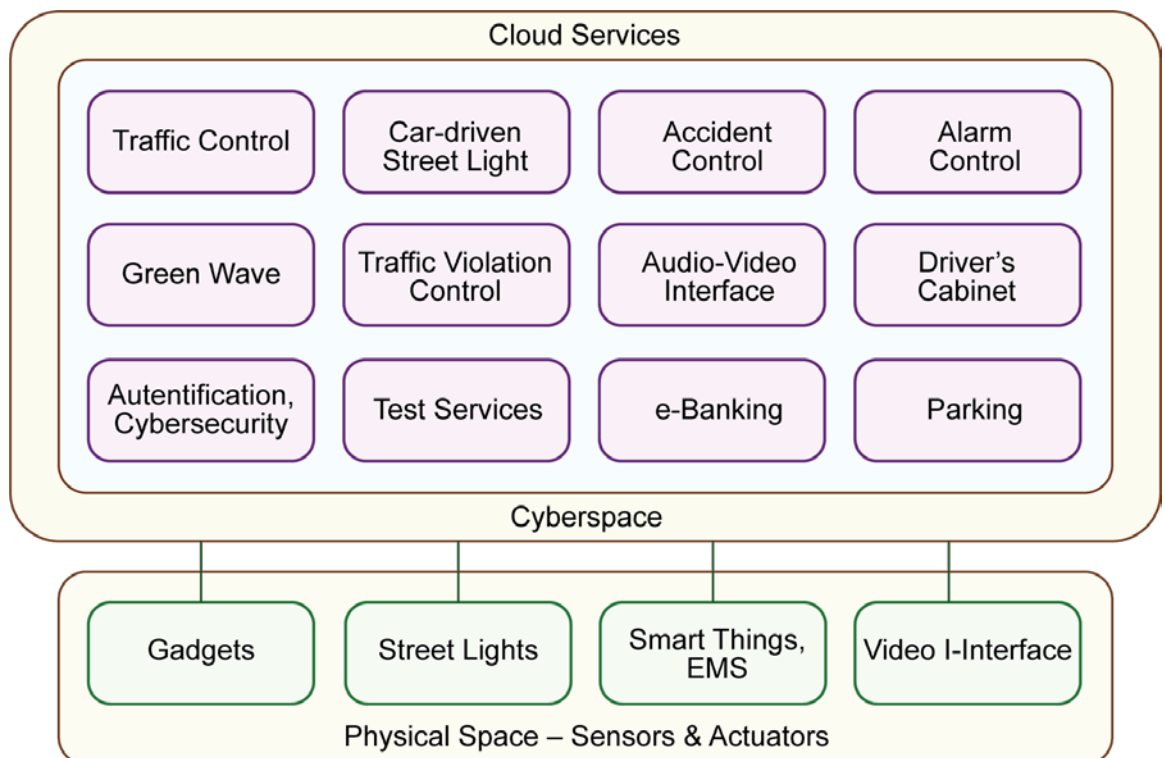


Рис. 4.7 – Cloud Traffic Control Services

1) Хмарне online управління дорожнім рухом на основі цифрового моніторингу транспортних засобів і їх маршрутів, використання віртуальних знаків і світлофорів. Персональне планування і online квазіоптимальне управління маршрутом руху на основі делегування хмарі координат автомобіля і замовленого шляху.

2) Car-driven хмарне online управління віртуальним і/або реальним світлофором на перехресті шляхом цифрового моніторингу позиціонування транспортних засобів.

3) Точний цифровий моніторинг і організаційне управління дорожньо-транспортною пригодою на основі просторово-часового та причинно-наслідкового аналізу колізії, яка трапилася.

4) Цифровий моніторинг несанкціонованого використання транспортного засобу в режимі 24/7, візуалізація позиціонування і екстрене управління блокуванням автомобіля.

5) Забезпечення безперешкодного руху за заданим маршрутом транспорту, спеціалізованих машин або кортежів (перевезення дітей, державні чиновники, швидка допомога, пожежна служба, військові колони, небезпечні вантажі).

6) Online сервіс попередження: водія про потенційну небезпеку на замовленому маршруті, отриманої від хмари в процесі руху; спеціальних служб про події і порушення, що трапилися на дорогах або з автомобілем. Сервіс аналітичних звітів і рекомендацій щодо поліпшення інфраструктури доріг, аналітика популярних запитуваних маршрутів, планування збільшення пропускної здатності завантажених ділянок. Online сервіс моніторингу стану дорожнього покриття в інтеграції з системами прогнозування і відображення погодних умов.

7) Сервіс ергономічного online надання відео-аудіо інформації водієві на склі-моніторі зі звуковим дублюванням дорожньої обстановки.

8) Сервіс особистого віртуального кабінету водія-автомобіля, що зберігає переваги в маршрутах, режимах руху, колізіях і інші особливості подорожей.

9) Сервіс захисту персональних даних і аутентифікації водія-автомобіля на основі персональних гаджетів, RFID-міток транспортного засобу і первинних біометричних показників.

10) Online сервіс тестування обладнання автомобіля за запитом водія з особистого віртуального кабінету при виникненні несподіваних поломок. Дана послуга була запропонована компанією Synopsys, Virtual Hardware ECU для істотного підвищення якості хмарного тестового сервісу automotive electronic an software systems.

11) Online банкінг товарів і послуг, пов'язаних з обслуговуванням автомобіля на заправках, станціях технічного обслуговування і користування платними дорогами і парковками.

12) Цифровий моніторинг і хмарне управління парковками на основі Embedded MicroSystems and Smart Things.

4.3 Розумний хмарний світлофор

Ключовий інноваційний компонент інфраструктури TCS – віртуальний і/або реальний смарт-світлофор (Smart-Streetlight – SS) функціонально являє собою стаціонарний мікроконтролер (цифрову систему на кристалі) з приймачем, безпосередньо пов'язаний з хмарою управління трафіком. Він організовує бездротову мережу для взаємодії з автомобільними комп'ютерами (Car-Computer – CC) на дистанції до 100 метрів за протоколом Wi-Fi. Стан світлофора є доступним для моніторингу в хмарному сервісі Інтернету і виводиться на екрани всіх машин, які на маршруті транспортування перетинають перехрестя і тих, які знаходяться в зоні його компетенції. Сигналами світлофора можуть управляти дистанційно спеціальні служби, або запити від автомобілів, які перетинають перехрестя. Розумний світлофор сканує і обробляє всі заявки від транспортних засобів, надходять через TCS-

хмара в реальному масштабі часу. Він генерує відповідні сигнали управління, функціонально залежні від дорожньої обстановки на сторонах перехрестя. Всі світлофори міста об'єднуються в мережевий технологічний шар хмарного сервісу, який обслуговує інфраструктуру вулиць і доріг в інтелектуальному і/або автономному режимах з можливістю ручного і/або дистанційного керування. Метод управління світлофором використовує взяття булевої похідної за лініями руху. Останні мають лінійні графіки накопичення $N^* = k*t + b$ і пропуску автомобілів $N^o = -k*t + b$ через перехрестя – вузьке горлечко дорожньої інфраструктури – з періодом сканування автомобілів, рівним 1 секунді. Тут k – пропускна здатність або кількість проїжджаючих автомобілів через перехрестя в секунду, t – реальний час, b – вихідна кількість машин на перехресті. Обчислення або оновлення похідної за лініями руху перехрестя для формування сигналів світлофора здійснюється синхронно з вказаним періодом, але зміна його стану, за наявності черг на всіх сторонах перехрестя, здійснюється з затримкою, що мінімізує колізії (сумарний час простою автомобілів) на всіх лініях руху транспортних потоків. Цикл перемикання світлофора має мінімальний період, що дорівнює часу проїзду перехрестя автомобілем.

Світлофор, як спочатку примітивний засіб регулювання рухом на перехресті, поступово трансформується в спеціалізовану кіберфізичну систему моніторингу та управління, що включає компоненти: LED-монітор, відеокамера, Wi-Fi прийомо-передавач для зв'язку з автомобільними комп'ютерами та Інтернетом, програмні засоби інтелектуального управління транспортом. Останні вже сьогодні використовують високу математичну культуру: теорії автоматів, черг та знань, мережі Петрі і нейронні мережі, генетичні алгоритми і прискореного навчання (reinforcement learning). Система світлофорних агентів або сенсорів для моніторингу транспорту включає: досить дорогі відеодетектори і індуктивні петлі, вбудовані в дорожнє покриття, значно рідше використовуються звукові індикатори шуму, мікрохвильові радары. Позбутися від усіх агентів світлофора можна шляхом

цифрової ідентифікації автомобілів на основі точного позиціонування автомобільного гаджета або комп'ютера. Сьогодні таким агентом може бути мобільний телефон водія авто, який за допомогою глобальної системи позиціонування делегує своє місце розташування в хмару управління трафіком. У відповідь мобільний телефон отримує всі сервіси з управління дорожнім рухом: сигнали хмарного віртуального світлофора, синхронізовані з реальними світлофорами, оптимальні маршрути пересування, «зелену хвилю» при неінтенсивному трафіку, економію часу і палива. Взаємодія світлофора з системою управління рухом по перехрестю показана на рис. 4.8, де представлені: Streetlight – LED-монітор, Gadget – гаджет автомобіля або мобільний телефон, автомобіль, ідентифікований на хмарній мапі гаджетом водія, GPS Satellite – для точного позиціонування транспорту або гаджета, Camera – відеокамера спостереження рухом на перехресті, Computer – для реалізації розумного управління світлофором, TCS-хмара-сервіс управління трафіком в Інтернеті.

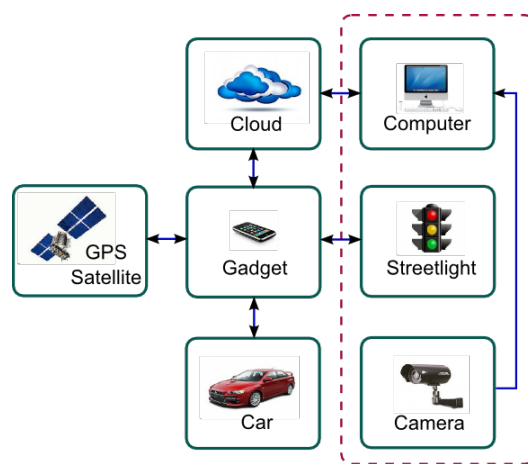


Рис. 4.8 – Взаємодія світлофора з інфраструктурою і TCS-хмарию

Червоним пунктиром обведені фізичні компоненти, які згодом перемістяться у віртуальний простір, що дозволить зробити інфраструктуру доріг набагато більше зеленої. Суттєвим доповненням до функціональності управління світлофором може служити технологічне рішення, запропоноване в [140-141]. Аналіз показує, що до 2020 року більшість ДТП відбуватиметься

на перехрестях. Найбільш частими причинами аварій тут є неадекватні спостереження (44,1%), помилкові припущення про маневрах інших автомобілів (8,4%), повороти з утрудненим оглядом (7,8%), незаконні маневри (6,8%), внутрішні відволікання уваги (5,7%), недооцінка інтервалу або швидкості інших транспортних засобів (5,5%). Підвищення безаварійності переїзду перехрестя автори пов'язують з розподіленою взаємодією (vehicular ad hoc networks – VANET) автомобілів між собою за протоколом V2V (IEEE 802.11p) за допомогою використання бездротових гаджетів, що передбачає наявність віртуального світлофора VTL для формування керуючих сигналів за принципом Car2X. Збільшення пропускної здатності перехрестя без додаткових витрат на розширення інфраструктури автори пов'язують з двома взаємодіями Car2Car і Car2 (X) VTL, які формують інтелектуальну децентралізовану мережу, на 60% підвищує пропускну здатність перехрестя. Подальше вдосконалення вуличного світлофора в інфраструктурі міст пов'язане з його перетворенням в кіберфізичну систему управління рухом на перехресті, структура якої представлена на рис 4.9.

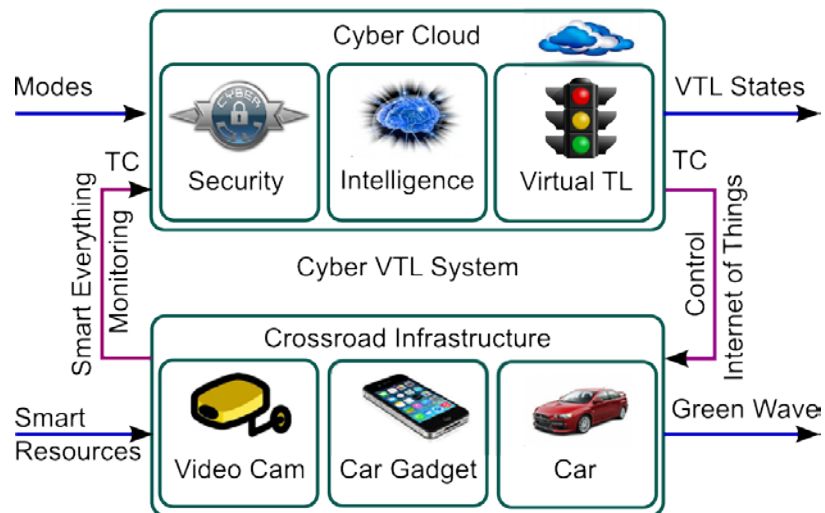


Рис. 4.9 – Кіберфізична система VTL-управління перехрестям

Кібер VTL-система призначена для організації «зеленої хвилі» при проїзді перехрестя автомобілем і містить два основних компоненти: віртуальний – Cyber Cloud і фізичний – Crossroad Infrastructure. Перший з них

призначений для формування керуючих впливів, що визначають сигнали світлофора, які залежать від передісторії, реального трафіку на перехресті. Другий компонент виконує моніторинг дорожньої обстановки за допомогою відеокамер, розумних датчиків і автомобільних гаджетів. Взаємний зв'язок хмарного світлофора з фізичною інфраструктурою перехрестя здійснюється за допомогою телекомунікаційних засобів (ТС): GPS Satellite, мобільного Інтернету, GSM / GPRS. Разом з тим, кіберсистема має можливість ручного управління світлофором шляхом використання входу Mode, а також спостереження за його станами в процесі функціонування за виходом VTL States. Два блоки Intelligence і Security забезпечують реалізацію розумних алгоритмів управління в залежності від реального трафіку, а також захист світлофора від несанкціонованого доступу. Інноваційна цінність кіберфізичної системи VTL-управління транспортом на перехресті полягає в використанні віртуального світлофора, його інтелектуальному управлінні залежно від дорожньої обстановки, а також у моніторингу останньої шляхом збору сигналів позиціонування і маршрутизації транспорту від гаджетів автомобілів.

Car-driven управління світлофором. Структура розумного світлофора характеризується оптимальним керуванням сигналами, які залежать від кількості транспортних засобів на смугах руху перехрестя. Кожна смуга руху складається з паралельних непересічних потоків транспортних засобів, спрямованих один проти одного. Ці потоки формують стан регістра (пам'яті), де кожен біт впливає на формування сигналу світлофора. Перевага віддається регістру, який має максимальну кількість одиничних значень. Беручи до уваги, що регістр може мати нульові і одиничні значення, в кожному циклі управління виконується регістрова операція зсуву вліво з ущільненням одиниць. Вона виключає арифметичні інструкції множення, додавання, ділення, вирахування з процесу управління інтелектуальним світлофором, що значно спрощує віртуальну (фізичну) реалізацію світлофора і збільшує швидкодію його роботи.

Для ущільнення даних за один такт використовується оригінальне

запатентоване рішення у вигляді регістра зсуву одиничних значень вліво з ущільненням [7]. Схема для обчислення керуючого сигналу, як гри двох транспортних потоків, представлена на рис. 4.10. Існують два потоки А і В, які за своєю суттю конфліктують (суперечать один одному). Вхід m визначає штучно створені переваги, які повинні мати місце в міській інфраструктурі. За замовчуванням ці переваги визначаються одиничними сигналами регістру m .

Процедура управління світлофором складається з трьох етапів:

- 1) Зрушення вліво з ущільненням

$$A = \text{SLC}(A), B = \text{SLC}(B).$$

- 2) Визначення смуги переможця

$$a = (A \& B) \oplus A, b = (A \& B) \oplus B$$

- 2) Формування світлофорних сигналів перехрестя

$$Q^a = \bigvee_{i=1}^n a_i, Q^b = \bigvee_{i=1}^n b_i, i = \overline{1, n}$$

де n – число бітів регістрів (кількість автомобілів для смуг, що не перетинаються).

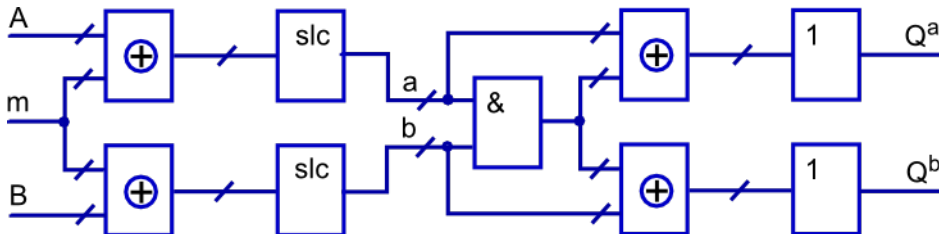


Рис. 4.10 – Cloud Sequencer світлофора на простому перехресті

Регістрові змінні a і b , що представляють вектори стислих вліво одиничних значень, об'єднані і інвертовані для одночасної реалізації хог-операції. Результати у вигляді станів регістрів є двома входами or-елемента, які утворюють стан двох булевих вихідних змінних для створення трьох комбінацій: 00, 01 і 10. Нульове значення (0) однієї з двох вихідних змінних є кращим рішенням, яке необхідно вибрати. Два нульових стани означають, що обидва рішення еквівалентні рівню переваги. В цьому випадку за замовчуванням вибирається потік А. Єдина комбінація двох булевих змінних неможлива.

Таким чином, точне управління світлофором відповідно до потоків трафіку на перехресті може і повинно виконуватися на основі тільки векторних логічних операцій `and`, `or`, `not`, `xor`, `slc`, без використання арифметичних функцій, що дає можливість проектувати віртуальні / фізичні `and-or` мультипроцесори, які значно скорочують часову затримку обчислень в циклі виконання хмарного управління перехрестям. У разі керування складним перетином доріг, де існує n конфлікуючих потоків трафіку, використовується модифікована схема управління, яка реалізується у вигляді векторно-логічного секвенсора, показаного на рис. 4.11. Тут вхідні регістри або пам'яті формують суперпозицію одиничних значень, що відповідають кількості автомобілів на непересічних напрямках руху. На одному з виходів секвенсора управління світлофором сигнал має нульове значення, яке ідентифікує потік трафіку з максимальною кількістю автомобілів. Установка сигналу трафіку в нульове значення являє собою компромісне рішення, яке залежить не тільки від кількості транспортних засобів на смузі руху, а й від числа автомобілів на інших потоках, які перетинають перехрестя.

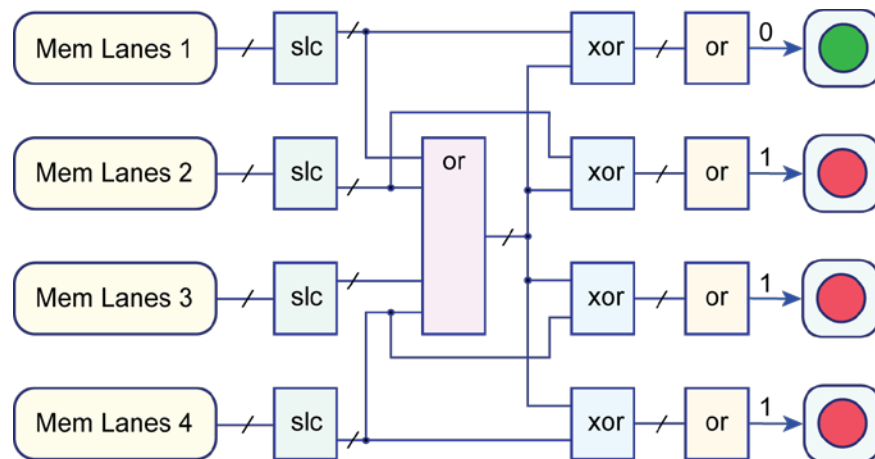


Рис. 4.11 – Cloud Sequenser світлофора на складному перехресті

Представлені схеми двох логічних секвенсоров інтегруються в структуру хмарного керування транспортними засобами на перехресті, рис. 4.12. При цьому світлофор поступово повинен трансформуватися в чисто хмарний сервіс, виведений на дисплей автомобіля.

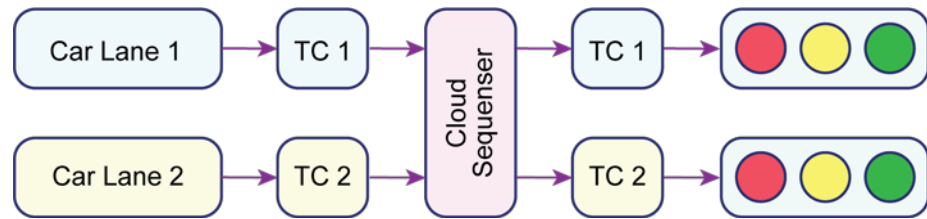


Рис. 4.12 – Структура хмарного управління транспортом

Тут Cloud Sequencer представлений логічною схемою формування керуючого сигналу для включення зеленого світлофора, зображеної на рис. 4.10, 4.11. Модуль Car Lane позначає множину автомобілів, що створюють лінію перетину перехрестя, яка трансформується у відповідний регістр з кількістю одиниць, що дорівнює кількості машин. Модуль TC є телекомунікаційний канал, що передає координати кожного автомобіля в хмару, а також із хмари до світлофора (автомобілю). Якщо число ліній перетину перехрестя більше двох, то структура управління збільшується пропорційно до їх кількості.

4.4. Структурно-аналітична модель хмарного управління

Являє собою кібер-фізичну систему у вигляді інтелектуальної хмари управління дорожнім рухом (Cyber Physical System – Smart Cloud Traffic Control) в реальному масштабі часу на основі створення хмарної інфраструктури дорожнього руху, інтегрованої з віртуальними вуличними світлофорами і дорожніми знаками, мобільними засобами ідентифікації автомобілів з метою підвищення якості та безпеки пересування транспортних засобів, мінімізації часових і матеріальних витрат при виконанні заданих маршрутів.

Інтелектуальне хмарне управління дорожнім рухом (Smart Cloud Traffic Control) має на меті включення параметра часу в цифрову карту планети, а також поступове перенесення дорожніх знаків і світлофорів в хмари, що радикально озеленює інфраструктуру дорожнього руху на землі і створює

потенційні можливості для економії: тисяч тонн металу на виготовлення світлофорів і непотрібних автомобільних номерів, мільйонів кіловат годин електроенергії на підтримку працездатності, мільйонів доларів на установку світлофорів, дорожніх знаків і експлуатаційні витрати, а також зменшення часу установки і актуалізації світлофорів у віртуальній інфраструктурі міст з декількох днів до декількох хвилин.

Формальна модель кіберсистеми представлена у вигляді двох хмарних компонентів або механізмів [1]: 1) f – моніторинг і управління; 2) g – виконавчі інфраструктурні механізми, які пов’язані між собою сигналами моніторингу, управління і ініціювання обох компонентів для реалізації сервісів. Аналітична форма [4] завдання TCS-системи і її структурний еквівалент зображені на рис. 4.13.

$$\begin{aligned}
 A &= (f, g, \mu, v, X, R, Y, P), \\
 \begin{cases} Y_t = f[(X, R, \mu)_t, Y_{t-1}]; \\ P_t = g[(X, R, v, Y)_t]. \end{cases} \\
 X &= (v, p, s); Y = [G(k), L, M, \bar{P}]; \\
 R &= (G, \bar{P})_R; P = (G(k), L, M, \bar{P})_P; \\
 \mu &= [G(k), L, M, \bar{P}]; \\
 v &= \{L_i, L_t, L_h, L_x\} = L; \\
 G\{R, P\} &= \{G(k), L, M, \bar{P}\}.
 \end{aligned}$$

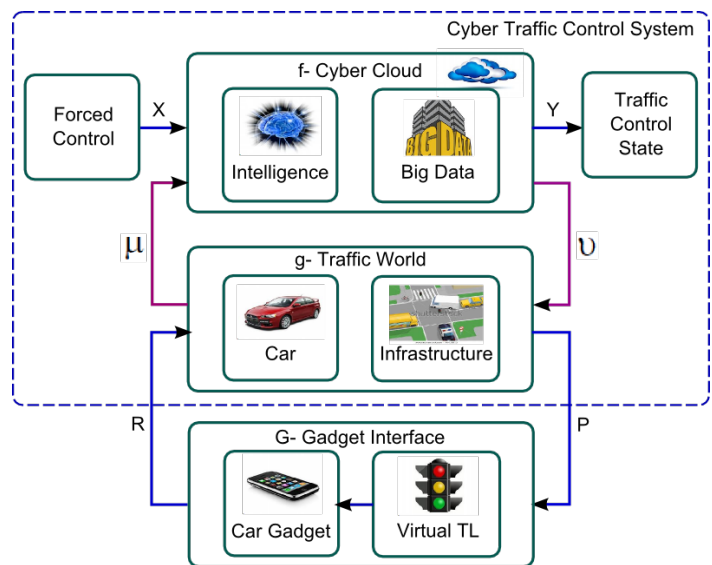


Рис. 4.13 – Аналітична і структурна форми TCS-системи

У моделі фігурують компоненти $A = (f, g, \mu, v, X, R, Y, P)$ відповідно: блоки управління і виконання, сигнали моніторингу та управління, входи керуючих завдань і виконавчих ресурсів, виходи індикації стану алгоритму реалізації завдання і надання сервісу. Тут також присутні сигнали зовнішнього управління дорожнім рухом $X = (v, p, s)$ для регулювання проїзду урядових

персон, поліцейських машин і автомобілів спеціального призначення відповідно. Сигнали управління транспортом $\nu = \{L_i, L_t, L_h, L_x\} = L$ використовують віртуальний світлофор, який працює в режимах: 1) інтелектуальний, функціонально залежний від дорожньої обстановки; 2) автоматичний, з фіксованими періодами перемикання; 3) ручний віртуальний режим на основі цифрового моніторингу перехрестя на екрані поліцейського комп'ютера, аналог – управління повітряним транспортом за допомогою монітора диспетчера в аеропорту; 4) екстренна зупинка $L_x \square \nu$ транспортного засобу за цифровим запитом поліції, який візуалізується на екрані монітора гаджета автомобіля. Хмарному моніторингу підлягають відповідно: $\mu = [G(k), L, M, \bar{P}]$ всі мобільні гаджети автомобілів з їх координатами, стан світлофорів, прив'язані до карти місцевості M , а також виконання замовлених маршрутів руху транспорту.

Перенесення світлофора $L \square \{\nu, P\}$ з реального перехрестя на хмарний завершує створення віртуальної інфраструктури планети, утворюючи замкнутий цикл системи моніторингу та управління з єдиним реальним компонентом у вигляді мобільного гаджета $G\{R, P\} = \{G(k), L, M, \bar{P}\}$ учасника дорожнього руху (УДД). Гаджет реалізує функцію інтерфейсу для зв'язку з хмарою: R-вхід до хмари – замовлення сервісу $R = (G, \bar{P})_R$ (делегування в хмару ID-гаджета і шляхи пересування) і P-вихід з нього – отримання сервісу $P = [M, G(k), \bar{P}, L]$ (карта, координати гаджета, оптимальний маршрут, світлофор).

Учасник отримує сервіс віконного скролінгу карти і сигнали світлофорів в реальному масштабі часу за шляхом руху \bar{P} , якщо він делегує свій гаджет в хмару. При замовленому шляху він додатково отримує квазіоптимальний маршрут руху і пріоритетний проїзд світлофорів. Насправді систему з позиції користувача створюють два компоненти: хмара і гаджет. При цьому новизна та оригінальність системи полягає в наданні хмарного сервісу – сигналів світлофора на екрані гаджета учасника дорожнього руху. Все інше: карти,

маршрути руху – вже існують і працюють. Впровадження запропонованої TCS-системи буде відбуватися шляхом створення віртуальних світлофорів, які дублюють реальні в синхронному режимі, а потім поступове усунення всіх фізичних пристроїв і знаків земної інфраструктури дорожнього руху за мірою набуття водіями нової технологічної культури еволюційним шляхом. Всі великі міста вже мають фактично централізоване комп'ютерне (хмарне) управління світлофорами. Тому перенесення світлофорів і дорожніх знаків до хмари не буде пов'язане з істотними додатковими витратами, а скоріше навпаки великі експлуатаційні витрати на підтримку світлофорної і знакової інфраструктури міст в працездатному стані трансформуються в нуль.

Мобільний гаджет транспортного засобу G є головним керуючим впливом для TCS-хмари, так само як і воно є основним споживачем світлофорних сигналів L управління рухом автомобіля, що виводяться на лобове скло, $L = \{L_i, L_t, L_b, L_x\} = F(L, G, V, T, D, \vec{P})$ де V – сигнали спецуправління, T – програмований цикл автономного управління світлофором, D – накопичені інтелектуальні статистичні дані по світлофору (проспекту, району), в тому числі, враховують пори року і доби, \vec{P} – актуальні замовлення на маршрути руху транспорту. Створення віртуальної системи світлофорів дає можливість практично без фінансових, часових, матеріальних і енергетичних витрат розміщувати шляхом програмування нові світлофори у віртуальному просторі, так само як і видаляти їх з хмари в процесі модернізації інфраструктури. Візуалізація на лобовому склі (мобільному моніторі) сигналів світлофора і голосове дублювання підвищить якість і безпеку дорожнього руху, знизить аварійну обстановку як для водія, так і для інфраструктури міст в цілому. Хмарний світлофор, як цифровий сигнал, на відміну від аналогового сприйняття водієм реального світлофора, є більш надійним засобом управління транспортом, в тому числі, і для подальшого впровадження в дорожній рух автопілота, що сприймає тільки детерміновані сигнали управління.

Учасники дорожнього руху ідентифікуються в хмарі гаджетом або iPhone, який парується при посадці в автомобіль. Статус учасника підвищується при проходженні світлофорів, якщо маршрут руху замовляється заздалегідь. Інші учасники дорожнього руху (пішоходи, мотоциклісти і велосипедисти) також мають право замовляти маршрут, підвищуючи свій статус для використання світлофорів. Пішоходи мають можливість отримати сервіс для замовлення комбінованого маршруту, що включає всі види наземного і підземного транспорту (автобуси, метро). Для автомобіля система управління формує функціонал, що оптимізує критерій якості обслуговування, який залежить від наступних змінних (час, довжина і якість маршруту):

$$Q = \min f(T, \vec{P}, K).$$

Для світлофора система управління формує функціонал, що мінімізує сумарний час простою автомобілів протягом доби (Z – цикл перемикавання світлофора):

$$Q = \min \frac{1}{n} \sum_{i=1}^n \left[\frac{T_i(\vec{P}_i, V_i, L_i, J_i)}{Z(\vec{P}_i, V_i, J_i)} \right]^{-1},$$

де в чисельнику і знаменнику представлені функціональні залежності часу простою і циклу від зазначених в дужках параметрів. Фактично підсумовуються частини світлофорних циклів, необхідні для проїзду кожному автомобілю через перехрестя. Якщо в результаті вийде оцінка якості, близька до одиниці $Q = 1$, то перехрестя функціонує нормально. В іншому випадку необхідно модифікувати цикл перемикавання або реконструювати перехрестя. Для інфраструктури міста або району система управління формує функціонал, що оптимізує критерій якості обслуговування автомобілів за проміжок часу (годину, добу), що залежить від сумарного часу проїзду автомобілів за заданими маршрутами з рекомендованою швидкістю, часу простою автомобілів на світлофорах і в пробках:

$$Q = \min \frac{1}{n} \sum_{i=1}^n \left[\frac{T_i(\vec{P}_i, V_i, L_i, J_i)}{T_i(\vec{P}_i, V_i)} \right]^{-1}.$$

Для проспекту міста система управління формує функціонал, що оптимізує критерій сумарного часу проїзду автомобілів від початку і до кінця вулиці за проміжок часу (годину, добу):

$$Q = \min \frac{1}{n} \sum_{i=1}^n \left[\frac{T_i(V_i, L_i, J_i)}{T_i(V_i)} \right]^{-1}.$$

Деталізована структура кібер-фізичної системи управління і моніторингу представлена на рис. 4.14, де основними блоками є: гаджет автомобіля, який одночасно є сенсором і активатором TCS, а також хмара, яка ділиться на дві частини. Перша з них g містить існуючу в часі інфраструктуру з компонентами: карта місцевості, координати гаджета – автомобіля, світлофори і дорожні знаки, а також пам'ять для зберігання замовлених маршрутів і статистики пересування транспорту. Друга частина хмари f представлена блоками пам'яті, моніторингу та управління, а також модулем захисту від несанкціонованого доступу.

Таким чином, запропонована інноваційна привабливість TCS-хмари характеризується наявністю взаємопов'язаних компонентів: фізичних – автомобіль з гаджетом і віртуальних – підсистем моніторингу та управління дорожнім рухом на основі хмарних світлофорів, що дає можливість: 1) квазіоптимально управляти кожним транспортним засобом в режимі реального часу на основі використання існуючих каналів зв'язку і мобільних гаджетів, спарених з автомобільними комп'ютерами; 2) оптимізувати процеси оптимального за часом, витратами і якістю управління дорожнім рухом для вирішення соціальних, гуманітарних, економічних, кримінальних, страхових і екологічних проблем; 3) радикально зменшити вартість реальної інфраструктури дорожнього руху і заощадити: матеріали для виготовлення дорожніх знаків і світлофорів, електроенергію на забезпечення їх

працездатності, фінанси на установку світлофорів і експлуатаційні витрати за рахунок перенесення світлофорів у віртуальну інфраструктуру дорожнього руху планети.

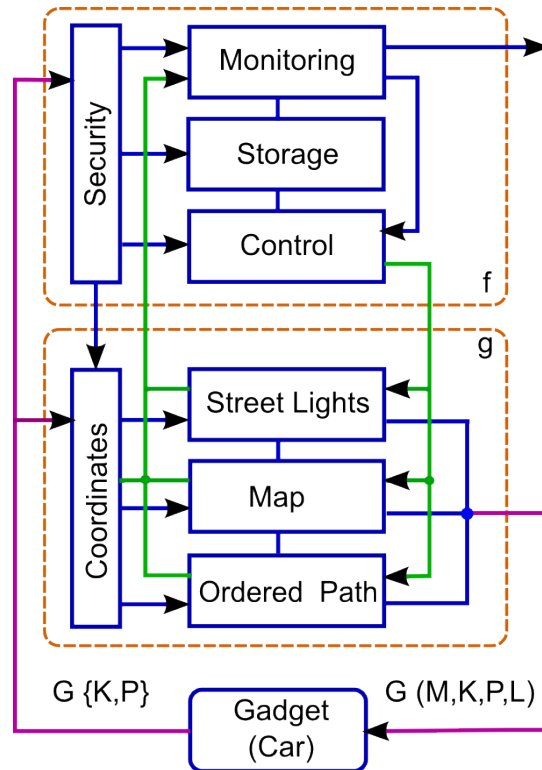


Рис. 4.14 – Структура TCS-системи

Алгоритм роботи хмарної системи управління містить наступні кроки: при посадці в автомобіль водій за допомогою гаджета замовляє маршрут пересування шляхом звернення до кібер-системи, яка верифікує валідність користувача за допомогою блоку «Security», який дозволить вхід до хмари, що визначає координату гаджета або автомобіля у вікні карти місцевості, обчислює оптимальний шлях у часі і просторі, а також здійснює супровід транспорту в реальному часі шляхом надання сервісів по управлінню за допомогою хмарних світлофорів, прив'язаних до координат або перехресть на маршруті руху. За відсутністю замовленого шляху користувач отримує тільки вікно карти місцевості зі світлофорами на поточному шляху пересування транспортного засобу, але при цьому сама кіберсистема може запропонувати

найбільш ймовірний маршрут руху, виходячи зі статистики поїздок, що зберігається в блоці історії «Storage». Гаджет може бути використаний і пішоходом для переходу через перехрестя і проспекти, замовлення маршруту пересування на основі використання існуючих транспортних засобів: автобус, метрополітен, поїзд, літак. Режими роботи кіберсистеми для користувача: 1) супровід учасника дорожнього руху шляхом надання інфраструктурної карти, координат і сигналів світлофора; 2) замовлення маршруту пересування, коли до функціональності першого режиму додається квазіоптимальний шлях, що мінімізує час, дистанцію, якість пересування, включаючи статистичну світлофорну зелену хвилю по шляху проходження; 3) особливі замовлення маршруту учасниками $X=(v,p,s)$, коли попереду та за шляхом проходження спеціального транспорту забезпечується запланована і гарантована зелена хвиля, що пропонує іншим учасникам дорожнього руху не перешкоджати спеціальним машинам, аж до зупинки, шляхом цифрової та голосової ідентифікації відповідних сигналів управління на екрані гаджетів автомобілів в зоні взаємного інфраструктурного впливу транспорту.

На рис. 4.15 основним компонентом є пам'ять як будь-яка субстанція для зберігання інформації, на якій визначаються структурні компоненти обчислень (Cloud Control і Car & Infrastructure Execution). Експлуатація системи здійснюється за допомогою моніторингу (сигналів датчиків) і управління (сигналів актуації) компонентами комп'ютерної системи. Входи і виходи системи з'єднуються із зовнішнім кібер-фізичним простором.

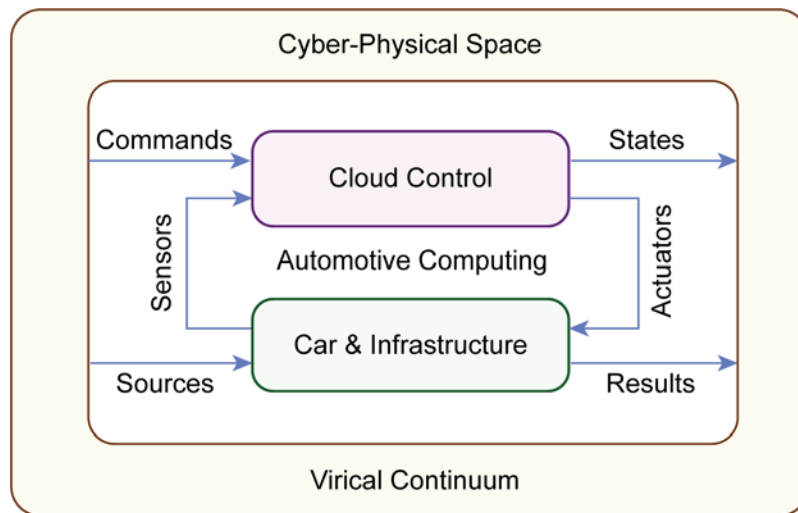


Рис. 4.15 – Automotive Computing for Cloud Traffic Control

На рис. 4.16 показана кібер-фізична система для цифрового моніторингу і хмарного управління трафіком, яка реалізована за технологією Internet-of-Things. Інноваційна особливість визначається відсутністю входів і виходів, які ідентифікуються з Big Data.

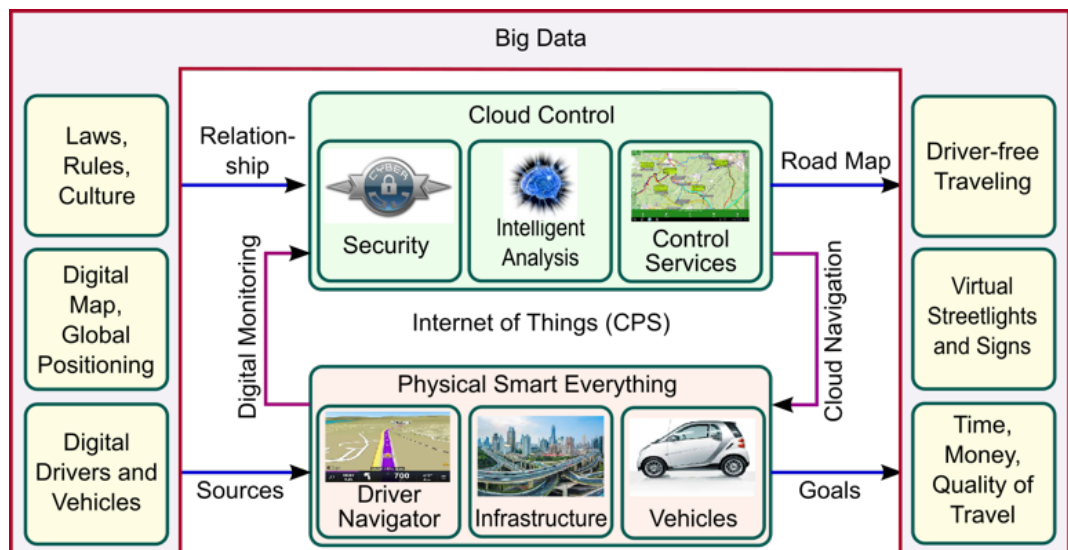


Рис. 4.16 – Global CPS for digital monitoring and cloud traffic control

Це означає, що входні дані пов'язані з інформаційними масивами кібер-фізичного простору. Результат аналізу даних, що надається хмарними

службами CPS, записується в спеціалізовані масиви великих даних. Природно, кіберфізична система має монітори і інтелектуальні датчики, розташовані в фізичному просторі планети. Ці монітори: сучасні автомобільні комп'ютери і гаджети водіїв або мандрівників. Гаджети є виконавчими механізмами, які контролюють маршрутизацію транспортних засобів. Обов'язковою умовою автономного пересування транспорту є глобальне позиціонування автомобіля в оцифрованому фізичному просторі на основі цифрової карти Google з точністю до 10 см.

На рис. 4.17 представлена доменно і IP-ідентифікована структура взаємодії кіберфізичних компонентів, що реалізують хмарний комп'ютинг-сервіс Cloud Traffic Control в контейнерно-орієнтованому середовищі компанії Docker, що працює в Google Computing Engine. Мета – створення відкритого і доступного для користувача сервісів Cloud Traffic Control, призначених для зменшення часових і матеріальних витрат на виконання заданого маршруту руху.

Задачі: 1) Створення хмарного середовища визначення квазіоптимального маршруту на основі контейнетов. 2) Розробка структури прототипу хмарного управління вуличним світлофором. 3) Інтеграція сервісів мовного введення інформації, зберігання історії виконання маршрутів, захисту даних і моніторингу дорожньої обстановки. 4) Верифікація структури і функціональностей хмарного сервісу.

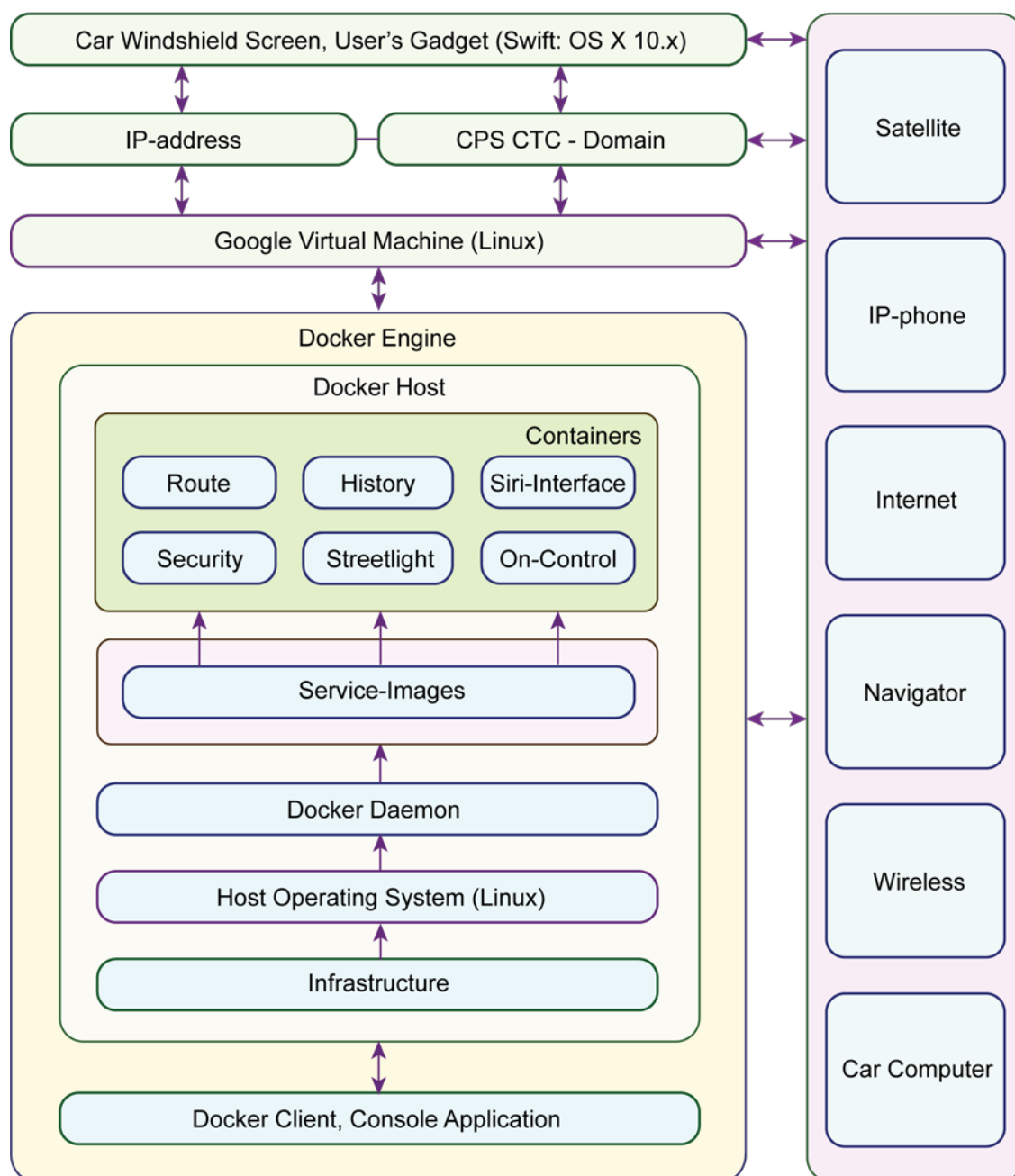


Рис. 4.17 – Хмарні сервіси Cloud Traffic Control

Огляд ринку проектування сучасних програмних продуктів показав, що в останні три роки набирають популярність гнучкі і масштабовані технології проектування програмних продуктів, що використовують контейнеризацію, орієнтовані на мінімізацію циклу time-to-market. Основна ідея контейнера полягає в створенні захищеної і доступного віртуального середовища проектування і експлуатації у кіберпросторі на основі використання локальної або глобальної апаратури. На інфраструктуру встановлюється практично

будь-яка операційна система Host operating system, зручна для розробника, наприклад, Ubuntu, Debian. У локально виконаній hardware інфраструктурі контейнерів можна інсталиювати також OSX і Windows.

Розробник інсталиюється Docker Engine, якщо він не встановлений спочатку, який за допомогою Docker Daemon (реалізація на Golang, C), управляє контейнерами, запущеними на основі попередньо побудованих образів (Images). Контейнери Containers уявляють собою легковажну віртуальну машину, під якою працюють функціональні блоки: Route, History, Siri, Security, Streetlight, Online Control, реалізовані на Swift (або Jawa), і відповідні бібліотеки. Docker Engine представляє собою комп'ютерингову систему контейнеризації, яка складається з двох взаємодіючих між собою компонентів: Docker Host, Docker Client (Console Application), які можна інтерпретувати як виконавчий Host і керуючого (Client) механізму хмарної системи Cloud Traffic Control. Далі Docker Engine є компонентом, який входить до складу Google Virtual Machine, якій ставиться у відповідність IP-address and CyUni domain, що робить розробку, а далі хмарний сервіс, доступним для фахівців і студентів. Останньою ланкою у вертикальній зв'язці (Back-End, Front-End) є User's Gadget (Swift OS X.10x). На кінцевому пристрої користувача встановлюються додатки, здатні привести результати роботи хмарних сервісів Cloud Traffic Control до мультіоконному формату візуалізації даних, зручною для сприйняття людиною. Слід зазначити, що набагато більш технологічно створити Front-End у вигляді додатку на JavaScript, HTML, CSS, який буде працювати через Internet browser. Це гарантує відсутність блокування клієнта з боку постачальника конкретної ОС (vendor lock). Для мобільних пристроїв доцільніше використовувати додаток індивідуального клієнта під кожну ОС. Переваги структурної організації хмарного сервісу Cloud Traffic Control: 1) Захищеність і доступність контейнерів для командної розробки і верифікації моделей і методів. 2) Глобальна доступність сервісів в кіберпросторі планети. 3) Контейнерна розширюваність хмарного комп'ютерингу при реалізації нових мікросервісів, що відповідають додатковим

методам і алгоритмам Cloud Traffic Control. 4) Просторово-часова інваріантність проектування і експлуатації кібер-фізичної системи, спрямована на доступність хмарних сервісів з будь-якої точки планети в будь-який час доби. 5) Інфраструктурна незалежність хмарного віртуального комп'ютингу забезпечує безвідмовну роботу невидимою апаратури, її своєчасну замінюваність на стороні Google (Amazon, IBM) без додаткових фінансових витрат для власника сервісів.

4.5 Модельно-фізичні експерименти на road-інфраструктурі

Заміна традиційного світлофора з фіксованим часом перемикання сигналів на розумне online управління дорожнім рухом (підрозділ. 4.3), залежне від фактичних транспортних потоків, істотно впливає на час проходження перехрестя автомобілем. Ефективність роботи перехрестя може бути підвищена за рахунок скорочення часу простою зеленого сигналу, коли на дозволений «зелений» а-смузі немає автомобілів, а на альтернативній b-смузі вони є. В цьому випадку розумний світлофор перемикається для пропуску автомобілів по b-смузі. Практично розумний світлофор виконує роль постового-регулювальника дорожнього руху, який здатний розрулити складні дорожні колізії на перехрестях, якщо є простої зеленого сигналу світлофора. Час простою зеленого сигналу світлофора на непересічних лініях перехрестя за наявності транспортних засобів на червоному світлофорі є резерв часу для підвищення пропускну здатності за рахунок розумного регулювання. Тому функція зміни стану світлофора ставиться у залежність не прямо від транспортного потоку, а від резерву часу, що визначається інтервалом простою зеленого кольору світлофора (T_0) на обох лініях (a, b) руху автомобілів: $F = f([T_0]_i^a, [T_0]_i^b)$.

Діаграми взаємодії транспортних потоків на двох пересічних лініях руху автомобілів при зеленому сигналі світлофора представлені на рис. 4.18.

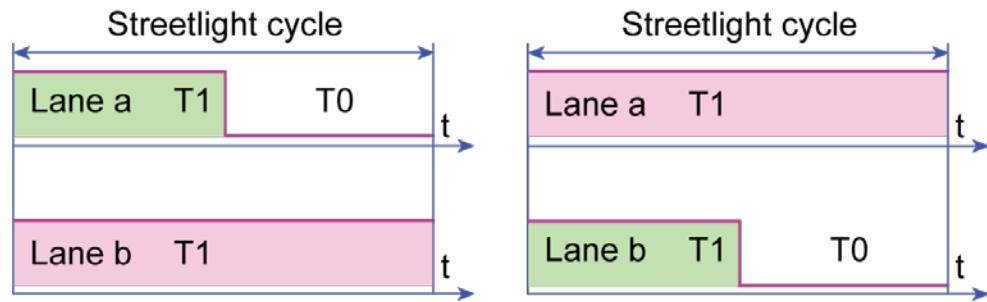


Рис. 4.18 – Взаємодія транспортних потоків на зеленому світлофорі

Тут представлені два зелених сигнали на напрямках перетину, де інтервал T_0 – час простою зеленого світлофора або резерв, який можна використовувати для перпендикулярного напрямку руху. Інтегральний час простою зеленого світлофора D на обох лініях руху рівнозначного перехрестя може бути метрикою оцінювання пропускної здатності (n – кількість циклів в інтервалі спостереження):

$$T_i^a = [T_0]_i^a + [T_1]_i^a; T_i^b = [T_0]_i^b + [T_1]_i^b; D = \sum_{i=1}^n ([T_0]_i^a + [T_0]_i^b).$$

Природно, що параметр D можна і потрібно зменшувати до нуля шляхом введення розумного світлофора, що працює під управлінням реальних транспортних потоків в режимі online. На рис. 4.19 представлені графіки проведених натурних і модельних експериментів на 10 "рівнозначних" перехрестях проспекту Науки в Харкові, обладнаних світлофорами. В залежності від часу доби (інтервал спостереження – 4 години) результати інтегрального оцінювання простою зеленого світлофора істотно відрізняються: в нічний час зелений світлофор на другорядних вулицях, що перетинають проспект, максимально впливає на затримку транспорту, що рухається прямо по проспекту.

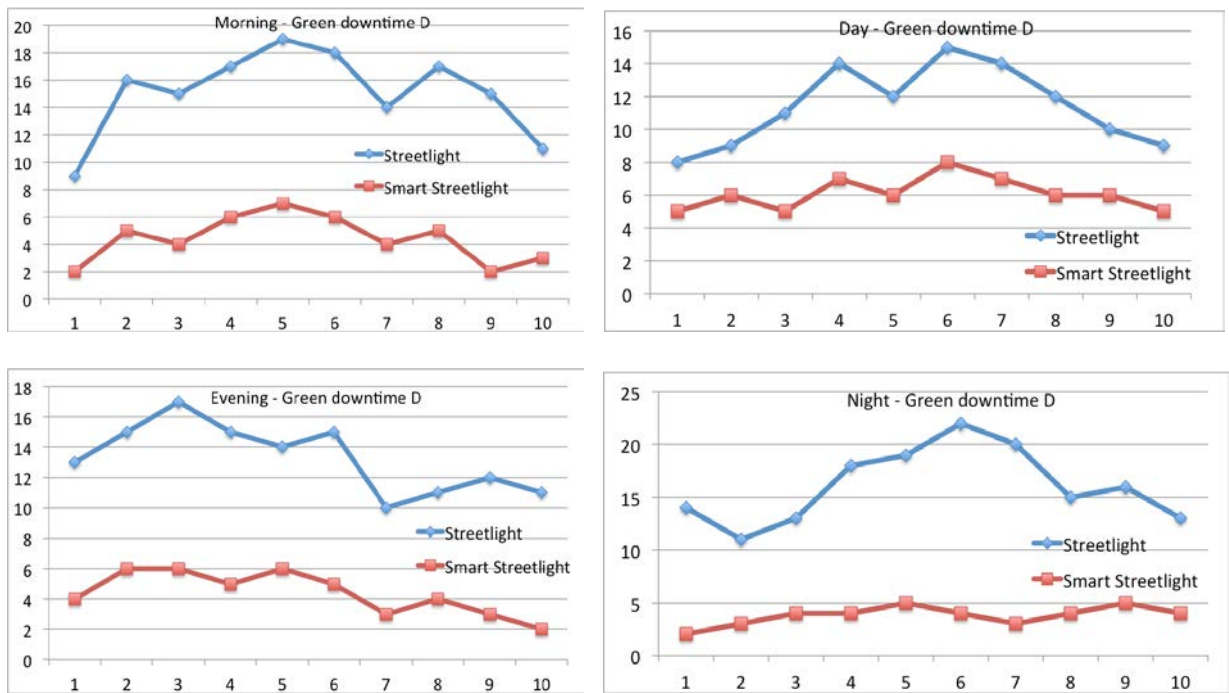


Рис. 4.19 – Графіки простоїв на звичайному і розумному світлофорах

Нульове значення часу простою при використанні розумного світлофора пов'язано з інертністю перемикання сигналу через жовтий колір і часом переїзду автомобіля через перехрестя, що становить перехідний процес транспортних потоків на перетині доріг при зміні сигналів світлофора.

Таким чином, впровадження розумних світлофорів дозволить на 10-30 відсотків снизити час простою зеленого сигналу, що в масштабах міста скоротить інтегральний час знаходження транспорту в шляху і витрати на енергоносії не менше, ніж на 10 відсотків.

Далі представлені експерименти на двох типах топологій перехресть. Середньої складності міське перехрестя зображене на рис. 4.20. Воно містить по дві смуги руху в кожному з чотирьох напрямків, які забезпечують маневр автомобіля за трьома напрямками.

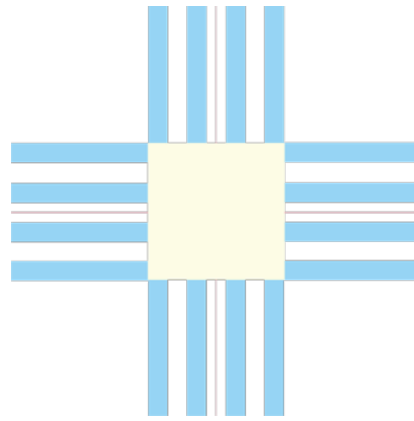


Рис. 4.20 – Приклад топології перехрестя

Сигнали світлофора формують чотири стани перехрестя, які забезпечують два типи топологій (діагональна і прямокутна) для формування транспортних потоків при проїзді перехрестя, представлених на рис. 4.21, 4.22. Перша і друга топології забезпечують два типи проїзду, які створюють метрику (n – число фрагментів смуг P_i до і після перетину перехрестя):

$$Q = \frac{1}{n} \times \sum_{i=1}^n P_i,$$

інтегрально визначаються наступними оцінками, що формують прохідність перехрестя:

$$Q_1 = \frac{10}{16} + \frac{10}{16} + \frac{6}{16} + \frac{6}{16} = \frac{32}{16} = 2,0; \quad Q_2 = \frac{8}{16} + \frac{8}{16} + \frac{8}{16} + \frac{8}{16} = \frac{8}{16} = 2,0.$$

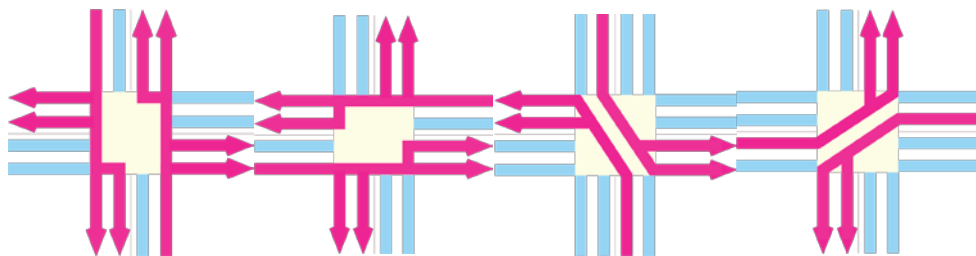


Рис. 4.21 – Діагональна топологія транспортних потоків

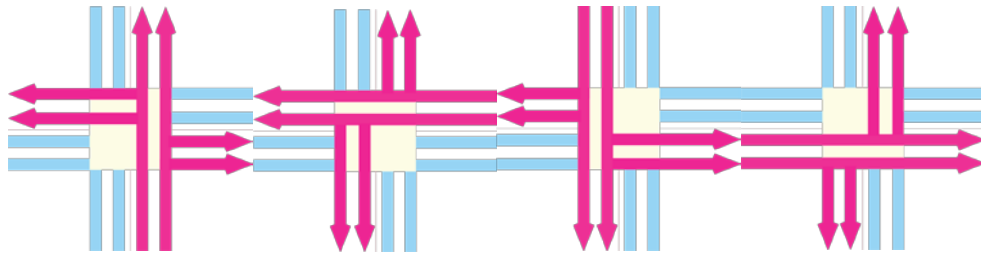


Рис. 4.22 – Прямокутна топологія транспортних потоків на світлофорі

Таким чином, обидві топології мають однакові оцінки. Однак друга топологія транспортних потоків, що формуються світлофорами, є більш ефективною для підвищення пропускної здатності, якщо необхідно швидко розвантажити транспортний потік з однієї вхідної черги. Важливі не тільки потенційні смуги перехрестя, але управління зеленими сигналами світлофора, які фактично формують транспортні потоки або ефективність використання смуг.

Кількість автомобілів N , що проїжджають через рівнозначне перехрестя, можна вирахувати, задавши наступні параметри відповідно: V_{car} , L_{car} , ΔT , τ^x , τ^l , S – швидкість автомобіля на перехресті; довжина автомобіля з дистанцією до транспорту, що йде попереду; проміжок часу, за який визначається кількість транспорту, що проїхав через перехрестя; час утримання жовтого сигналу світлофора; час утримання зеленого сигналу світлофора; кількість станів світлофора на перехресті. Формула для обчислення пропускної здатності однополосного перехрестя матиме такий вигляд:

$$N = \frac{V_{\text{car}}}{L_{\text{car}}} \left[\Delta T - \frac{S \times \tau^x \times \Delta T}{(\tau^l + \tau^x) \times S} \right] = \frac{V_{\text{car}}}{L_{\text{car}}} \left[\Delta T - \frac{\tau^x \times \Delta T}{(\tau^l + \tau^x)} \right].$$

Підставивши дані за всіма параметрами, заданими у формулі, можна отримати модельне число транспортних засобів, що проїжджають через перехрестя без колізій, в даному випадку за одну годину або 3600 секунд

(діагональна топологія транспортних потоків на перехресті) в одному напрямку:

$$N = \frac{2,77m/c}{7m} \left[3600c - \frac{4c \times 3600c}{(30c + 4c)} \right] = 1257.$$

З урахуванням існування P паралельних транспортних смуг, що проходять через перехрестя, пропускна здатність буде пропорційно збільшуватися:

$$N = P \times \frac{V}{L} \left[\Delta T - \frac{\tau^x \times \Delta T}{(\tau^1 + \tau^x)} \right].$$

У прямокутної (діагональної) топології транспортних потоків на перехресті на кожному сигналі світлофора число транспортних паралельних потоків дорівнює двом ($P = 2$). Тому модельне значення пропускної здатності буде визначатися таким виразом:

$$N = 2 \times \frac{2,77m/c}{7m} \left[3600c - \frac{4c \times 3600c}{(30c + 4c)} \right] = 2504.$$

У реальності пропускна здатність перехрестя може виявитися істотно нижче. Це пов'язано з тим, що існують прості зеленого сигналу світлофора, зтягування жовтого сигналу, наявність інертності в реакції водіїв на дорожню обстановку, аварійні ситуації, проїзди через перехрестя спеціальних транспортних засобів, транспортів великої довжини. Підтвердженням цього факту є вимір реальних транспортних потоків на діагональному перехресті в місті Харкові. Виявилося, що за 4 години він обслужив 8,4 тисячі автомобілів в період з 9 до 13 годин. Модельний експеримент на перехресті показав можливу пропускну здатність транспорту за 4 години, рівну 10016

автомобілів. Це на 19 відсотків краще, ніж в реальності. Виключити істотну різницю між модельними і реальними потоками транспорту, які проїжджають через перехрестя, слід шляхом впровадження розумних світлофорів, які враховують вхідні транспортні потоки на перехрестях, попереджувальний час простою зеленого сигналу.

4.6 Висновки до розділу 4

Важко передбачити і перерахувати всі позитивні соціальні, технологічні та технічні наслідки радикального перетворення існуючого світу після впровадження цифрових сервісів інтелектуальної хмари дорожнього руху. Через 5 років слід очікувати масової появи безпілотного транспорту при виконанні маршрутів [SAE World Congress eXperience 2017, Detroit, Forbes]. Для найближчого майбутнього нижче представлені інноваційні науково-технічні рішення соціальних, гуманітарних, економічних та екологічних проблем, пов'язані з хмарним моніторингом і управлінням дорожнього руху.

Наукова новизна Cyber Physical Systems "Smart Cloud Traffic Control" характеризується введенням в цифрову карту планети параметра часу і перенесенням світлофорів у віртуальний кіберпростір, що завершує створення віртуальної інфраструктури дорожнього руху для підвищення якості життя водіїв, зменшення часу перебування в дорозі і витрат на паливо, економії тисяч тонн металу для виготовлення світлофорів і дорожніх знаків, мільйонів кіловат-годин електроенергії на підтримку працездатності, мільйонів доларів на установку світлофорів і експлуатаційні витрати, а також зменшення часу установки і актуалізації світлофорів і дорожніх знаків у віртуальній інфраструктурі міст до декількох хвилин, що у сукупності дає можливість автоматизувати процеси квазіоптимального управління транспортом і дорожнім рухом в режимі реального часу і вирішувати соціальні, гуманітарні, економічні та екологічні проблеми.

Практична значимість визначається наданням нових послуг державним структурам, учасникам дорожнього руху, дорожній поліції, спеціальним службам і організаціям:

1) Сервіс безперешкодного руху за заданим маршрутом спеціалізованих машин або кортежів (перевезення дітей, важливі державні чиновники, швидка допомога, пожежна служба, військові колони, небезпечні вантажі).

2) Сервіс оптимального управління віртуальними світлофорами в режимі on-line на дорогах і перехрестях за допомогою точного цифрового моніторингу дорожнього руху шляхом використання гаджета водія або комп'ютера автомобіля, що дає можливість мінімізувати час проходження маршруту усіма учасниками дорожнього руху.

4) Сервіс інтелектуальної історії пересування автомобіля, що має віртуальну модель у кіберпросторі – індивідуальну комірку в хмарі, інваріантну по відношенню до водіїв, які обслуговують транспортний засіб, що дає можливість відстежити будь-які пересування транспортного засобу в минулому, а також прогнозувати бажані маршрути і поїздки в майбутньому вже без участі водія.

5) Сервіс хмарного моніторингу (мобільних) цифрових автometrічних паспортів транспортних засобів в режимі on-line дозволить прибрати автомобільні номери з системи обліку, і як наслідок: виключити безпосередню участь дорожньої поліції в фіксації порушень правил дорожнього руху (перевищення швидкості, проїзд на заборонні сигнали світлофорів, порушення правил маневрування, легкі зіткнення); заощадити тисячі тонн металу на виготовлення металевих номерів і спростити реєстрацію автомобілів при покупці з декількох днів до декількох хвилин; автоматизувати оформлення ДТП без участі співробітників дорожньої поліції шляхом цифрового моніторингу цифрової карти події, скопійованої з хмари; скоротити чисельний склад дорожньої поліції, оскільки історія переміщень автомобіля і його дорожніх порушень стає абсолютно прозорою для хмари, що дозволить автоматично списувати з рахунків водія вартість порушень відповідно до

законодавства конкретної країни; виключити корупцію у відносинах водія з дорожньою поліцією, завдяки неможливості стерти інформацію про порушення в хмарі; ліквідувати криміналітет в області викрадення автомобілів, завдяки вбудованому в машину цифровому автOMETричному паспорту, що забезпечує цілодобову спостережуваність транспорту в режимі on-line, якщо автомобіль фізично не знищений; спростити легалізацію водія шляхом ідентифікації водія ліцензії (driver's license) в списку дозволених осіб цифрового паспорта хмарного осередку автомобіля за протоколом «blue tooth», що дозволяє усунути виготовлення паперів та довіреностей на водіння автомобіля іншими особами і виключити посередницькі послуги нотаріуса; зменшити кількість ДТП, істотно підвищити якість життя водіїв і пасажирів, завдяки тотальному моніторингу порушень і невідворотності покарань; зменшити автомобільні викиди вуглекислого газу за рахунок зниження часу простоїв на перехрестях і вибору оптимальних режимів і маршрутів пересування транспорту; забезпечити високу ринкову привабливість хмари за рахунок продажу сервісів компаніям і приватним особам, що гарантує отримання високого прибутку – від сотень мільйонів до мільярдів доларів, що масштабується залежно від площі покриття сервісами: міста, країни, всього світу.

Запропонована архітектура розумного хмарного світлофора, яка характеризується використанням логічних операцій і часом простою зеленого сигналу, що дає можливість на 10-30 відсотків зменшити час простою зеленого світлофора і істотно збільшити пропускну здатність транспортних потоків на перехресті доріг. Впровадження розумних світлофорів в масштабах міста дозволить скоротити інтегральний час знаходження транспорту в шляху і витрати на енергоносії не менше, ніж на 10 відсотків.

Основні результати розділу опубліковані в роботах [146, 149, 153-159].

ВИСНОВОК

Запропоноване дослідження частково вирішує проблему створення надійних масштабованих комп'ютерних сервісів цифрового моніторингу і хмарного управління трафіком, яка інтегрує два найактуальніших і модних наукових напрямки: транспорт і комп'ютеринг для отримання якісно нових умов життя в розумному автомобілі, що використовує кіберфізичну інфраструктуру дорожнього руху.

Ринкова привабливість дослідження. Впровадження хмарного сервісу управління транспортом призведе до збереження екології планети і до зменшення: 1) часу проходження замовлених маршрутів, 2) споживання енергетичних ресурсів і матеріальних витрат на створення і експлуатацію світлофорів, дорожніх знаків, автомобільних номерів, 3) числа аварій і крадіжок автомобілів.

Для досягнення поставленої *мети* – підвищення якості та безпеки дорожнього руху за рахунок створення кіберфізичної моделі комп'ютерної online взаємодії водія з хмарними сервісами керування автомобілем на основі цифрового моніторингу дорожньої інфраструктури і транспортних потоків, при використанні розумних сенсорів, засобів телекомунікації та навігації – були вирішені завдання, що мають *наукову новизну*, а саме:

1) *Вперше запропоновано* модель транспортного комп'ютерингу, яка характеризується кіберфізичною взаємодією автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

2) *Вперше запропоновано* архітектуру розумного хмарного світлофора, яка характеризується використанням логічних операцій і часом простою зеленого сигналу, що дає можливість істотно збільшити пропускну здатність транспортних потоків на перехресті доріг.

3) *Удосконалено* метрика і критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання.

4) *Удосконалено* алгоритм Дейкстри, який відрізняється можливістю аналізу кіберфізичної інфраструктури дорожнього руху для online пошуку квазіоптимального маршруту транспортного засобу в умовах виникнення колізій.

Практична реалізація моделей і методів цифрового моніторингу і хмарного керування автомобілем виконана в рамках створення і верифікації програмних компонентів кіберфізичної архітектури дорожнього руху «Cloud Traffic Control» з подальшим тестуванням модельних потоків транспортних засобів на ділянках дорожньої інфраструктури.

Отримані в процесі виконання дослідження наукові висновки і результати є достовірними, що підтверджується проведеними експериментами, тестуванням і верифікацією запропонованих моделей і методів моніторингу та управління дорожнім рухом. *Практична значимість наукових досліджень* підтверджується істотним зменшенням виконання маршрутів руху при моделюванні реальних процесів на фрагментах моделей дорожньої інфраструктури. Результати дисертації в складі моделей, методів і архітектури кіберфізичної інфраструктури впроваджені в навчальний процес Харківського національного університету радіоелектроніки (акт про впровадження від 03.09.2017) при читанні курсів: «Дискретна математика», «Cloud-Fog кіберфізичні системи». Розроблена архітектура розумного хмарного світлофора, а також модель транспортного комп'ютингу можуть бути імплементовані як компонент проекту при синтезі хмарного сервісу (Довідка від 28.09.2017, ТОВ «Перший інститут надійного програмного забезпечення»).

Список використаних джерел

1. The e-IRGSP4 project for e-IRG, Best Practices for the use of e-Infrastructures by large-scale research infrastructures. – 2015 February 24. – <http://e-irg.eu/>.
2. <http://e-irg.eu/documents/10920/277005/Best+Practices+for+the+use+of+e-Infrastructures+by+large-scale+research+infrastructures.pdf>
3. Geoffrey C. Fox, Web 2.0, grids and cyberinfrastructure / International Symposium on e-infrastructure Collaborative Technologies and Systems (CTS 2007). <http://ieeexplore.ieee.org/document/4621721/>
4. Sy Holsinger, Sergio Andreozzi, EGI: Implementing service management in a largescale e-Infrastructure // IEEE Network Operations and Management Symposium (NOMS). – 2014. – <http://ieeexplore.ieee.org/document/6838371/>
5. David C. H. Wallom, Matteo Turilli, Michel Drescher, Federating Infrastructure as a Service Cloud Computing Systems to Create a Uniform E-infrastructure for Research // IEEE 11th International Conference on e-Science (e-Science). – 2015. – <http://ieeexplore.ieee.org/document/7304286/>
6. E-infrastructure: The ecosystem for innovation, www.bis.gov.uk
7. Richard O. Sinnott, Gerson Galang, Martin Tomko, Robert Stimson. Towards an e-Infrastructure for Urban Research across Australia // IEEE 7th International Conference on E-Science (e-Science). – 2011. – <http://ieeexplore.ieee.org/document/6123291/>
8. Geoffrey C. Fox, Web 2.0, grids and cyberinfrastructure / Collaborative Technologies and Systems e-infrastructure (CTS 2007). – 2007. International Symposium on, <http://ieeexplore.ieee.org/document/4621721/>.
9. Luong Thi Thu Phuong, Nguyen Trung Hieu, Jin Wang, Sungyoung Lee, Young-Koo Lee, Energy Efficiency based on Quality of Data for Cyber Physical Systems // IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing. – 2011. – P.232-241.
10. Jane Hunter, Charles Brooking, Wilfred Brimblecombe, Ross G. Dwyer, Hamish A. Campbell, Matthew E. Watts, Craig E. Franklin, OzTrack. E-

Infrastructure to Support the Management, Analysis and Sharing of Animal Tracking Data, eScience (eScience), 2013 IEEE 9th International Conference on, <http://ieeexplore.ieee.org/document/6683901/>

11. Sy Holsinger, Sergio Andreozzi, EGI: Implementing service management in a largescale e-Infrastructure, Network Operations and Management Symposium (NOMS), 2014 IEEE, <http://ieeexplore.ieee.org/document/6838371/>.

12. Sheikh Mahbub Habib, Sebastian Ries, Max Muhlhauser, Towards a Trust Management System for Cloud Computing // IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). – 2011.

13. Mohamed Almorsy, John Grundy, Amani S. Ibrahim. Collaboration-Based Cloud Computing Security Management Framework // 2011 IEEE International Conference on Cloud Computing (CLOUD), <http://ieeexplore.ieee.org/document/6008731/>

14. Mahmoud Al-Ayyoub, Yaser Jararweh, LoAi Tawalbeh, Elhadj Benkhelifa, Anas Basalamah, Power Optimization of Large Scale Mobile Cloud Computing Systems // 3rd International Conference on Future Internet of Things and Cloud (FiCloud). – 2015. – <http://ieeexplore.ieee.org/document/7300885/>

15. Zhen Jiang Li and Cheng Chen, Cloud Computing for Agent-Based Urban Transportation Systems // Intelligence transportation systems, published by IEEE Computer Society. – January/February 2011. – P.73-79.

16. Drescher M. MS518 EGI Platform Roadmap // June 2014, [online] Available: <https://documents.egi.eu/document/2232>.

17. Habib, Sheikh Mahbub et al. “Trust as a facilitator in cloud computing: a survey.” *Journal of Cloud Computing: Advances, Systems and Applications*. – 2012. – P. 1-18. DOI: 10.1186/2192-113X-1-19

18. Sasko Ristov, Marjan Gusev, Aleksandar Donevski, "Security Vulnerability Assessment of OpenStack Cloud", Conference on Sixth International Computational Intelligence Communication Systems and Networks (CICSyN) – 2014. – Pp. 95-100,

19. New England E-infrastructure:

<http://www.rcuk.ac.uk/research/xrcprogrammes/OtherProgs/eInfrastructure/>

20. Sinnott R. O., Bayliss C., Galang G., Greenwood P., Koetsier G., Mannix D., Morandini L., Nino-Ruiz M., Pettit C., Tomko M., Sarwar M., Stimson R., Voorsluys W., Widjaja I. A data-driven urban research environment for Australia // IEEE 8th International Conference on E-Science (e-Science). – 2012. – Pp. 1-8.

21. Tawalbeh L., Jararweh Y., Ababneh F., Dosari F., Large scale cloudlets deployment for efficient mobile cloud computing // Journal of Networks. – 2015. – Vol. 10, no. 01.

22. Chronopoulos A.T. , Johnston C.M. A real-time traffic simulation using a communication latency hiding parallelization // IEEE Transactions on Vehicular Technology. – 2002. – Vol. 51, Issue 3. – Pp. 498 – 510. DOI: 10.1109/TVT.2002.1002499

23. Swapnaja Hiray, Rajesh Ingle. Context-Aware Middleware in Cyber Physical Cloud (CAMCPC) // International Conference on Cloud & Ubiquitous Computing & Emerging Technologies (CUBE). – 2013. – Pp. 42-47, 2013.

24. Arini Widhiyasi, Vasuky Mohanan, Muhammad Fermi Pasha, Rahmat Budiarto. Vertical Handover Scheme for Car-to-Car Communication based on IEEE 802.21 Standard // Second International Conference on Computer Engineering and Applications. – 2010. – P. 143-147.

25. Deyi Li, Xiaodong Wang, Wen He, Mu Guo, Tianlei Zhang, Study on Interaction Behaviours of Micro-Autonomous Vehicles // Tenth International Symposium on Autonomous Decentralized Systems. – 2011. – P. 399-406.

26. Zheng Li, Nenghai Yu, Zhuo Hao, A Novel Parallel Traffic Control Mechanism for Cloud Computing // 2nd IEEE International Conference on Cloud Computing Technology and Science. – 30 Nov. – 3 Dec. 2010. – P.376-382.

27. Visit Hirankitti, Jaturapith Krohkaew, An agent approach for intelligent traffic-light control // Proc. of the First Asia International Conference on Modelling & Simulation (AMS'07). – 27-30 March 2007.

28. Tian Hai, Zhang Yong, Cui Gui-mei, Design and realization of intelligent traffic light monitor and control system based on wireless control // 2012 31st Chinese Control Conference (CCC). – 25-27 July 2012. – <http://ieeexplore.ieee.org/document/6390937/>.

29. Smith R. H. and Chin D. C., Evaluation of an adaptive traffic control technique with underlying system changes // Winter Simulation Conference. – 1995. – P. 1124–1130.

30. Chen W., Chen L., Long Chen Z., and Liang Tu S., A realtime dynamic traffic control system based on wireless sensor network // ICPP Workshops. – 2005. – P. 258 – 264.

31. Hirankitti V., Krohkaew J., and Hogger C. J., A multi-agent approach for intelligent traffic-light control // World Congress on Engineering. – 2007. – P. 116–121.

32. Mohandas B., Liscano R., and Yang O., Vehicle traffic congestion management in vehicular ad-hoc networks // LCN. – 2009. – Pp. 655 – 660.

33. Hong Y. and Yang O. W. W., Design of adaptive pi rate controller for best-effort traffic in the internet based on phase margin // IEEE Trans. Parallel Distrib. Syst. – 2007. – Vol. 18, no. 4. – Pp. 550–561.

34. Cheng Hu, Yun Wang, A Novel Intelligent Traffic Light Control Scheme // Ninth International Conference on Grid and Cloud Computing. – 2010. – Pp. 372-376.

35. Jubair Mohammed Bilal, Don Jacob. Intelligent Traffic Control System // ICSPC. – 2007. – Pp. 496-499.

36. Haimeng Zhao, Xifeng Zheng, Weiya Liu. Intelligent Traffic Control System Based on DSP and Nios II // International Asia Conference on Informatics in Control, Automation and Robotics. – 2009. – Pp. 90-94.

37. Wu Hejun, Miao Changyun. Design of intelligent traffic light control system based on traffic flow // International Conference on Computer and Communication Technologies in Agriculture Engineering. – 2010. – Pp. 368-371.

38. Liang-Tay Lin, Hung-Jen Huang, Jim-Min Lin, Fongray Frank Young. A New Intelligent Traffic Control System For Taiwan // ITST 2009. – Pp. 138-142.
39. Chen Zhaomeng. Intelligent Traffic Control Central System of Beijing-SCOOT // MACE. – 2010. – Pp. 5067 – 5069.
40. Vivek Kumar Sehgal, Samruddh Dhope, Prince Goel. An Embedded Platform for Intelligent Traffic Control // UKSim Fourth European Modelling Symposium on Computer Modelling and Simulation. – 17-19 Nov. 2010. – P. 541-545.
41. Lawrence Y. Deng, Nick C. Tang, Dong-liang Lee, Chin Thin Wang, Ming Chih Lu, Vision Based Adaptive Traffic Signal Control System development // Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA). – 2005.
42. Jianwu Li, Zhengxian Huangpu. Design of fuzzy control system for city single highway intersection // Electrical Drive Automation. 2000. – Vol. 22, No. 2, Pp. 22-24.
43. Bingham E. Reinforcement learning in neuro-fuzzy traffic signal control // European Journal of Operational Research. – 2001. – 131. – Pp. 232-241.
44. Abdulhai B., Pringle R. Machine learning based adaptive signal control using autonomous Q-learning agent // Proceeding of the IASTED International Conference. Intelligent Systems and Control. – Honolulu, Hawaii, USA. – August 14-16, 2000. – Pp. 320-327.
45. Xiangjun Cheng, Zhaoxia Yang, Intelligent Traffic Signal Control Approach Based on Fuzzy-Genetic Algorithm // Fifth International Conference on Fuzzy Systems and Knowledge Discovery. – 18-20 Oct. 2008. – P. 222-225.
46. Mingshu Li, John Hallam, Louise Pryor, Stanley Chan, Ken Chong. Cooperative Intelligent System for Urban Traffic Problems // Proceedings of the IEEE International Symposium on Intelligent Control. – Dearborn, MI. – September 15-18, 1996. – Pp. 162-167.
47. Goldberg, D.E., Genetic algorithms in search optimization, and machine learning. – Addison Wesley. – 1989.

48. Jeffrey L. Adler, Victor J. Blue. A Cooperative Multi-Agent Transportation Management and Route Guidance System // Transportation Research. – 2002. – Part C, No.10. – Pp. 433-454,

49. John France and Ali A. Ghorbani. A Multiagent System for Optimizing Urban Traffic / Proceedings of the IEEE WIC International Conference on Intelligent Agent Technology (IAT'03). – 2003.

50. CHENG Xiangjun, YANG Zhaoxia, Distributed Traffic Signal Control Approach Based on Multi-Agent // Sixth International Conference on Fuzzy Systems and Knowledge Discovery. – 2009. – P.582-587.

51. Baogui Cao, Zhaosheng Yang, Haiyan Zhu, Fusion and Integration Framework Study on Road Network and Dynamic Traffic Data in Vehicle Navigation // Third International Symposium on Intelligent Information Technology Application Workshops. – 2009. – P.70-73.

52. Mohamed M. Atia, Shifei Liu, Heba Nematallah. Integrated Indoor Navigation System for Ground Vehicles with Automatic 3-D Alignment and Position Initialization // IEEE Transactions on Vehicular Technology. – 2015. – Vol. 64, Issue 4. – Pp. 1279 – 1292, <http://ieeexplore.ieee.org/document/7027835/>.

53. Ming Li, Zh. H. Liu, J.A. Huang. Artificial Landmark Positioning System Using Omnidirectional Vision for Agricultural Vehicle Navigation // Second International Conference on Intelligent System Design and Engineering Application (ISDEA). – 2012. – <http://ieeexplore.ieee.org/document/6173294/>.

54. Liu Dawei, Application of Assisted TDOA Technology in Vehicle Positioning and Navigation System // TENCON 2005. IEEE Region 10, <http://ieeexplore.ieee.org/document/4085074/>.

55. Ying Ji, Man Zhang, Gang Liu, Positions research of agriculture vehicle navigation system based on Radial Basis Function neural network and Particle Swarm Optimization // Natural Computation (ICNC). – 2010 Sixth International Conference on, <http://ieeexplore.ieee.org/document/5583145/>.

56. Mingke Fang, Lei Li, Wei Huang, Research of Hybrid Positioning Based Vehicle Interactive Navigation System // International Conference on Multimedia

Information Networking and Security (MINES). – 2010. –
<http://ieeexplore.ieee.org/document/5671177/>.

57. Kwak S.H., McKeon J.B., Clynych J.R. Incorporation of global positioning system into autonomous underwater vehicle navigation // Proceedings of the Symposium on Autonomous Underwater Vehicle Technology (AUV'92). – 1992. –
<http://ieeexplore.ieee.org/document/225224/>.

58. Song Ying, Yu Yang, Study on Vehicle Navigation System (VNS) with Real-time Traffic Information // International Conference on Computer Science and Software Engineering. – 2008. – P.1079-1082.

59. Kim S.-B., Bazin J.-C., Lee H.-K. Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data // IET Radar, Sonar & Navigation. – 2011. – Vol. 5, Issue 8. – P. 814 – 823, <http://ieeexplore.ieee.org/document/6036233/>.

60. Masaya Yoshikawa, Hidekazu Terai, IEEE. Car Navigation System based on Hybrid Genetic Algorithm // World Congress on Computer Science and Information Engineering. – 2009. – P. 62-64.

61. Holland J., Adaptation in Natural Artificial Systems. – University of Michigan Press. – 1992.

62. Fei-Yue Wang, Agent-Based Control for Networked Traffic Management Systems // IEEE Intelligent Transportation Systems. – 2005. – P.92-96.

63. Yu Shaowei, Li Xiuhai, Liu Qingling, Zhang Tongzhu, A Car-Steering Model Based on an Adaptive Cloud Neural Controller // International Conference on Digital Manufacturing & Automation. – 2010. – P. 7-10.

64. Kiattisin Kanjanawanishkul, Andreas Zell, Path following for an omnidirectional mobile robot based on model predictive control // IEEE International Conference on Robotics and Automation (ICRA'09). – 2009. –, <http://ieeexplore.ieee.org/document/5152217/>.

65. Sandor P. Fekete, Christiane Schmidt, Axel Wegener, Stefan Fischer, Recognizing Traffic Jams with Hovering Data Clouds // Second International

Symposium on Leveraging Applications of Formal Methods. Verification and Validation. – 15-19 Nov 2006. – P. 199-203.

66. Yin Zhu¹, Junli Wang¹, Huapu Lu, Study on the Intelligent Pre-control Model of Urban Arterial // IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. – 2008. – P.114-118.

67. You-Ren Chen, Keng-Pin Chen, Pao-Ann Hsiungy. Dynamic traffic light optimization and Control System using model-predictive control method // 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) – <http://ieeexplore.ieee.org/document/7795937/>.

68. Yang Zhao-sheng, Zhou Xi-Yang, Tian Chun-Lin, Research on traffic signal cycle optimization method based on bi-level programming model // 3rd International Conference on Computer Science and Network Technology (ICCSNT). – 2013. – <http://ieeexplore.ieee.org/document/6967178/>.

69. Xiao-Hua Yu, Stubberud A.R. Markovian decision control for traffic signal systems // Proceedings of the 36th IEEE Conference on Decision and Control. – 1997. – <http://ieeexplore.ieee.org/document/649773/>.

70. Ludovica Adacher, Marco Tiriolo, Distributed urban traffic signal optimization based on macroscopic model // Sixth International Conference on Innovative Computing Technology (INTECH). – 2016 – <http://ieeexplore.ieee.org/document/7845123/>.

71. Glenn T. Donovan, Position Error Correction for an Autonomous Underwater Vehicle Inertial Navigation System (INS) Using a Particle Filter // IEEE Journal of Oceanic Engineering. – July 2012. – Vol. 37, Issue 3. – Pp. 431 – 445. – <http://ieeexplore.ieee.org/document/6200890/>

72. Lee H., Optimal estimation of vehicle position in an acoustic transponder navigation system // IEEE International Conference on Engineering in the Ocean Environment. – Ocean 73. – <http://ieeexplore.ieee.org/document/1161299/>

73. Pedro Batista, Carlos Silvestre, Paulo Oliveira, Position and Velocity Navigation Systems for Unmanned Vehicles // IEEE Transactions on Control

Systems Technology. – May 2009. – Vol. 17, Issue 3. – P. 707 – 715. – <http://ieeexplore.ieee.org/document/4798174/>

74. Du J., Masters J., Barth M. Lane-level positioning for in-vehicle navigation and automated vehicle location (AVL) systems // Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems. – 2004. – <http://ieeexplore.ieee.org/document/1398868/>

75. Braden K., Browning C., Gelderloos H. Integrated inertial navigation system/Global Positioning System (INS/GPS) for manned return vehicle autoland application // Position Location and Navigation Symposium (IEEE PLANS). – 1990. – A Decade of Excellence in the Navigation Sciences. – <http://ieeexplore.ieee.org/document/66160/>

76. Satoh K., Hozumi H., Okada S. Autonomous mobile patrol system for nuclear power plants: field test report of vehicle navigation and sensor positioning // Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96). – <http://ieeexplore.ieee.org/document/571045/>

77. Jian Bu, Rui Sun, Hongyang Bai. Integrated method for the UAV navigation sensor anomaly detection // IET Radar, Sonar & Navigation 2017. – Vol. 11, Issue 5. – P. 847 – 853 – <http://ieeexplore.ieee.org/document/7914109/>

78. Weishan Li, Ming Bai, Runda Lu. The design of high accuracy differential positioning vehicle terminal based on BeiDou Navigation System // <http://ieeexplore.ieee.org/document/7867337/>

79. Hoang Van Dung, Trinh Minh Chi, Nguyen Thi Nga, A vehicle positioning and navigation system design using Mini2440 ARM and Linux // Third World Congress on Information and Communication Technologies (WICT). – 2013. – <http://ieeexplore.ieee.org/document/7113112/>

80. Azura Che Soh, Mohammad Hamiruce Marhaban, Marzuki Khalid, Modelling and Optimisation of a Traffic Intersection Based on Queue Theory and Markov Decision Control Methods // First Asia International Conference on Modelling & Simulation (AMS 2007). – 2007. – <http://ieeexplore.ieee.org/document/4148707/>.

81. Chin D.C., Spall J.C., Smith R.H. Evaluation of system-wide traffic signal control using stochastic optimization and neural networks // Proceedings of the American Control Conference. – 1999. <http://ieeexplore.ieee.org/document/786341/>.

82. Samuel Coogan, Murat Arcaç, Calin Belta. Formal Methods for Control of Traffic Flow // IEEE Control Systems Automated Control Synthesis from Finite-State Transition Models. – April 2017. – Vol. 37, Issue 2. – P. 109 – 128. – <http://ieeexplore.ieee.org/document/7879917/>

83. Kale M.M., Chipperfield A.J. Reconfigurable flight control strategies using model predictive control // Proceedings of the IEEE International Symposium on Intelligent Control. – 2002. – <http://ieeexplore.ieee.org/document/1157736/>

84. Trodden P., Richards A. Robust distributed model predictive control using tubes // American Control Conference. – 2006 – <http://ieeexplore.ieee.org/document/1656519/>

85. Center for automotive research // Annual report 2016.

86. Hitachi Inspire the Next, Hitachi automotive systems, Ltd. Guide to Hitachi Automotive Systems. – Printed in Japan. – 2016. – P. 5-12.

87. <http://www.hitachi-automotive-mm.com/en/>

88. Vejlupek Josef, Trailer backing-up assistant using ultrasound sensors based control units to safely back-up the car with trailer // 17th International Conference on Mechatronics – Mechatronika (ME). – 7-9 Dec. 2016. – <http://ieeexplore.ieee.org/document/7827834/>.

89. Sandor Marcon, Yasuhiro Komatsu, Akitomo Yamanaka. Linear motor coils as brake actuators for Multi-car elevator // International Conference Electrical Machines and Systems (ICEMS). – 8-11 Oct. 2007. – <http://ieeexplore.ieee.org/document/4412276/>.

90. Jakob Lombacher, Markus Hahn, Jürgen Dickmann, Detection of arbitrarily rotated parked cars based on radar sensors // 16th International Radar Symposium (IRS). – 24-26 June 2015. – <http://ieeexplore.ieee.org/document/7226281/>.

91. Chan Wei Hsu, Tsung Hua Hsu, Kuang Jen Chang. Implementation of car-following system using LiDAR detection // 12th International Conference ITS Telecommunications (ITST). – 5-8 Nov. 2012. <http://ieeexplore.ieee.org/document/6425157/>.

92. Iuliia Goncharova; Stefan Lindenmeier. An interoperable antenna for GPS and GLONASS services on a car // IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting. – 2015. – <http://ieeexplore.ieee.org/document/7304393/>.

93. Gerd Saala, Jochen Hopf, Stefan Lindenmeier. Small satellite car antenna for simultaneous reception of LHCP and RHCP signals // 3rd European Conference on Antennas and Propagation (EuCAP). – 2009. – <http://ieeexplore.ieee.org/document/5068167/authors>.

94. Tobias Haberle, Lambros Charissis. The Connected Car in the Cloud: A Platform for Prototyping Telematics Services // IEEE Software. – 2015. – Vol. 32, Issue 6. – Ph.11 – 17. – <http://ieeexplore.ieee.org/document/7310999/>.

95. Arini Widhiasi, Vasuky Mohanan, Muhammad Fermi Pasha, Rahmat Budiarto, Vertical Handover Scheme for Car-to-Car Communication, based on IEEE 802.21 Standard // Second International Conference on Computer Engineering and Applications. – 2010. – P. 143-147.

96. Deyi Li, Xiaodong Wang, Wen He, Mu Guo, Tianlei Zhang, Study on Interaction Behaviours of Micro-Autonomous Vehicles // Tenth International Symposium on Autonomous Decentralized Systems. – 2011. – P. 399-406.

97. Хаханов В.И., Меликян В.Ш, Саатчян А.Г., Шахов Д.В. «Зеленая волна» – облако мониторинга и управления дорожным движением // Армения. Вестник «Информационные технологии, электроника, радиотехника». 2013. – Вып. 16(№1). – С.53-60.

98. Hahanov V.I., Guz O.A., Ziarmant A.N., Ngene Christopher Umerah, Arefjev A. Cloud Traffic Control System // Proc. of IEEE East-West Design and Test Symposium. Rostov-on-Don. – 27-30 September 2013. – P.72-76.

99. Hahanov V., Gharibi W., Baghdadi Ammar Awni Abbas, Chumachenko S., Guz O., Litvinova E. Cloud traffic monitoring and control // Proceedings of the 2013 IEEE 7th International conference on intelligent data acquisition and advanced computing systems (IDAACS). – Berlin. – September 12-14, 2013. – P. 244-248.

100. Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И. Структура логического ассоциативного мультипроцессора // Автоматика и телемеханика. – 2012. – № 10. – С. 71-92.

101. Lu Antao, Li Yushan, Sun Yufang, Cao Chongzhen, Gao Kuigang, Xu Jing. Research on the Integrated Management of Highway Based on Radio Frequency Identification Technology // Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). – 2011. – Vol.3. – P. 116-119.

102. Pandit A.A., Talreja J., Mundra A.K. RFID Tracking System for Vehicles (RTSV) // First International Conference on Computational Intelligence, Communication Systems and Networks. – 2009. – P.160-165.

103. Jiang Lin-ying, Wang Shuai, Zhang Heng, Tan Han-qing. Improved Design of Vehicle Management System Based on RFID // International Conference on Intelligent System Design and Engineering Application (ISDEA). – 2010. – Vol. 1. – P. 844-847.

104. Chen Xue-Mei, Wei Zhong-Hua. Vehicle management system based on multi-node RFID cards // 30th Chinese Control Conference (CCC).– 2011.– P. 5497-5499.

105. Дудников С., Боенко И. Бесконтактная идентификация транспорта, основанная на RFID // Компоненты и технологии №1. – 2007. – http://www.kit-e.ru/assets/files/pdf/2007_01_140.pdf

106. Manikonda P., Yerrapragada A.K., Annasamudram S.S. Intelligent traffic management system // IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (Student).– 2011. – P. 119-122.

107. Samad T. Perspectives in Control Engineering Technologies, Applications, and New Directions // Intelligent Transportation Systems: Roadway Applications. – Wiley-IEEE Press. – 2001. – P. 348 -369.

108. Schutte J. Recent trends in automatic train controls // IEEE Intelligent Transportation Systems. – 2001. – P. 813 -819.

109. Zingirian N., Valenti C. Sensor clouds for Intelligent Truck Monitoring // IEEE Intelligent Vehicles Symposium (IV). – 2012.– P. 999-1004.

110. Branisso L.B., Kato E.R.R., Pedrino E.C., Morandin O., Tsunaki R.H. An Intelligent Autonomous Vehicle Management System // Second Brazilian Conference on Critical Embedded Systems (CBSEC). – 2012. – P. 42-47.

111. Brizgalov V.V., Chukhantsev V., Fedorkin E., Architecture of traffic control systems using cloud computing // International Conference and Seminar on Micro/Nanotechnologies and Electron Devices (EDM).– 2010.– P. 215-216.

112. Frank van Diggelen and Kathy Tan (2014) Interchangeability Accomplished // Tri-Band Multi-Constellation GNSS in Smartphones and Tablets – GPS World. – June 2014. – Pp. 46-52.

113. Бабаков В.Н. Высокоточная навигация.
<http://www.gisa.ru/102286.html>

114. Роффе А. Возможности высокоточной навигации –
<http://www.gisa.ru/102624.html>

115. <http://www.gisa.ru/103133.html>

116. <http://top.rbc.ru/economics/28/05/2014/926718.shtml>

117. Leandro, R., Landau H., Nitschke M., Glocker M., Seeger S., Chen X., Deking A., Ben Tahar M., Zhang F., Stolz R., Talbot N., Lu G., Ferguson K., Brandl M., Gomez Pantoja V., Kipka A.. GNSS Positioning Trimble TerraSat. Positioning: the Next Generation of cm-accurate Real-time. – GmbH, Germany. –2011. – RTX, ION-GNSS-2011, 2011, Portland, OR, USA.

118. Vollath, U., Buecherl, A., Landau, H. Pagels, C., Wagner, B. Multi-Base RTK using Virtual Reference Stations // Proceedings of ION GPS, Salt Lake City. – 2000. – P. 19-22.

119. http://www.leica-geosystems.com/en/Leica-GNSS-Spider_83498.htm
120. <http://www.systemnet.com.ua/ru/network/preimushchestva>
121. <http://gpsinformation.net/exe/waas.html>
122. http://www.esa.int/Our_Activities/Navigation/The_present_-_EGNOS/What_is_EGNOS
123. <http://www.gisa.ru/103059.html>
124. <http://www.novatel.com/about-us/news-releases/news>
125. <http://www.novatel.com/products/span-gnss-inertial-systems/mems-interface-card/>
126. <http://www.novatel.com/products/span-gnss-inertial-systems/span-combined-systems/span>
127. <http://www.kharkovgnssgroup.net/>
128. Ariane Hellinger, Ariane Hellinger, Heinrich Seeger. Cyber-Physical Systems. Driving force for innovation in mobility, health, energy and production. – Acatech. National Academy of Science and Engineering. – 2011. – 48p.
129. Alessandro Bazzi, Alberto Zanella, Barbara M. Masini, Gianni Pasolini. A Distributed Algorithm for Virtual Traffic Lights with IEEE 802.11p. // IEEE European Conference Networks and Communications (EuCNC). – 2014. – P. 1 – 5.
130. Ferreira, M., d'Orey, P.M. On the Impact of Virtual Traffic Lights on Carbon Emissions Mitigation // IEEE Transactions on Intelligent Transportation Systems. – 2012. – Volume 13, Issue 1. – P. 284 – 295.
131. Conceicao H., Ferreira M., Steenkiste P. Virtual traffic lights in partial deployment scenarios // IEEE Intelligent Vehicles Symposium (IV). – 2013. – P. 988 – 993.
132. Hahanov Vladimir, Wajeb Gharibi, Abramova L.S., Chumachenko Svetlana, Litvinova Eugenia; Hahanova Anna, Rustinov Vladimir, Miz Vladimir, Zhalilo Aleksey, Ziarmant Artur. Cyber physical system – smart cloud traffic control // Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2014). 2014. Pages: 1 - 18, DOI: 10.1109/EWDTS.2014.7027107

133. Hahanov, V., Gharibi, W., Man, K.L., Litvinova, E., Chumachenko, S., Guz, O. Intelligent road control and monitoring // Lecture Notes in Electrical Engineering. Volume 235 LNEE, 2013, Pages 327-335 2013 International Conference on Future Information and Communication Engineering, ICFICE 2013; Shenyang; China; 24 June 2013 - 26 June 2013; Code 98013.

134. Gorobets A., Chugurov I., Scherbin D., Chumachenko S. Dijkstra Algorithm for cyber structures analysis // Published in: Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), 2013. 12th International Conference on the 19-23 Feb. 2013 (Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого 2013. Львів – Поляна). – С.61-65.

135. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. – 1959. – Vol. 1. – P. 269-271.

136. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. Алгоритмы: построение и анализ = Introduction to Algorithms. М.: Вильямс, 2006. – 1296 с.

137. Левитин А. В. Глава 9. Жадные методы: Алгоритм Дейкстры // Алгоритмы: введение в разработку и анализ = Introduction to The Design and Analysis of Algorithms. – М.: Вильямс, 2006. – С. 189-195.

138. Кузнецов Н.А., Фетисов В.Н. Алгоритм Дейкстры с улучшенной робастностью для управления маршрутизацией в IP-сетях // Автоматика и телемеханика. – 2008. – № 2. – С. 80–85.

139. Томас Т.М. Структура и реализация сетей на основе протокола OSPF. Руководство Cisco = OSPF Network Design Solutions. – М.: Вильямс, 2004. – 816 с.

140. Alessandro Bazzi, Alberto Zanella, Barbara M. Masini, Gianni Pasolini. A Distributed Algorithm for Virtual Traffic Lights with IEEE 802.11p. Networks and Communications (EuCNC), IEEE European Conference. – 2014. – P. 1 – 5.

141. Ferreira, M., d'Orey, P.M. On the Impact of Virtual Traffic Lights on Carbon Emissions Mitigation // IEEE Transactions on Intelligent Transportation Systems,. – Vol. 13, Issue 1. – 2012. – P. 284 – 295.

142. Немченко В.П. Моделирование сетевых протоколов при построении тестовых последовательностей / В.П. Немченко, А.Н. Зиарманд, Ю.А. Чепелев // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2011. – №5(90). – С. 18-21. (Індексується GoogleScholar, Реферативна база “Наукова періодика України”, РИНЦ (eLibrary), National Library of Ukraine named after Vernadsky).

143. Немченко В.П. Использование энергосберегающих технологий в современных сетях / В.П. Немченко, А.Н. Зиарманд, А.С. Изотов // Вестник Херсонского национального технического университета. – 2012. – №1(44). – С. 146-148. (Індексується РИНЦ (eLibrary), Google Scholar, National Library of Ukraine named after Vernadsky).

144. Зиарманд А.Н. Модели и методы мониторинга и управления транспортом / А.Н. Зиарманд, В.И. Хаханов // Радиоэлектроника и информатика. – 2016. – №3(74). – С. 71-87. (Входить до міжнародних наукометричних баз Index Copernicus, Google Scholar, OECSP, OAJI, Scholar Steer, SIS, Cyberleninka, CiteFactor, TIU Hannover, I2OR).

145. Vladimir Hahanov. Internet-driven Cyber Control of Traffic / Vladimir Hahanov, Wajeb Gharibi, Svetlana Chumachenko, Evgeniya Litvinova, Vladimir Miz, Arthur Ziarmand // Australian Journal of Scientific Research. – 2014. – Volume IV, No.1(5). – Pp. 217-224.

146. Hahanov V. Cloud-Driven Traffic Monitoring and Control Based on Smart Virtual Infrastructure / V. Hahanov, Wajeb Gharibi, E. Litvinova, S. Chumachenko, A. Ziarmand, I. Englesi, I. Gritsuk, V. Volkov, A. Khakhanova // SAE Technical Paper. USA. 2017-01-0092, 2017. 6 p. doi:10.4271/2017-01-0092. (Входить до міжнародної наукометричної бази Scopus, ORCID).

147. Зиарманд А.Н. Модель облачного сервиса для поиска оптимального пути / С. В. Чумаченко, Е. И. Литвинова, В. И. Хаханов, А. Н. Зиарманд //

Paradigmata poznání. Vědecko vydavatelské centrum «Sociosféra-CZ», s.r.o., Praha, Česká republika. – 2017. – Вып. 3. – С.63-83. doi: 10.24045/pp.2017.3.6 (The journal is indexed by Electronic Research Library, Russia; Research Bible, China; Scientific Indexing Services, USA; Cite Factor, Canada; General Impact Factor, India; Scientific Journal Impact Factor, India; CrossRef, USA; ORCID, USA).

148. *Ziarmand A.H.* Теоретическая суть проекта «Smart Roads» / *A.H. Ziarmand* // Автоматизированные системы управления и приборы автоматики. – 2013. – Вып. 162. – С. 28-34. (Входить до міжнародних наукометричних баз Google Scholar, Cyberleninka).

149. Hahanov V. I. Cloud traffic control system / V. I. Hahanov, O. A. Gus, A. Ziarmand, Ngene Christopher Umerah, A. Arefjev // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 72-76. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

150. *Ziarmand A.* Smart road infrastructure / *A. Ziarmand* // Proc. Of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 430-434. (Work in Progress). (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

151. *Ziarmand A.* Cloud Service for Traffic Control / *Artur Ziarmand, Vladimir Hahanov, Volodymyr Miz, Anastasya Hahanova, Aleksey Priymak* // Proc. of the XII International IEEE Conference “Modern Problems of Radio Engineering, Telecommunications, and Computer Science”. 2014. – Lviv-Slavske, Ukraine. – P. 557-559. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

152. Hahanov V. Cyber physical system – smart cloud traffic control / V. Hahanov, Wajeb Gharibi, L.S. Abramova, S. Chumachenko, E. Litvinova, A. Hahanova, V. Rustinov, V. Miz, A. Zhalilo, A. Ziarmand // Proc. of the IEEE Design & Test Symposium. – 2014. – Kiev, Ukraine. – P. 49-66. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

153. *Ziarmand A.* Transport monitoring and control systems / *A. Ziarmand, D. Kucherenko, T. Soklakova* // Proc. of the IEEE East-West Design & Test

Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 474-477. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

154. *Зиарманд А.Н.* Формальная модель киберсистемы / А.Н. Зиарманд, И.И. Чугуров // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2014. – С. 65.

155. *Зиарманд А.Н.* Формальна модель хмарного управління транспортом / А.Н. Зиарманд, К.Э. Мороз // Материалы XIX Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2015. – С. 4.

156. *Зиарманд А.Н.* I-Cloud Traffic Control / А.Н. Зиарманд, В.А. Мизь // 6-я Международная студенческая конференция и конкурс научных работ по вопросам информационной безопасности «CyberSecurity for the Next Generation». – 2014. – “Kaspersky Office”, Москва, РФ. – С. 13.

157. Chumachenko S. Quantum data structures for SoC design / S. Chumachenko, A. Shkil, A. Hahanova, A. Ziarmand, A. Pryimak // Proc. of the 13th International Conference “IEEE Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”. – 2015. Polyana-Lviv. – Ukraine. – P. 355-357. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

158. Soklakova T. Big data visualization in smart cyber university / T. Soklakova, A. Ziarmand, S. Osadchyieva // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 469-473. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

159. *Зиарманд А.Н.* Анализ компетентности пользователя компьютерной системы как диагностический эксперимент / Зиарманд А.Н., Кучеренко Д.Е. // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2011. – С. 40-41.

ДОДАТОК А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Немченко В.П. Моделирование сетевых протоколов при построении тестовых последовательностей / В.П. Немченко, А.Н. Зиарманд, Ю.А. Чепелев // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2011. – №5(90). – С. 18-21. (Індексується GoogleScholar, Реферативна база “Наукова періодика України”, РИНЦ (eLibrary), National Library of Ukraine named after Vernadsky).
2. Немченко В.П. Использование энергосберегающих технологий в современных сетях / В.П. Немченко, А.Н. Зиарманд, А.С. Изотов // Вестник Херсонского национального технического университета. – 2012. – №1(44). – С. 146-148. (Індексується РИНЦ (eLibrary), Google Scholar, National Library of Ukraine named after Vernadsky).
3. *Зиарманд А.Н.* Модели и методы мониторинга и управления транспортом / *А.Н. Зиарманд, В.И. Хаханов* // Радиоелектроника и информатика. – 2016. – №3(74). – С. 71-87. (Входить до міжнародних наукометричних баз Index Copernicus, Google Scholar, OECSP, OAJI, Scholar Steer, SIS, Cyberleninka, CiteFactor, TIU Hannover, I2OR).
4. Vladimir Hahanov. Internet-driven Cyber Control of Traffic / Vladimir Hahanov, Wajeb Gharibi, Svetlana Chumachenko, Evgeniya Litvinova, Vladimir Miz, *Arthur Ziarmand* // Australian Journal of Scientific Research. – 2014. – Volume IV, No. 1(5). – Pp. 217-224.
5. Hahanov V. Cloud-Driven Traffic Monitoring and Control Based on Smart Virtual Infrastructure / V. Hahanov, Wajeb Gharibi, E. Litvinova, S. Chumachenko, *A. Ziarmand*, I. Englesi, I. Gritsuk, V. Volkov, A. Khakhanova // SAE Technical Paper. USA. 2017-01-0092, 2017. 6 p. doi: 10.4271/2017-01-0092. (Входить до міжнародної наукометричної бази Scopus, ORCID).
6. *Зиарманд А.Н.* Модель облачного сервиса для поиска оптимального пути / С. В. Чумаченко, Е. И. Литвинова, В. И. Хаханов, *А. Н. Зиарманд* // Paradigmata poznání. Vědecko vydavatelské centrum «Sociosféra-CZ», s.r.o.,

Praha, Česká republika. – 2017. – Вып. 3. – С.63-83. doi: 10.24045/pp.2017.3.6 (The journal is indexed by Electronic Research Library, Russia; Research Bible, China; Scientific Indexing Services, USA; Cite Factor, Canada; General Impact Factor, India; Scientific Journal Impact Factor, India; CrossRef, USA; ORCID, USA).

7. *Ziarmand A.H.* Теоретическая суть проекта «Smart Roads» / *A.H. Ziarmand* // Автоматизированные системы управления и приборы автоматики. – 2013. – Вып. 162. – С. 28-34. (Входит до міжнародних наукометричних баз Google Scholar, Cyberleninka).

8. Hahanov V. I. Cloud traffic control system / V. I. Hahanov, O. A. Gus, A. Ziarmand, Ngene Christopher Umerah, A. Arefjev // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 72-76. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

9. *Ziarmand A.* Smart road infrastructure / *A. Ziarmand* // Proc. Of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 430-434. (Work in Progress). (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

10. *Ziarmand A.* Cloud Service for Traffic Control / *Artur Ziarmand, Vladimir Hahanov, Volodymyr Miz, Anastasya Hahanova, Aleksey Priymak* // Proc. of the XII International IEEE Conference “Modern Problems of Radio Engineering, Telecommunications, and Computer Science”. 2014. – Lviv-Slavske, Ukraine. – P. 557-559. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

11. Hahanov V. Cyber physical system – smart cloud traffic control / V. Hahanov, Wajeb Gharibi, L.S. Abramova, S. Chumachenko, E. Litvinova, A. Hahanova, V. Rustinov, V. Miz, A. Zhalilo, A. Ziarmand // Proc. of the IEEE Design & Test Symposium. – 2014. – Kiev, Ukraine. – P. 49-66. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

12. *Ziarmand A.* Transport monitoring and control systems / *A. Ziarmand, D. Kucherenko, T. Soklakova* // Proc. of the IEEE East-West Design & Test

Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 474-477. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

13. *Зиарманд А.Н.* Формальная модель киберсистемы / А.Н. Зиарманд, И.И. Чугуров // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2014. – С. 65.

14. *Зиарманд А.Н.* Формальна модель хмарного управління транспортом / А.Н. Зиарманд, К.Э. Мороз // Материалы XIX Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2015. – С. 4.

15. *Зиарманд А.Н.* I-Cloud Traffic Control / А.Н. Зиарманд, В.А. Мизь // 6-я Международная студенческая конференция и конкурс научных работ по вопросам информационной безопасности «CyberSecurity for the Next Generation». – 2014. – “Kaspersky Office”, Москва, РФ. – С. 13.

16. Chumachenko S. Quantum data structures for SoC design / S. Chumachenko, A. Shkil, A. Hahanova, A. Ziarmand, A. Pryimak // Proc. of the 13th International Conference “IEEE Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”. – 2015. Polyana-Lviv. – Ukraine. – P. 355-357. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

17. Soklakova T. Big data visualization in smart cyber university / T. Soklakova, A. Ziarmand, S. Osadchyieva // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 469-473. (Входит до міжнародних наукометричних баз Scopus, IEEE Xplore).

18. *Зиарманд А.Н.* Анализ компетентности пользователя компьютерной системы как диагностический эксперимент / Зиарманд А.Н., Кучеренко Д.Е. // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2011. – С. 40-41.

ДОДАТОК Б

АПРОБАЦІЯ РЕЗУЛЬТАТІВ

1. Hahanov V. I. Cloud traffic control system / V. I. Hahanov, O. A. Gus, A. Ziarmand, Ngene Christopher Umerah, A. Arefjev // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 72-76. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

2. Ziarmand A. Smart road infrastructure / A. Ziarmand // Proc. Of the IEEE East-West Design & Test Symposium (EWDTS). – Rostov-on-Don, Russia. – 2013. – P. 430-434. (Work in Progress). (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

3. Ziarmand A. Cloud Service for Traffic Control / Artur Ziarmand, Vladimir Hahanov, Volodymyr Miz, Anastasya Hahanova, Aleksey Priymak // Proc. of the XII International IEEE Conference “Modern Problems of Radio Engineering, Telecommunications, and Computer Science”. 2014. – Lviv-Slavske, Ukraine. – P. 557-559. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

4. Hahanov V. Cyber physical system – smart cloud traffic control / V. Hahanov, Wajeb Gharibi, L.S. Abramova, S. Chumachenko, E. Litvinova, A. Hahanova, V. Rustinov, V. Miz, A. Zhalilo, A. Ziarmand // Proc. of the IEEE Design & Test Symposium. – 2014. – Kiev, Ukraine. – P. 49-66. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

5. Ziarmand A. Transport monitoring and control systems / A. Ziarmand, D. Kucherenko, T. Soklakova // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 474-477. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

6. Зиарманд А.Н. Формальная модель киберсистемы / А.Н. Зиарманд, И.И. Чугуров // Материалы XVIII Международного молодежного форума

«Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2014. – С. 65. – *очна участь з доповіддю.*

7. *Зиарманд А.Н.* Формальна модель хмарного управління транспортом / *А.Н. Зиарманд, К.Э. Мороз* // Матеріали XIX Міжнародного молодіжного форуму «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2015. – С. 4. – *очна участь з доповіддю.*

8. *Зиарманд А.Н.* I-Cloud Traffic Control / *А.Н. Зиарманд, В.А. Мизь* // 6-я Міжнародна студентська конференція і конкурс наукових робіт по питанням інформаційної безпеки «CyberSecurity for the Next Generation». – 2014. – “Kaspersky Office”, Москва, РФ. – *очна участь з доповіддю.*

9. Chumachenko S. Quantum data structures for SoC design / S. Chumachenko, A. Shkil, A. Hahanova, A. Ziarmand, A. Pryimak // Proc. of the 13th International Conference “IEEE Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”. – 2015. Polyana-Lviv. – Ukraine. – P. 355-357. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

10. Soklakova T. Big data visualization in smart cyber university / T. Soklakova, A. Ziarmand, S. Osadchyieva // Proc. of the IEEE East-West Design & Test Symposium (EWDTS). – Yerevan, Armenia. – 2016. – P. 469-473. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore). – *очна участь з доповіддю.*

11. *Зиарманд А.Н.* Анализ компетентности пользователя компьютерной системы как диагностический эксперимент / *Зиарманд А.Н., Кучеренко Д.Е.* // Матеріали XV Міжнародного молодіжного форуму «Радиоэлектроника и молодежь в XXI веке». – Харьков, Украина. – 2011. – С. 40-41. – *очна участь з доповіддю.*

ДОДАТОК В

**ДОКУМЕНТИ, ЩО ПІДТВЕРДЖУЮТЬ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ
ДИСЕРТАЦІЇ**

«ЗАТВЕРДЖУЮ»
Проректор з НМР ХНУРЕ
Рубан І.В.
2017 р.

АКТ

про впровадження у навчальний процес ХНУРЕ результатів дисертаційної роботи Зіарманда Артура Нісаровича «Моделі і методи кіберфізичного комп'ютингу для цифрового моніторингу та хмарного управління транспортом», представленої на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти

Комісія у складі: зав. каф. АПОТ проф. Чумаченко С.В., доц. каф. АПОТ Шкіля О.С., проф. каф. АПОТ Литвинова Є.І. розглянула матеріали дисертаційної роботи Зіарманда А.Н., які використовуються в навчальному процесі кафедри АПОТ ХНУРЕ у 2016/2017 навчальному році, і прийшла до наступного висновку.

Розроблені у дисертаційній роботі методи і моделі кіберфізичної інфраструктури, а саме:

– удосконалений алгоритм Дейкстри, який відрізняється можливістю аналізу кіберфізичної інфраструктури дорожнього руху для online пошуку квазіоптимального маршруту транспортного засобу в умовах виникнення колізій;

– удосконалені метрика і критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання;

впроваджені у навчальний процес кафедри АПОТ ХНУРЕ та використовуються у таких навчальних дисциплінах:

1. У навчальній дисципліні «Дискретна математика» для бакалаврів напрямку «Комп'ютерна інженерія» у лекційному матеріалі та у практичних заняттях за темою «Алгоритм Дейкстри».

2. У навчальній дисципліні «Cloud-Fog кіберфізичні системи» для магістрантів спеціалізацій «Спеціалізовані комп'ютерні системи», «Проектування вбудованих мікросистем» у лекційному матеріалі за темами «Аналіз кіберфізичного простору», «Internet of Things. Основні поняття IoT як складової кіберфізичних систем. Інтернет речей і розумні сенсори. Фізичний Binary Computing», «Computing sustainable development matrix. Метрика для кіберфізичного комп'ютингу. Структури секвенсорів для прийняття рішень», «Трикутна топологія кіберпростору. Оцінка якості кіберфізичної топології», а також у лабораторному практикумі за темою «Синтез логічної схеми управління світлофором».

 Зав. каф. АПОТ проф. Чумаченко С.В.,

 Доц. каф. АПОТ Шкіля О.С.,

 Проф. каф. АПОТ Литвинова Є.І.



ТОВ «Перший інститут надійного програмного забезпечення»

ДОВІДКА

про можливість впровадження методів та моделей кіберфізичної транспортної хмарної інфраструктури

Розроблена на кафедрі Автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки за участю (30%) здобувача Зіарманда Артура Нісаровіча архітектура розумного хмарного світлофора характеризується використанням логічних операцій і часом простою зеленого сигналу, що дає можливість на 10-30% зменшити час простою зеленого світлофора і істотно збільшити пропускну здатність транспортних потоків на перехресті доріг. Впровадження розумних світлофорів в масштабах міста, дозволить скоротити інтегральний час знаходження транспорту на маршруті і витрати на енергоносії не менше, ніж на 10%.

У рамках практичної реалізації моделей і методів цифрового моніторингу і хмарного керування автомобілем з метою створення і верифікації програмних компонентів кіберфізичної архітектури дорожнього руху «Cloud Traffic Control» виконано тестування модельних потоків транспортних засобів на ділянках дорожньої інфраструктури. Достовірність результатів підтверджується проведеними експериментами, тестуванням і верифікацією запропонованих моделей і методів моніторингу та управління дорожнім рухом.

Вперше запропоновано модель транспортного комп'ютингу, яка характеризується кіберфізичною взаємодією автомобіля з хмарним сервісом за допомогою еволюційного переміщення світлофора з фізичного у віртуальний простір для цифрового моніторингу транспортних потоків і квазіоптимального управління дорожнім рухом.

Розроблені методи і моделі кіберфізичної інфраструктури:

– удосконалений алгоритм Дейкстри, який відрізняється можливістю аналізу кіберфізичної інфраструктури дорожнього руху для online пошуку квазіоптимального маршруту транспортного засобу в умовах виникнення колізій;

– удосконалені метрика і критерії оцінювання якості інфраструктури, яка відрізняється можливістю online аналізу кіберфізичного простору для пошуку квазі-оптимального маршруту і зменшення часу його виконання;

можуть бути імплементовані як компонент проекту при синтезі хмарного сервісу.

28.09.2017

Директор /Чертков П.М./

Голова комісії:

/Чертков П.М./



ДОДАТОК Г

ПЕРЕВАГИ ХМАРНИХ СЕРВІСІВ КІБЕРСИСТЕМИ

Впровадження СТС надасть можливість суттєво поліпшити комфортність пересування водіїв і пасажирів, зберегти сотні мільярдів доларів, матеріалів і часу, а також зберегти екологію планети для майбутніх поколінь. Переваги впровадження точного цифрового моніторингу і хмарного управління транспортом мають глобальний і локальний характер і полягають у наступному:

1) Збереження екології планети і регіонів за рахунок зменшення забруднення навколишнього середовища, підвищення якості життя людей за кермом, економія паливних ресурсів і скорочення часу руху завдяки вибору оптимального маршруту, зменшення кількості і складності трафіку за рахунок розумної інфраструктури та інтелектуальних світлофорів.

2) Повне виключення дорожньої поліції для ідентифікації автомобілів, які порушують правила дорожнього руху, точний моніторинг позиціонування транспортних засобів в часі і просторі, включаючи викрадення, колізії, несанкціоновані маршрути. Дистанційне керування через хмарні сервіси екстреним вимиканням двигуна автомобіля, при створенні реальної небезпеки іншим учасникам дорожнього руху. Істотне зниження аварійності за рахунок моніторингу та прорахунку рівня безпеки маневрів, зменшення наслідків дорожньо-транспортних пригод, підвищення безпеки і комфорту учасників дорожнього руху.

3) Моніторинг позиціонування і пересування транспортних засобів в автомобільних компаніях, оптимальне виконання замовлень з перевезення пасажирів і вантажів з позиції мінімізації матеріальних і / або часових витрат.

4) Надання сервісів водієві, пов'язаних з прокладанням оптимальних маршрутів і графіка руху з урахуванням негативних дорожніх факторів в існуючій інфраструктурі для мінімізації матеріальних і часових витрат в online режимі; істотне зниження аварійності за рахунок цифрового моніторингу

дорожньої обстановки, закритою для візуального перегляду ділянок руху і прорахунку рівня безпеки маневрів.

5) Надання сервісів пасажиром з моніторингу позиціонування і руху пасажирських транспортних засобів на зупиночних або транспортних терміналах завдяки використанню стаціонарних моніторів або мобільних гаджетів, пов'язаних з відповідними хмарними сервісами. Візуалізація на моніторі автомобіля критичних точок маршруту руху транспортного засобу в реальному часі шляхом використання камер відеоспостереження. Замовлення гібридного маршруту руху до пункту призначення шляхом використання різних транспортних засобів.

6) Автоматична і точна цифрова реєстрація в часі і в просторі всіх порушень правил дорожнього руху та дорожньо-транспортних пригод, які тягнуть зняття з рахунку грошового еквівалента штрафів і формують історію водія в страховій компанії, що дає можливість виключити участь поліції в процедурах огляду ДТП.

Д.1 Технічні і функціональні можливості TCS

1) Моніторинг (24/7) реальної швидкості руху всіх транспортних засобів та інформування про зони швидкісного режиму. Цифровий моніторинг проїзду на заборонні знаки і сигнали світлофорів.

2) Економія палива, зменшення забруднення навколишнього середовища, скорочення часу руху, завдяки вибору оптимального маршруту, наданого хмарою.

3) Зменшення кількості та складності пробок за рахунок планування руху транспортних засобів, що враховує майбутні маршрути учасників. Коригування маршруту руху транспортного засобу в реальному часі при зміні дорожньої ситуації.

4) Інтелектуальне управління циклами перемикання світлофорів залежно від дорожньої обстановки на перехрестях.

5) Генерування аналітичних, статистичних звітів та рекомендацій щодо поліпшення інфраструктури доріг, розстановки віртуальних знаків, світлофорів та централізоване програмування циклів їх перемикання.

6) Протидія угону транспортного засобу і самовільне залишення місця ДТП, завдяки моніторингу місця розташування кожного автомобіля. Хмарна цифрова реєстрація для страхових компаній всіх необхідних деталей і динаміки ДТП, не пов'язаних з травмами, без участі дорожньої поліції.

7) Інформування за допомогою віртуальної тривожної кнопки спеціальних служб про події, що сталися на дорогах або з автомобілем.

8) Попередження водія про потенційну небезпеку на замовленому маршруті, інформація про яку отримана від хмари в процесі руху.

Д.2 Проблеми, які вирішуються за допомогою цифрової ідентифікації

Уже сьогодні існує досить багато актуальних і практично орієнтованих задач, які можна вирішити за допомогою радіоцифрових паспортів:

1. Ідентифікація виробу (об'єкта або суб'єкта) в локальній або глобальній системі координат.
2. Збереження параметрів, що характеризують основні властивості об'єкта.
3. Накопичення і збереження історії життєвого циклу об'єкта.
4. Передача інформації про об'єкт або явище за санкціонованою вимогою в хмару управління.
5. Прийом санкціонованої інформації, що дає можливість модифікувати окремі властивості електронного паспорта об'єкта.
6. Санкціонована взаємодія з електронними паспортами (гаджетами) інших об'єктів, що знаходяться в полі радіовідимості об'єкта.
7. Передача інформації про взаємодію об'єкта з іншими ідентифікаторами в межах радіовідимості.

Таким чином, електронний цифровий ідентифікатор об'єкта є автономною цифровою системою на кристалі з прийомо-передавачем, який

здатний зберігати інформацію про ідентифікований об'єкт, модифікувати її за командою центру управління, а також зберігати інформацію про взаємодії з навколишнім середовищем і іншими цифровими ідентифікаторами з можливістю передачі даних в хмару управління. Інші варіанти ID комунікацій пов'язані з використанням: 1) мережі мобільної телефонії; 2) супутникових систем для прийому і передачі інформації.

Д.3 Аргументи проти впровадження хмари в масштабах країни

1. «Порушення права на недоторканність приватного життя, оскільки теоретично TCS хмара здійснює тотальний моніторинг всіх транспортних засобів». Насправді сьогодні існує система законного перехоплення телекомунікацій, реалізована відповідно до міжнародних вимог. Перехоплення телефонних переговорів абонента використовується тільки в ході слідства і з санкції суду. Є можливість відстежувати місце знаходження абонента спеціальними службами. Даний факт для законослухняних громадян ніякої проблеми не створює, якщо служби укомплектовані чесними чиновниками.

2. «Додаткові витрати на придбання апаратно-програмних мобільних гаджетів ідентифікації і орендна плата, близько 100 доларів в рік, за використання сервісів TCS-системи». Уже сьогодні практично всі мешканці планети мають такі пристрої, а економічні переваги хмари, пов'язані з економією палива і зменшенням часу поїздки цілком гідно компенсують витрати на придбання сервісів.

3. «Ненадійність віртуальних світлофорів, пов'язана з падінням хмарного сервісу». Однак, реальні світлофори більш уразливі і менш надійні, ніж віртуальні, з причин відключення електроенергії, фізичних порушень через аварії, відмов компонентів світлофорів. Віртуальні світлофори знищити практично неможливо.

Д4 Практичні приклади впровадження компонентів TCS-системи

1. Програмний додаток управління корпоративними перевезеннями використовується для оптимального планування рейсів з доставки вантажів, що приводить до зменшення часових і матеріальних витрат за рахунок: 1) зниження витрат на паливно-мастильні матеріали; 2) оптимального розподілу замовлень між автомобілями; 3) прогнозування поставок товарів для зменшення складських витрат; 4) економії робочого часу персоналу або скорочення штатних співробітників; 5) зменшення числа автомобілів для виконання заданого обсягу перевезень; 6) моніторингу та оперативного управління автомобілями при доставці вантажів в реальному часі. Ринкова привабливість хмарного сервісу транспортної логістики: оптові компанії, регіональні дистриб'ютори продовольчих і промислових товарів (хлібозаводи, молокозаводи, м'ясокомбінати, пиво-безалкогольні комбінати, промислові підприємства, автотранспортні підприємства, торгові мережі, логістичні оператори, транспортно-експедиторські компанії, вендингові компанії, швидка допомога, інкасаторські служби, кур'єрські служби, інтернет-магазини, клінінгові компанії).

2. Дистанційний модуль "SHERLOCK" призначений для побудови розподілених систем моніторингу та управління об'єктами, включаючи мобільні. Являє собою електронний виріб, побудований на основі використання трьох новітніх технологій Mobile-to-Mobile, GPS і GPRS. Задачі, які вирішуються за допомогою модуля: 1) автоматичне визначення місцезнаходження транспортних засобів; 2) управління автомобілями транспортного парку, логістика; 3) автоматизація служб таксі; 4) моніторинг маршруту і розкладу руху транспорту; 5) моніторинг режимів експлуатації транспортних засобів. Технічні характеристики: GPS – багатоканальний приймач з високою чутливістю і малим енергоспоживанням, спеціально призначений для роботи в умовах міської забудови та наявності відбитих сигналів <http://gps.rfid.com.ua>. Для отримання доступу до сервісу користувач повинен авторизуватися за допомогою логіна і пароля. На головній сторінці

сервісу більшу частину площі екрану займає вікно з картою, на яку накладаються дані про місцезнаходження мобільних об'єктів. Для візуалізації використовується картографічна інформація компанії ВІЗІКОМ. При візуалізації на карті піктограмами відображається стан об'єктів і маршрут руху за обраний інтервал часу, а також тривалі стоянки. Розмір і положення карти можна змінювати за допомогою миші і елементів управління. У нижній частині головної сторінки розташовуються елементи управління, що дозволяють швидко перемикатися між частинами маршруту і між об'єктами, а також статистична інформація. При виборі відображення тільки одного об'єкта доступна функція розрахунку відстані. Комплект поставки телеметричного модуля: антена GPS; антена GSM; кабель з'єднувальний; інструкція з експлуатації; SIM-карта.

Д.5 Технології високоточної ГНСС-навігації транспортних засобів

Навігація – визначення місця розташування об'єкта, що рухається в заданій системі координат, а також вектора швидкості і кутової орієнтації з точною прив'язкою часу до навігаційних параметрах. Оскільки функції кіберсистеми – не тільки моніторинг транспортних засобів, а й управління їх рухом, то вимоги до точності і надійності навігаційних операцій повинні задовольняти самим сучасним і перспективним вимогам державних радіонавігаційних планів, що пред'являються до критичних об'єктів управління в типових і в складних дорожніх умовах. При цьому для управління використовуються сигнали глобальних навігаційних супутникових систем (ГНСС) – GPS (США), ГЛОНАСС (РФ), BeiDou / Compass (Китай), Galileo (ЄС), також їх регіональних функціональних доповнень (РФД) - WAAS (США), EGNOS (ЄС), MSAS (Японія), QZSS (Японія), GAGAN (Індія). Як показує аналіз, основні вектори розвитку навігаційного ринку в світі визначаються застосуванням супутникових навігаційних технологій на транспорті, в першу чергу автомобільному. Хорошим прикладом використання супутникових технологій слугує проект екстреного реагування при аваріях на дорогах «ЕРА-ГЛОНАСС» (РФ). Перспективи навігаційних

супутникових технологій – у високоточній навігації транспорту, які стануть масово і матеріально доступними через 5 років завдяки диференційним технологіям RTK (Real Time Kinematic), PPP-RTK визначення траєкторії руху об'єкта з сантиметровою (міліметровою) точністю. Метод, який використовує широкозонні корекції регіональних функціональних доповнень – WAAS (США), EGNOS (ЄС), MSAS (Японія), QZSS (Японія), GAGAN (Індія) формує WADGNSS-системи (Wide Area Differential Systems). Тут виконується трансляція із супутників диференційних поправок, що дозволяє підвищити точність визначення планових координат до 1,0 м. Традиційний диференційний метод корекції похибок ГНСС-вимірювань шляхом використання диференційних корекцій від окремих ГНСС-станцій дозволяє збільшити точність визначення планових координат до рівня 0,5-0,8 метрів з меншою надійністю навігаційних визначень в порівнянні з методом широкозонної навігації.

Бортове навігаційне обладнання транспортного засобу – радіонавігаційні приймач і радіомодем має забезпечувати: 1) визначення координат об'єкту з точністю $0,5 \div 1,0$ м в плані і $1,0 \div 2,0$ м за висотою; 2) визначення складових вектора швидкості з точністю $0,05 \div 0,10$ м/с; 3) визначення параметрів кутової орієнтації об'єкта з точністю $0,15 \div 0,30$ кут. град. за курсом, або $0,5 \div 1,0$ кут. град. по тангажу і / або крену; 4) визначення відходу бортової шкали часу з точністю $50 \div 100$ нс. При управлінні рухомими об'єктами бортове навігаційне обладнання повинне забезпечити визначення координат динамічних об'єктів з точністю 0,05 м (в плані) і 0,1 м (по висоті). Зазначені вище значення точностних характеристик (рівні допустимих похибок) навігаційних визначень відповідають 95%-му вірогідністному довірливому інтервалу. Вимоги до надійності навігаційних визначень: 1) доступність або експлуатаційна готовність (Availability) – з ймовірністю 0,999; 2) цілісність (Integrity) включає «рівень тривоги» (Alert Limit) – 2 м і інтервал часу до повідомлення про вихід характеристик за допустимі межі (Time to Alarm) – 5 секунд; 3) безперервність обслуговування (Continuity) – з

ймовірністю 0,999/год. У разі необхідності отримання даних про параметри кутової орієнтації об'єктів можливо використовувати ГНСС-технологію визначення кутових параметрів спільно з інерційної підсистемою, наприклад, одного зі світових лідерів – компанії NovAtel. Реалізація технології RTK вимагає використання телекомунікаційних засобів (мобільний Інтернет, GSM/GPRS) для отримання диференційних корекцій від мережі референтних станцій. Така мережа вже існує і на Україні та забезпечує практично повне покриття її території. Навігаційна технологія WADGNSS використовує прийом поправок споживачем як від геостаціонарних супутників-ретрансляторів корекцій на частоті L1 GPS, так і по мережі Інтернет в умовах міських забудов і висотних будівель при маскуванні сигналів від супутників-ретрансляторів корекцій. У разі неможливості отримати RTK-корекції, наприклад, при перервах зв'язку з мережею референтних ГНСС-станцій або при виїзді об'єкта із зони дії найближчій ГНСС RTK-мережі, приймач споживача переходить в режим роботи WADGNSS, отримуючи сигнали корекції безпосередньо від геостаціонарних супутників на приймальню ГНСС-антену. Приймаються одночасно як сигнали ГНСС, так і коригувальні сигнали від супутників-ретрансляторів. При цьому точність навігаційних визначень зменшується з кількох сантиметрів до 1 метра за плановими координатами.

Інтегрування ГНСС + INS дозволяє виключити втрати навігаційної інформації там, де відсутня стеження за сигналами навігаційних супутників: під мостами, в тунелях, в умовах міських «каньйонів» і «допомогти» ГНСС-приймачу здійснити миттєве відновлення стеження за навігаційними сигналами ГНСС. Датчик ГНСС, в свою чергу, забезпечує отримання початкових умов (поточні координати і складові вектора швидкості об'єкта) для роботи інерційної підсистеми. Крім того, інтегрування ГНСС + INS в разі підтримки RTK-визначень дозволяє забезпечити необхідні характеристики (Availability, Integrity, Continuity) надійності навігаційних визначень. На сучасному етапі окремо RTK-технологія не забезпечує необхідної надійності

навігаційних визначень для задач кіберуправління. Інерціальні пристрої INS можуть включати в себе системи датчиків-акселерометрів і датчиків-гіроскопів в різних конфігураціях для визначення місця розташування та параметрів кутової орієнтації. В останні роки для транспортних додатків використовуються недорогі інерціальні мікромеханічні датчики MEMS, які інтегруються з ГНСС OEM-модулями.

Таким чином, найкращим варіантом побудови бортової підсистеми навігаційних визначень для моніторингу та управління транспортом є: 1) інтегрування ГНСС + INS на основі технологій MEMS (координатні і, за необхідністю, кутові визначення); 2) мульти-системність – одночасний прийом сигналів всіх чотирьох ГНСС + коригувальних сигналів РФД (WADGNSS); 3) RTK / WADGNSS / DGNSD технології точного позиціонування / навігації; 4) здійсненність кутових визначень за сигналами ГНСС з інтеграцією з INS. Устаткування, що реалізує перераховані функції з заданим якістю, існує, його ціна становить \$ 15-20 тисяч. Через 5 років можна очікувати зниження ціни до рівня ~ \$ 1,0 тисячі при масовому випуску апаратури. Розробку прототипу бортової апаратури управління автомобілем і проведення експериментальних досліджень в рамках проекту можуть виконати співробітники НДІ «Супутникові мережеві технології високоточного позиціонування» ХНУРЕ, які мають багаторічний досвід з розробки супутникових технологій точного позиціонування і навігації: FP7 «EEGS – EGNOS Extension to Eastern Europe» – грант № 247698; «EEGS2 - EGNOS Extension to Eastern Europe Applications» – грант № 287179.

Д.6 Система управління корпоративними перевезеннями

Система вже використовується для оптимального планування рейсів з доставки вантажів, що приводить до зменшення часових і матеріальних витрат за рахунок: 1) зниження витрат на паливно-мастильні матеріали (ПММ); 2) оптимального розподілу замовлень між автомобілями; 3) прогнозування поставок товарів для зменшення складських витрат; 4) економії робочого часу персоналу або скорочення штатних співробітників; 5) зменшення числа

автомобілів для виконання заданого обсягу перевезень; б) моніторингу та оперативного управління автомобілями при доставці вантажів в реальному часі.

Ринкова привабливість хмарного сервісу транспортної логістики: оптові компанії, регіональні дистриб'ютори продовольчих і промислових товарів (хлібозаводи, молокозаводи, м'ясокомбінати, пиво-безалкогольні комбінати, промислові підприємства, автотранспортні підприємства, торгові мережі, логістичні оператори, транспортно-експедиторські компанії, вендингові компанії, швидка допомога, інкасаторські служби, кур'єрські служби, інтернет-магазини, клінінгові компанії). Більше 7500 підприємств тільки в Україні.

Технології логістики. Послуги з перевезення вантажів – це комплексна, багатокритеріальна задача, що включає велику кількість параметрів, від яких залежить ефективність виконання договору з замовником, а значить, і прибуток підприємства. Транспортна задача є NP-повною, де кількість варіантів знаходиться в експоненційній залежності від числа вхідних значень. Точне рішення може бути отримано методом повного перебору всіх можливих варіантів. Для реальних завдань бізнесу використовуються квазіоптимальні методи, які не дають точного рішення, а значить, і максимально можливої економії коштів. Пропонується оптимальний метод вирішення транспортної задачі на основі оригінального алгоритму, що істотно зменшує час, який стає прийнятним для аналізу більшості практичних ситуацій на картах регіонів.

Бізнес-моделі: 1) Продаж ліцензій на використання програмного продукту з подальшою оплатою сервісного супроводу. 2) Продаж сервісів відповідно до абонентської плати за час користування функціональностями хмари дорожнього руху.

Д.7 Організація зв'язків «хмара – автомобіль» і «хмара – інфраструктура»

Найважливішим аспектом технологічної (технічної) реалізації CPS є організація зв'язку між чотирма компонентами системи (рис. 6), інтегрованими з хмарою: Cloud Servers – сервери, що створюють хмару довготривалого зберігання розподілених даних і сервісів; Buffer Computers – буферні комп'ютери, що забезпечують збір даних від моніторів інфраструктури та доставку сервісів управління дорожнім контролерам; C-RFID – комп'ютерні блоки радіочастотної ідентифікації транспортних засобів; I-CMC – інфраструктурні контролери моніторингу та управління дорожнім рухом на основі радіочастотної ідентифікації транспортних засобів.

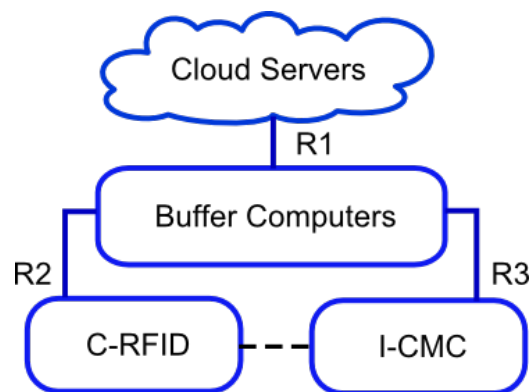


Рис. 6 – Структура зв'язків між компонентами ПДД

Структура комунікаційної інтеграції чотирьох компонентів ПДД представлена транзакціями: $(R1 * R2) = (SC, BC, C-RFID)$ – доставка хмарних сервісів до споживача; $(R1 * R3) = (SC, BC, I-CMC)$ – доставка сигналів управління до контролерів дорожнього руху. Маршрут першого типу використовує традиційні технології GPRS, HSPA, Wi-Fi, WiMAX на основі мережі Internet. Для другого типу транзакцій, зважаючи на їх надзвичайну важливість, а також високі вимоги до надійності, завадостійкості та захищеності, необхідні додаткові науково-технічні дослідження в процесі створення масштабованого прототипу.

Передбачається, що в блоці C-RFID будуть записані індивідуальний код транспортного засобу (CID), електронний код реєстрації за місцем

проживання (NID), а також код водія (DID), що виконує експлуатацію даного автомобіля в поточний момент часу. Зчитувати тріаду кодів (CND-ID) повинні радіопристрої, якими будуть оснащені всі світлофорні об'єкти, мости, тунелі, залізничні переїзди та інші пункти вулично-дорожньої мережі, істотні з позиції управління дорожнім рухом, в тому числі, деякі критичні контрольні точки. Структура блоку C-RFID представлена на рис. 7, де модулі (CND-ID, CT, SP, ALB, M, D, CU) позначають: універсальний код автомобіля, приймач, модуль захисту, арифметико-логічний пристрій, модуль пам'яті, дисплей і модуль управління.

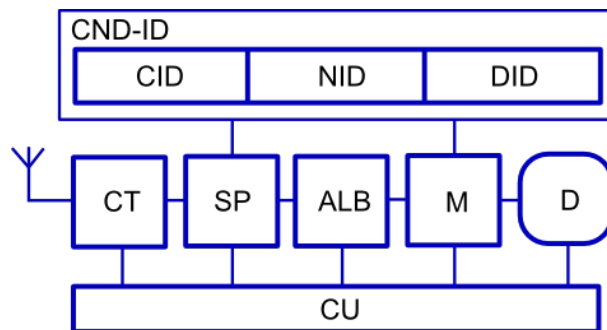


Рис. 7 – Структура блоку C-RFID

Світовий досвід застосування засобів радіочастотної ідентифікації в сфері транспорту дозволяє робити оптимістичні прогнози щодо впровадження подібних технологій в Україні. Так, у 2012 році МВС Росії успішно провело випробування з постановки RFID-міток на номери автомобілів в рамках проекту "Розумне місто". В даному випадку чіп RFID був інтегрований в табличку з номерним знаком, вироблений пітерським ВАТ "Авангард". У Малайзії обов'язкова постановка RFID-чіпа на автомобільний номер введена з 2007 року. Дорожній поліцейський тут може перевірити будь-який автомобіль, навіть не зупиняючи його, як зі стаціонарного поста, так і з мобільної патрульної машини. У США з початку 1990-х років діяла система 3M GM Automotive Adhesive, яку можна вважати прототипом сучасної технології радіочастотної ідентифікації.

Дослідження, проведені Московським технологічним університетом в 2001 році, показали, що технологія RFID дозволяє ідентифікувати як стаціонарні об'єкти, так і рухомі транспортні засоби з досить високою точністю і достовірністю, а також має високу надійність, довговічність і захищеність. Однак поряд з численними перевагами цієї технології мають місце також її недоліки. Перш за все, дальність дії RFID-міток залишає бажати кращого. Проте, опубліковані в *Components & Technologies* результати досліджень російських вчених претендують на дальність до 300 метрів. Відзначається також негативний вплив електронних чіпів на живу і неживу матерію. Так, в червні 2008 року *Journal of the American Medical Association* опублікував результати впливу RFID на медичну техніку. Електронні перешкоди від радіоміток скидають налаштування внутрішньовенних крапельниць, перепрограмують електронні кардіостимулятори і вносять збої в роботу медичної техніки. Більше третини проведених тестів дійсно виявили збої в роботі медичного обладнання, що знаходилося на відстані від сантиметра до шести метрів від джерела RFID. В іншій третині тестів були зафіксовані серйозні порушення в роботі апаратів штучної вентиляції легенів, інфузійних насосів, апарати для гемодіалізу, моніторах електрокардіограм (ЕКГ). Негативний вплив транспондерів на живі організми і людину час від часу мусується в ЗМІ та мережі Інтернет, що створює певні труднощі реальної перспективи електронної паспортизації населення.

У запропонованій системі RFID враховані зазначені вище фактори. Використовуються активні RFID мітки з двома каналами передачі даних – радіочастотним і оптичним. При застосуванні активної мітки дальність дії обмежена, перш за все, вихідною потужністю мітки при незмінному коефіцієнті спрямованої дії антени, чутливості приймального тракту. Система RFID має можливість оперативного регулювання вихідної потужності передавача при обов'язковому обмеженні за максимумом рівня + 4дБм. Це виключає будь-який вплив на живу і неживу матерію, оскільки на кілька порядків менше норми допустимої SAR (Specific Absorption Rate) – за питомим

коефіцієнтом поглинання електромагнітного випромінювання організмом людини. Вимірюється SAR у Ватах на кілограм (Вт/кг). Федеральною комісією зі зв'язку в США (FCC), Міністерством промисловості Канади (IC), а також регулюючими органами деяких інших країн прийнята норма SAR рівна 1,6Вт/кг. У країнах Європейського Союзу прийнята норма SAR рівна 2Вт/кг. Вихідна потужність пропонованої RFID не перевищує одиниць міліватт на відміну від мобільних телефонів з вихідною потужністю до двох ват. Крім того, RFID модуль встановлюється на значній відстані від водія і пасажирів, що повністю виключає негативний вплив високочастотного випромінювання.

Що стосується перешкод медичного обладнання, то, по-перше, на автодорогах таке обладнання відсутнє, а в разі реанемобіля швидкої допомоги медичне обладнання знаходиться всередині екрануючого кузова автомобіля. По-друге, перешкоди створює в основному потужне електромагнітне поле, створюване рідером для живлення транспондера (мітки), і в нашому випадку воно відсутнє, оскільки живлення транспондерів здійснюється не за рахунок поля рідера, а за рахунок бортової мережі автомобіля або резервного акумулятора транспондера. В крайньому випадку, радіоканал транспондера може бути відключений і залишиться тільки оптичний канал.

Доповненням технології RFID може стати GPS навігація. Сучасні GPS приймачі на базі чіпсета SiRF Star III, фіксують сигнал навіть в ангарах і цехах із залізобетонним перекриттям. Приймачі останнього покоління підтримують як сучасну Європейську систему глобального позиціонування Galileo, так і російську систему ГЛОНАСС, що розвивається. Недоліком GPS навігації є неможливість передачі даних про позицію ТЗ на супутник. Таким чином, при розробці підсистеми позиціонування ПДД можна одночасно реалізувати обидві зазначені технології з метою їх детального дослідження для застосування до важливості справ моніторингу та управління транспортними потоками.

Д.8 Структура автомобільного блоку CAR-ID

В основу пропонованої концепції CAR-ID покладені принципи, які використовуються в системі управління повітряним рухом ADS-B. Суть CAR-ID полягає в тому, що транспондер транспортного засобу періодично передає широкомовне повідомлення, яке включає ідентифікаційну інформацію і дані про координати і швидкість транспортного засобу, одержуваних від вбудованого GPS приймача. Крім того, контролер CAR-ID веде протокол динаміки руху транспортного засобу, отримуючи інформацію від датчика прискорень.

Передача повідомлення здійснюється за двома каналами – бездротовому і/або оптичному. Повідомлення приймаються транспортними засобами або стаціонарними станціями, які перебувають в зоні оптичної або радіовидимості. Стаціонарні станції об'єднані в мережу і розташовані в місцях, де є електроживлення (світлофори). Після отримання повідомлення контролер CAR-ID перевіряє його наявність в «історії» і в разі відсутності додає його в пам'ять контролера. При попаданні в зону дії стаціонарного монітора (станції) відбувається скидання всієї інформації, накопиченої з моменту попереднього зчитування, з пам'яті контролера в пам'ять станції. Вона формує інформаційні пакети і періодично передає їх до хмари.

Для забезпечення високої перешкодозахищеності, структурної скритності сигналу і виключення впливу перешкод іншим радіотехнічним засобам блок CAR-ID використовує технологію прямого розширення спектра DSSS. Модуль може працювати в неліцензованому ISM діапазоні з вихідною потужністю 0 – 4дБм. Цього достатньо для забезпечення зони радіовидимості до 100 метрів при застосуванні ненапрямлених антен.

Вся інформація, передана по відкритих каналах, попередньо кодується. Як антиколізійний метод в блоці застосовується метод Slotted – ALOHA. При необхідності вся інформація, накопичена контролером за добу, може бути зчитана поліцейськими або іншими фіскальними службами при наявності спеціального рідера. Таким чином, створюється розподілена інтелектуальна

бездротова мережа на основі використання блоку RFID (рис. 8), перевагою якої є наявність пристроїв розподіленого зберігання і оперативного обміну інформацією.

Структура блоку CAR-ID містить наступні модулі: Optical front-end – оптичний інтерфейс; RF front-end – радіочастотний інтерфейс; Synchrogenerator – генератор частот; Baseband processor – обробки сигналів після демодулювання; GPS – модуль позиціонування; Cryptomodule – модуль криптографічного захисту; Controller, OP-code detect, EEPROM control, Mode control – система управління блоком; Test connector – перемикач тестування модулів; Test logic (Test points) – модуль управління тестуванням і програмуванням; Memory (EEPROM crypto key, ID code) – модуль пам'яті для зберігання даних і службової інформації; MEMS sensors – модуль сенсорних датчиків.

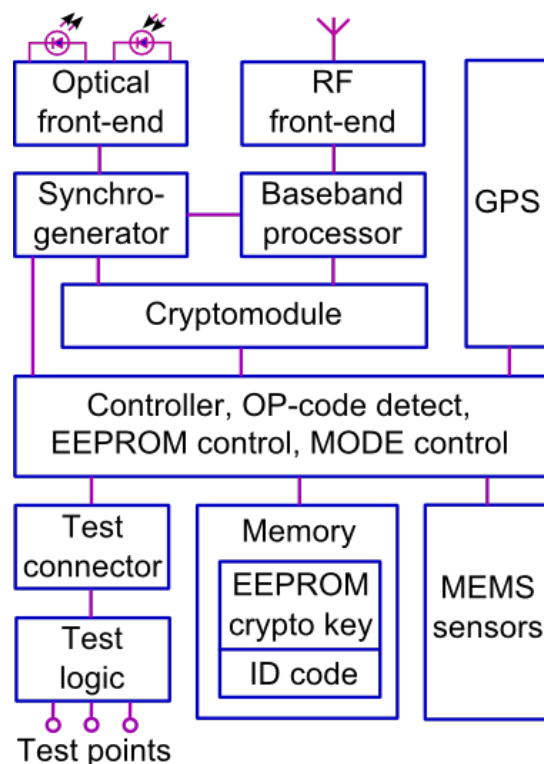


Рис. 8 – Детальна структура блоку C-RFID

Д.9 Засоби моніторингу та управління дорожнім рухом

Сучасні великі міста мають складну дорожню інфраструктуру, де управління дорожнім рухом здійснюється за допомогою світлофорного регулювання із застосуванням систем управління дорожнім рухом (СУДР), що включають в себе сотні світлофорних об'єктів. Тут під світлофорним об'єктом будемо розуміти підсистему СУДР, що забезпечує моніторинг і управління транспортними потоками на окремій ділянці вулично-дорожньої мережі. Центральну частину цієї підсистеми (див. рис. 9) складає дорожній контролер, як правило, спеціалізований з вбудованими комутованими силовими ланцюгами для управління світлофорами. Слід зауважити, що контролери SITRAFFIC C800 спроможні опитувати до 84 детекторів транспорту індуктивного типу і управляти 48 групами сигналів сумарною потужністю 4 кВт в режимі реального часу з максимальним допустимим циклом в 300 секунд. Контролер C800VX підтримує використання до 120 таких модулів в одному сегменті управління, кожен сегмент здатний функціонувати автономно, інтегрується в мережу СУДР на основі бездротових технологій (GPRS, WiMAX) та централізовано управляється з центру управління рухом.

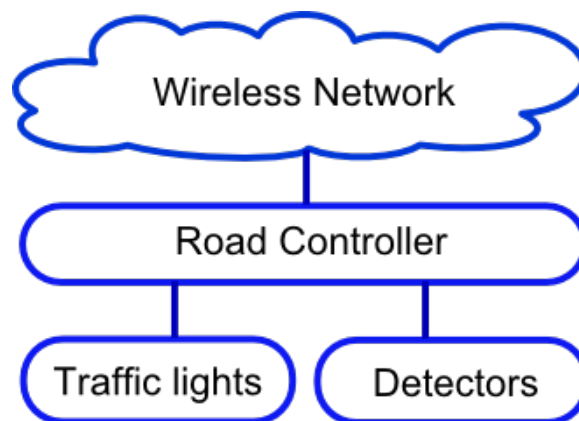


Рис. 9 – Загальна структура світлофорного об'єкта

При збереженні існуючих тенденцій розростання дорожньої інфраструктури стає ясно, що використання подібних рішень ставить під загрозу надійність експлуатації таких систем. Відомо, що з ростом структури

СУДР (фактично це виражається в збільшенні числа світлофорних об'єктів, контрольованих системою) її надійність падає. Тому актуальною науковою і виробничою задачею є розробка більш надійних структур СУДР, серед яких особливе місце займає структура розподіленої автоматизації. Встановлено, що для підвищення надійності системи поряд із забезпеченням інформаційних і керуючих функцій хмари доцільно організувати СУДР по централізовано-децентралізованому принципу. В такому випадку буферний комп'ютер ПДР на рис. 6 виконує функції сервера даних і забезпечує зв'язок з периферійними робочими станціями, які керують кількома контролерами, вся сукупність яких сегментується (10 - 20 світлофорних об'єктів на сегмент) за географічним принципом. Така архітектура суддю дозволяє розташувати сервери в будь-якій точці міста і організувати мобільний центр управління, що забезпечує збереження координованої роботи всіх дорожніх контролерів на об'єкті при виході з ладу центральної частини системи. Структура компонента I-CMC (див. рис. 6) представлена у вигляді матриці (див. рис. 10),

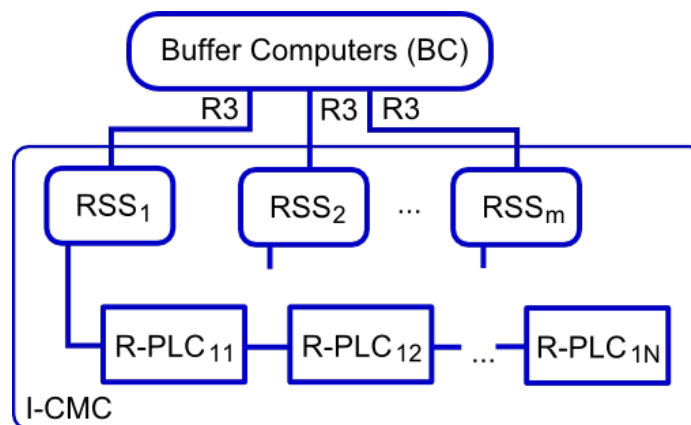


Рис. 10 – Структура блоку I-CMC з рис. 6

Тут модуль RSS являє надійний комп'ютер промислового виконання, а компонент R-PLC побудований на базі компактного і досить потужного програмованого логічного контролера SIMATIC S7-1200 фірми SIEMENS для програмування технологічних процесів для вирішення завдань автоматичного регулювання та керування рухом. Характеристика PLC S7-1200: 1) висока

надійність, середній час напрацювання на відмову – більше 30 років; 2) можливість перепрограмування під час роботи контролера; 3) сервісне обслуговування за місцем розташування об'єкта; 4) продуктивність – 105 команд в секунду з періодом 15 нс; мова програмування STEP-7 Basic з вбудованими засобами нечіткої логіки.

Д.10 Дистанційний модуль "SHERLOCK"

Призначений для побудови розподілених систем контролю і управління об'єктами, включаючи мобільні. Являє собою електронний виріб, побудований на основі використання трьох новітніх технологій Mobile-to-Mobile, GPS і GPRS.

Задачі, які вирішуються за допомогою модуля: 1) Автоматичне визначення місцезнаходження транспортних засобів (AVL). 2) Управління автотранспортним парком, логістика. 3) Автоматизація служб таксі. 4) Моніторинг маршруту і розкладу руху транспорту. 5) Моніторинг режимів експлуатації транспортних засобів. Технічні характеристики: GPS – багатоканальний приймач з високою чутливістю і малим енергоспоживанням, спеціально призначений для роботи в умовах міської забудови та наявності відбитих сигналів. GSM – трьохдіапазонний GSM / GPRS модуль, працює у всіх існуючих на Україні GSM мережах. Входи – 8 цифрових / 1 аналоговий. Виходи – 7 цифрових (відкритий колектор). Інтерфейси – CAN 2.0 шина для підключення до бортової мережі транспортного засобу, управління виконавчими пристроями і опитування додаткових датчиків. Пам'ять – 512 кБ вбудованої пам'яті для зберігання телеметричної інформації. Вбудований датчик температури. Вбудований апаратний самоконтроль. Автомобільний діапазон робочих температур і живлячих напруг.

Дистанційний модуль "SHERLOCK" конструктивно виконаний в компактному пластиковому корпусі, має один 24-контактний системний роз'єм для підключення до джерела живлення, виконавчих пристроїв і датчиків. Два високочастотних SMA роз'єми призначені для підключення GPS і GSM антен. Дистанційний контролер для роботи в режимі GPRS вимагає

конфігурації імені точки доступу (APN, Access Point Name), імені або IP-адреси сервера, і номера порту.

Функціонування модуля. Спроби виходу на зв'язок за допомогою GPRS робляться кожні 10 хв. Дані про зміну координат, прийнятих GPS приймачем, передаються на сервер з інтервалом від 10 до 90 секунд в залежності від швидкості руху об'єкта, на якому встановлено пристрій. Віддалене управління пристроєм здійснюється за допомогою SMS-команд: 1) Запит стану пристрою. 2) Конфігурація режимів роботи по GSM / GPRS. 3) Управління виходами. 4) Запит на виконання USSD команд. 5) Онлайн сервіс моніторингу.

Доступ до онлайн сервісу моніторингу здійснюється цілодобово із сторінки <http://gps.rfid.com.ua>. Для отримання доступу до сервісу користувач повинен авторизуватися за допомогою логіна і пароля. На головній сторінці сервісу більшу частину площі екрану займає вікно з картою, на яку накладаються дані про місцезнаходження мобільних об'єктів. Для візуалізації використовується картографічна інформація компанії ВІЗІКОМ. Вгорі головної сторінки передбачені елементи управління, за допомогою яких можливий вибір одного, двох, або всіх належних користувачу об'єктів, і часового діапазону, для якого необхідно переглянути інформацію про переміщення. При візуалізації на карті піктограмами відображається стан об'єктів і маршрут руху за обраний інтервал часу, а також тривалі стоянки. Розмір і положення карти можна змінювати за допомогою миші і елементів управління. У нижній частині головної сторінки розташовуються елементи управління, що дозволяють швидко перемикатися між частинами маршруту і між об'єктами, а також статистична інформація. При виборі відображення тільки одного об'єкта доступна функція розрахунку відстані.

Налаштування. На сторінці налаштувань користувач може ввести інформацію про свою електронну адресу, змінити пароль, розмір карти і переглянути зведену інформацію про налаштування об'єктів, розсилок. Об'єкти: з даної сторінки можлива зміна назви, опису об'єктів, а також їх параметрів. Правила розсилки: пункт меню, призначений для завдання правил

розсилки повідомлень про переміщення об'єктів між окремими географічними областями, введеними користувачами в систему. Дозволяє в інтерактивному режимі додавати правила розсилки у відповідь на такі події, як входження об'єкта в область, вихід з неї, а також перехід з однієї області в іншу. Области: в даному пункті меню з'являється запрошення визначити замкнуті області на карті, які можуть бути використані при конфігурації розсилок. Координати: на даній сторінці буде відображено короткий звіт статистична інформація про місцезнаходження мобільних об'єктів в поточний момент часу, а також інформація про найближчий географічний об'єкт, відомому системі. У базі даних системи знаходиться інформація про координати декількох десятків тисяч адрес по місту Києву. Зв'язок: даний пункт меню призначений для отримання статистичної інформації про роботу системи. У табличному вигляді відображаються останні координати мобільних об'єктів, час, що минув з моменту їх отримання, інша телеметрична інформація, а також інформація від геокодера про область, в якій знаходиться кожен з об'єктів. Комплект поставки телеметричного модуля: антена GPS; антена GSM; кабель з'єднувальний; інструкція з експлуатації; SIM-карта, а також інформація про найближчому географічному об'єкті, відомому системі. У базі даних системи знаходиться інформація про координати декількох десятків тисяч адрес по місту Києву.

Реальний світ, потребуючи здійснених процесів хмарного управління, давно усвідомив необхідність абсолютно точної радіочастотної цифрової ідентифікації всієї виробленої продукції і природних об'єктів на планеті, включаючи людину і тварин. Наступний крок – створення хмарних цифрових моделей об'єктів реального світу для точного симулювання, моніторингу та управління будь-якими відносинами: природними, соціальними, технічними, технологічними.

MAT (Memory – Address – Transaction) Computing. Комп'ютер (квантовий) майбутнього є пам'ять і адресні транзакції. Пам'ять організовується на будь-якій формі існування матерії. У пам'яті реалізуються

механізми управління і виконання. Адресні транзакції створюють всі процеси і алгоритми. Доставити одну інструкцію з control unit в пам'ять ефективніше, ніж передати величезні обсяги інформації в АЛУ і назад. Очевидно пляшкове горлечко існує 70 років. Дані (big data) слід обробляти там, де вони існують. Це – найближче майбутнє кібер-фізичного комп'ютингу.

Кібер-соціальний комп'ютинг. Повна аналогія. Мегаполіси засмоктують мільйони людей вранці через пляшкове горлечко трафіку жахливих доріг для їх екзакуції в офісах, а ввечері випльовують їх назад по домівках, в близькі й далекі села. Гігантські матеріальні, часові та фінансові витрати на бензин, оренду офісів і переміщення в реальному просторі, стреси в трафіку, забруднення атмосфери. Рішення проблеми – кібер фізичний моніторинг і управління. Доставляти інструктивні впливи через гаджети співробітникам в їх особисті віртуальні кабінети, завдяки узаконеній online організації і створенню максимального комфорту для роботи співробітників, інваріантних до географічній точці позиціонування людини (будинок, готель, подорожі, відпочинок). Умови оплати праці – своєчасне і якісне виконання завдання. Освіта і наука повинні виконуватися в режимі online. On-site зустрічі, лекції та семінари розглядаються як розкіш живого спілкування. Сьогодні існують всі компоненти е-інфраструктури для реалізації кібер соціального комп'ютингу: хмарний сервіс управління, edge gadgets користувачів і розумні речі для реалізації інтеракцій. Єдине, чого немає – законодавства, що легалізує мульти-мільярдну над-інновацію в масштабах країни і планети.

Д. 11 Сьогодення і цифрове майбутнє планети

Аддитивна наноелектроніка, IoT-комп'ютинг і кіберпростір складають сьогодні три еволюціонують рівня ієрархії цифровий планети. На лані даної технологічної культури в даний час працює третина всіх інтелектуальних ресурсів людства, створюючи цифрову карту дзеркального і хмарного відображення фізичних процесів і явищ реального світу з метою підвищення якості життя людства і створення «зеленої» кіберекосістеми. Це означає оцифрування не тільки всіх мобільних і стаціонарних об'єктів, а й «чистого»

структурованого земного простору для точного завдання координат процесів і явищ, що відбуваються на реальній планеті.

Прискорення розвитку знань про природу останнім часом пов'язується з технологічною сингулярністю або вибухом в розумінні законів мікро- і макросвіту на короткому проміжку часу, що неодмінно призведе до створення через два десятиліття глобального кібермозку для управління людством, а також процесами і явищами реального світу. Це стає можливим завдяки розвитку трьох компонентів: біоінженерії, штучного інтелекту кіберпростору і нанотехнологій адитивного виробництва промислових виробів. Тут мається на увазі: 1) Вбудований інтерфейс безпосереднього зв'язку мозку людини з комп'ютером і/або кіберпростором шляхом усунення послідовних мовних інтерфейсів між ними; 2) Створення штучного інтелекту для самонавчання і самовдосконалення небіологічних (комп'ютерних) структур, програм і процесів; 3) Нано-вирощування комп'ютера шляхом адитивного структурування атомів. 4) Найцікавіше рішення пов'язане з невідворотністю відмови людства від функцій управління біологічними, соціальними і технічними об'єктами і процесами на користь кіберсистем.

Ринкова привабливість «зелених» нанотехнологій висхідного проектування – побудова або вирощування (квантового) комп'ютера шляхом структурування атомів – полягає в безвідходності, мікромініатюрності, наднизькому енергоспоживанні, абсолютно мінімальній витратності матеріалів, а в майбутньому і вартості, надвисокій швидкодії і необхідній масштабованості, сумірною з класом сервісів, що доставляються. Сучасні технології дозволяють сьогодні не тільки сканувати атомні структури, а й послідовно будувати або вирощувати їх методом 3D-друку. Однак на шляху вирішення проблеми нано-технологічного напрямку розробки квантового комп'ютера на ринку є три привабливих, але ще не вирішених завдання: 1) Відкриття технологій з високою швидкістю вирощування необхідних гетерогенних атомних структур відповідно до заданої програмної специфікації обчислювача; 2) Створення ефективного транзакційного механізму для

реалізації простого моніторингу та управління квантовими станами вирощених атомних обчислювальних структур з адресованими компонентами;

3) Забезпечення необхідної в часі стабільності станів компонентів атомної структури, що реалізує пам'ять.

Стадії еволюції кіберпростору планети ілюструють послідовні періоди переходу науково-технологічної моди від моніторингу (відображення) фізичних, біологічних і соціальних процесів до їх управління на основі взаємодії реальних і віртуальних структур: 1) 1980-і роки – формування парку персональних комп'ютерів; 2) 1990-ті роки – впровадження Інтернет-технологій у виробничі процеси і побут людини; 3) 2000-і роки – підвищення якості життя за рахунок повсюдного використання мобільних пристроїв і хмарних сервісів; 4) 2010-і роки – створення цифрової інфраструктури моніторингу, управління і взаємодії між собою стаціонарних і рухомих об'єктів, включаючи повітряний, морський, наземний транспорт і роботів; 5) 2015-і роки – створення глобальної цифрової інфраструктури кіберпростору. При цьому досить ясно простежується процес інтелектуалізації та інтеграції фізичних і віртуальних систем: Embedded Systems – Networked Embedded Systems – Cyber Physical Systems – Internet of Things, Data and Services для вирішення наукових, технологічних, економічних, політичних і соціальних проблем. Конкретно, під структурну дикцію кіберфізичних систем підпадають такі галузі: автомобільна промисловість, медицина, енергетика, автоматизація виробництва, мобільний зв'язок, локація і навігація, сільське господарство, транспортна логістика, створення розумних міст, будівель і будинків, соціальні мережі та спільноти, організація дорожнього руху, управління фізичними і віртуальними процесами.

Куди сьогодні прагне реальний кібернетичний світ? Корпоративні мережі, персональні комп'ютери, а також окремі сервіси (програмні продукти) йдуть в хмари кіберпростору, які мають яскраво виражену тенденцію до розшарування Інтернету за спеціалізованими сервісами. Якщо сьогодні 4 мільярди користувачів з'єднуються в інтернеті ($1 \text{ zettabytes} = 10^{21} = 2^{70}$ байт) за

допомогою 50 мільярдів гаджетів, то через п'ять років кожен активний користувач буде мати не менше 10 пристроїв для зв'язку з кіберпростором. Стає неможливим використання персональних гаджетів і комп'ютерів без часткового або повного їх відображення і синхронізації на хмарах Інтернету. Це дає можливість вирішувати проблему віддаленого доступу до особистих даних і сервісів персонального комп'ютера при переміщенні користувачів в просторі. Економічний чинник неефективного використання придбаних додатків, розміщених в гаджетах і персональних комп'ютерах, змушує користувача відмовлятися від їх покупки на користь майже безкоштовної оренди сервісів на хмарах. Все згадане вище є істотним аргументом і незаперечним доказом неминучого переходу всього людства в кіберпростір віртуальних мереж і комп'ютерів, розташованих в професійно надійних хмарах сервісів. Переваги віртуального світу полягають в тому, що мікро-комірки і макро-мережі в хмарах інваріантні по відношенню до численних гаджетів кожного користувача або корпорації. Хмарні технології знімають практично всі згадані вище проблеми надійності, безпеки, сервісного обслуговування та практично не мають недоліків. У зв'язку з глобальним переходом корпорацій і користувачів в хмари надзвичайно актуальною і ринково привабливою стає проблема захисту інформації та компонентів кіберпростору від несанкціонованого доступу, деструктивних проникнень, вірусів. Необхідно створювати надійну, тестопригодну і захищену від несанкціонованих проникнень інфраструктуру кіберпростору і його компонентів (віртуальні персональні комп'ютери і корпоративні мережі) за аналогією з існуючими сьогодні рішеннями в реальному кібернетичному світі. Таким чином, кожен сервіс, що розробляється в реальному світі, повинен бути поміщений у відповідну клітинку хмари, яка об'єднує близькі за функціональними і корисними людині компоненти. Сказане безпосередньо стосується й сервісів проектування цифрових систем на кристалах, які економічно і технічно доцільно зберігати в кіберпросторі для подальшого використання за призначенням. Персональний комп'ютер перетворюється в

зручний інтелектуальний мікромініатюрний інтерфейс для доступу до власного цифрового осередку або до бажаного хмарного сервісу кіберпростору. Для створення персональних інтерфейсів-гаджетів в формі цифрових систем на кристалах ринок нано-електронних технологій надає розробникам до 1 мільярда вентилів на пластині розмірністю 2x2 см при її товщині, що дорівнює 5 мікрон. При цьому сучасні технології допускають створення пакета або «сендвіча», що містить до 7 кремнієвих кристалів. Практично «бездротове» з'єднання таких пластин ґрунтується на технологічній можливості свердління порядку 10 тисяч наскрізних отворів (vias) на 1 квадратному сантиметрі. Крім того, поява тривимірних FinFETs транзисторів і заснованих на них 3D-технологій реалізації об'ємних цифрових систем надають нові можливості для створення більш швидкодіючих обчислювальних пристроїв за рахунок зменшення затримок. Маючи згадані технічні можливості, можна і потрібно використовувати «жадібні» до апаратури моделі і методи для створення швидкодіючих засобів паралельного вирішення практичних завдань. Для цього слід використовувати дискретність і багатозначність структур опису інформаційних процесів, властивість паралелізму, закладене в квантових обчисленнях, що сьогодні є затребуваним при створенні ефективних і інтелектуальних обчислювачів для аналізу кіберпростору, хмарних структур Big Data, а також для проектування нових функціональностей Інтернету. Взаємодія хмарних керуючих сервісів з реальними земними сенсорами і актюаторами (виконавчими механізмами) створює нове, і ще не досить поширене, поняття Traffic Cyber Physical Systems (TCS). Cyber-Physical System – сукупність взаємопов'язаних реальних і віртуальних компонентів з вираженими функціями адекватного фізичного моніторингу та оптимального хмарного управління для забезпечення якості продукції, процесів або сервісів в умовах обмежень на час і ресурси. Одним з її варіантів є система інтелектуального (розумного) хмарного управління дорожнім рухом. Вона заснована на цифровому відображенні в кіберпросторі земної інфраструктури доріг та рухомих об'єктів для подальшого

моделювання всіх процесів на хмарі з метою запропонувати кожному водієві якісні умови виконання маршруту з економією часу і коштів, де сенсорами і актюаторами системи виступають мобільні гаджети або автомобільні комп'ютери. Технологічним середовищем, що об'єднує реальні і віртуальні компоненти кіберсистеми оптимізації трафіку в масштабах планети, є Інтернет. Феномен його становлення і розвитку характеризується інтеграцією взаємодії фізичних і віртуальних об'єктів у просторі і у часі шляхом нашарування і подальшого глибокого взаємного проникнення Інтернет-культур, що історично складаються: 1) Шар об'єднання стаціонарних комп'ютерів в єдину павутину для підвищення обчислювальної та інформаційної потужності планети. 2) Шар інтегрування мобільних пристроїв, що дає можливість комунікації між користувачами в соціальних мережах і доступу до ресурсів Інтернет, інваріантного до місця розташування людини. 3) Шар об'єднання всіх поступово «розумних» (Smart Everything, TV) речей, об'єктів, процесів і явищ з метою їх розпізнавання, моніторингу та управління – Internet of Things. 4) Шар сервісів інтегрування всіх рухомих об'єктів на планеті з метою моніторингу значущих живих істот, позиціонування, навігації та безпілотного управління штучних механізмів і транспортних засобів. Всі рухомі механізми автономно взаємодіють між собою за допомогою Інтернет, сервіси якого дають можливість їх точного позиціонування і квазіоптимальної навігації без участі людини. Таким чином, чотири шари Інтернету, створювані за останні 40 років, утворюють організм замкнутої кібер-фізичної системи планети, яка об'єднує фізичний і віртуальний світи для підвищення якості життя людини. 5) Шар інтегрування накопиченої людством інформації на Big Data структурах, що заповнюються інтелект-історією, і надпотужних паралельних обчислювачів-сервісів для їх аналізу завершує створення кібермозку людства, який буде управляти кібер-фізичним простором (Cyber Physical Space) планети і його вдосконаленням. Таким чином, управління фізичними та соціальними процесами поступово зміщуватиметься з Землі в хмарний кіберпростір, що призведе до створення загальної віртуальної

технології законного управління людством з високим рівнем колективної довіри, вільної від помилок суб'єктивізму соціальних управлінських еліт. Як наслідок, хмарна кіберсистема зробить глобальний і локальний менеджмент справедливим, а планету – максимально зеленою, оскільки головною метою управління буде збереження екосистеми планети та умов для якісного життя як людини, так і соціальних утворень.

Д 12 Ринкова привабливість і характеристика проекту TCS

Кожні чотири роки локально змінюється технологія виробництва товарів і сервісів в певному ринковому сегменті, яка вимагає перенавчання кадрів, зміни інфраструктури, системи управління і відносин, але головне – напрямки руху на ринку товарів і науково-освітніх послуг. Кожні 20 років в передових галузях глобально змінюються технологічні уклади типових виробництв, які вимагають мільярдних капіталовкладень, непідйомних навіть для лідерів в ринкових сегментах, які формують основні капітал-індекси на Уолл-стріт. Це означає, що новий технологічний уклад формується тільки на основі взаємної кооперації тих компаній і країн, які мають вільні капітали, підготовлені кадри і ідеї, точно потрапляють в "десятку" мішені ринкової моди наступного 20-річчя. Саме сьогодні компанії та університети України можуть зробити технологічний ривок на ринку товарів і послуг, включаючи саму модну тріаду (біо- і нано-технології, штучний інтелект), яка формує найважливіший Cyber-Physical-Space-сегмент Nasdaq-ринку: Internet of Things, Smart Everything and Cyber-Physical Systems. Україна повинна приєднатися кадровим потенціалом до науково-технологічних перегонів євро-американо-азіатських компаній, щоб зайняти в майбутньому призове місце на п'єдесталі економічного зростання Європи. Потенціал країни фантастично великий (кадри, центрова територія, науково-освітня культура). Для його використання в модному ринковому напрямку Cyber-Physical-Space повинна бути вироблена політична воля – керуючий вплив, що створює всебічні умови для формування науково-технологічної культури, як основи європейського майбутнього України. Найцікавіші глобальні проекти сьогодні виконуються під егідою

об'єднання двох просторів в єдине ціле. Кібер-фізичний простір – метрика взаємодії (взаємної інтеграції) фізичних, біологічних і соціальних об'єктів, процесів і явищ з віртуальними або хмарними технологіями управління - покликане інтегрувати найбільш перспективні кібер-фізичні технологічні рішення. Найцікавіше з них пов'язано з невідворотністю відмови людства від функцій управління біологічними, соціальними і технічними об'єктами і процесами на користь кіберсистем.

Джерела фінансування проекту: 1) Державні структури (міністерства: освіти і науки, транспорту, внутрішніх справ); 2) Приватні інвестори України та зарубіжжя; 3) Європейські та американські міждержавні програми; 4) Start-up пропозиції для інверстіційних програм провідних компаній планети (Google, IBM, Microsoft, Apple); 5) Венчурні і автомобільні компанії розвинених країн; 6) Власні кошти розробників і учасників проекту.

Джерела отримання прибутку: 1) Орендна плата за хмарний сервіс управління трафіком з боку водіїв; 2) Плата за розміщення реклами; 3) Орендна плата за корпоративний хмарний сервіс моніторингу і управління трафіком транспорту підприємства; 4) Відрахування державних служб за спеціальний сервіс моніторингу та управління транспортними засобами; 5) Відрахування муніципальних державних служб за сервіс моніторингу транспортними засобами та управління вуличними (віртуальними) світлофорами міської інфраструктури; 6) Продаж акцій, завдяки капіталізації бізнесу та відкритого акціонування активів компанії-розробника хмарного сервісу; 7) Продаж Start-up TCS-проекту однієї з компаній Google, Apple або Samsung за 150 мільйонів доларів.

Техніко-економічне обґрунтування проекту в масштабах України і світу: 1) Країна має сьогодні 120 тисяч фірм, серед яких 70 тисяч – суб'єкти IT-індустрії, понад 7 тисяч – транспортні компанії і державні структури, зацікавлені в сервісах моніторингу та управління корпоративних транспортних засобів. Крім того, 9 мільйонів автомобілів по країні потенційно потребують хмарних сервісах, що забезпечують моніторинг і управління

дорожньої обстановки по замовленому маршруту руху. Далі, обслуговування та експлуатаційні витрати одного світлофора обходяться казні приблизно в 50 доларів на рік, а його установка коштує від 1000 до 10000 доларів. Кількість таких пристроїв по Україні дорівнює 80 000 штук. Їх разова установка і експлуатаційні витрати (за 10 років служби) складають в середньому: $(5500 + 500) \times 80000 = \$ 480\,000\,000$ бюджетних грошей. Півмільярда доларів на світлофори можна перетворити для бюджету країни практично в нульові витрати на віртуальне розміщення та експлуатацію в хмарному сервісі пристроїв управління трафіком. У масштабах всієї планети можна звільнити для потреб людства близько 500 мільярдів доларів. До того ж весь арсенал земних світлофорів "пожирає" мільйони кіловат-годин електроенергії, що робить планету зеленішою. Сюди ж можна додати сотні тисяч тонн металу, сотні тисяч кілометрів кабелів і проводів. Без усього перерахованого можна обійтися, простим перенесенням світлофорів в кіберпростір управління трафіком, накладених на карту інфраструктури земних доріг. Економічний ефект від перенесення в масштабі України – 500 мільйонів доларів тільки за рахунок ліквідації витрат на утримання і установку пристроїв управління перехрестями. 2) Друга частина ринково орієнтованої інновації проекту полягає у 15% економії палива за рахунок пропозиції з боку хмарного сервісу квазіоптимальних маршрутів в реальному часі, а також в 15% зменшенні часу виконання поїздки, що істотно вплине на собівартість перевезення вантажів та пасажирів. Економія від використання хмарного сервісу для одного водія на рік становить: 15000 гривень – середня вартість річного споживання бензину $\times 0,15 = 2250$ гривень або 200 доларів. В цілому по країні це дасть можливість зберегти в гаманці власників автомобілів кругленьку суму, порядку: $200 \times 9000000 = 1\,800\,000\,000$ доларів. Приблизно така ж сума виходить за рахунок 15% економії часу проходження замовленого маршруту, що по країні створює інтегральну економію коштів трудящих у розмірі 3,6 мільярда доларів на рік. 3) Витрати на створення промислового макета масштабованої хмарної кібер-фізичної системи управління трафіком міського фрагмента (вулиця

Клочківська, місто Харків) включають покупку і інсталяцію програмно-апаратного забезпечення ($M1=\$200000$), розробку і верифікацію нових програмно-апаратних компонентів хмарного управління, інтегрованих зі згаданими вище промисловими покупними виробами ($M2=\$500000$), оснащення 100 автомобілів, що беруть участь в експериментах, моніторами для хмарного управління транспортом ($M3 = \$ 100000$). Оснащення перехресть міського фрагмента інфраструктури мікроконтролерами з приймально-передавачами для створення дублюючого вулицю хмарного макета, синхронно працює з реальними світлофорами ($M4 = \$ 100000$). 4) Сумарні витрати для проведення реальних експериментів на фрагменті міської інфраструктури складають: $M = M1 + M2 + M3 + M4 = \$ 900000$. Накладні витрати – $\$ 100000$. У масштабі України інтегральні витрати не будуть носити адитивного характеру в процесі масштабування, оскільки система управління створюється інваріантною до розмірності інфраструктурної ділянки доріг, але яка потребує створення потужного дата центру, вартістю $\$ 10\ 000\ 000$. Крім того, були здійснені витрати, пов'язані з нанесенням і актуалізацією всіх дорожніх світлофорів (дорожніх знаків) на існуючі карти, близько $\$ 1000000$. Таким чином, інтегральні витрати на створення масштабованого програмно-технічного комплексу хмари кібер-фізичної системи управління, інтегрованого з фрагментом міської інфраструктури та з автомобільною експериментальною групою складають: $900000 + 10000000 + 1000000 + 100000 = 12\ 000\ 000$ доларів. 5) Доходи компанії-резидента, в доларах, від впровадження хмарної системи управління трафіком будуть визначатися наступними складовими: надходженням орендної оплати за хмарний сервіс від транспортних компаній $7000 \times 1000 = 7000000$. Оплатою трафік-сервісів, що надходить від водіїв особистих автомобілів $100 \times 9\ 000\ 000 = 900\ 000\ 000$. З огляду на те, що кіберсистема хмарного управління транспортом стане за значимістю рівною Google-сервісам, то рівень капіталізації такого Інтернет-шару очікується близько 10-50 мільярдів доларів. При цьому очікувані надходження від реклами складуть не менше 5 відсотків (500 мільйонів

доларів). Через 5-10 років автомобільний кібер-шар Інтернету об'єднає в інтелектуальну мережу один мільярд автомобілістів разом з автомобілями і 4 мільярди жителів планети. Рівень капіталізації всіх компаній, які формують кібер-шар управління транспортом, може інтегрально досягти 300-500 мільярдів доларів. Автомобіль з монітором, комп'ютером та Інтернетом стане робочим місцем і зоною відпочинку, де водій буде проводити до 20% свого життя, вирішуючи виробничі і соціальні проблеми спілкування в процесі пересування. Економічно від впровадження сервісу управління виграють всі: водії – від економії палива і колійного часу, компанії – від зниження собівартості доставки вантажів і пасажирів, держава і природа – від зниження злочинності, транспортних правопорушень, рівня вихлопних газів і підвищення безпеки руху транспорту.

Стадії інновації на стороні сервер-хмари і гаджет-користувача: 1) Створення квантових структур даних для паралельного зберігання даних і матричного аналізу Big Data форматів кіберпростору. 2) Розробка квантоподібних віртуальних паралельних процесорів для швидкого пошуку, розпізнавання та прийняття квазіоптимальних рішень з управління транспортом та інфраструктурою міст. 3) Створення комплексу програм для реалізації хмарних сервісів моніторингу та управління транспортом і дорожнім рухом в реальному часі на основі використання віртуальних хмарних світлофорів. 4) Розробка і практична реалізація моделі експериментальної ділянки міста для управління дорожнім рухом за допомогою хмарних світлофорів, моніторингу та управління транспортними засобами з анімаційної візуалізацією всіх масштабованих процесів на екрані монітора.

Власна оцінка науково-технічного рівня проекту: 1) Не має аналогів в Україні – в частині створення хмарного сервісу моніторингу та управління дорожнім рухом, інфраструктурою вуличних світлофорів і надання сервісів прокладки оптимальних маршрутів в реальному масштабі часу. 2) На рівні існуючих в світі аналогів – в частині інтелектуального управління вуличними

світлофорами, що враховує дорожню ситуацію на кожному перехресті в режимі реального часу. 3) Не має аналогів в світі – в частині реалізації керування транспортними потоками на основі використання розумних (віртуальних) хмарних світлофорів, виведених на екрани моніторів автомобілів і керованих реальними транспортними потоками.

Стадія готовності бізнес-плану: 1) Проведення маркетингових досліджень – 90% від запланованих заходів. 2) Рівень опису бізнес-процесів – 80%. 3) Техніко-економічне обґрунтування – 90%. 4) Стан розрахунків – 80%. 5) Календарний план виконання проекту – 80%. 6) План виробництва масштабованих віртуальних інфраструктур міст – 50%. 7) План збуту продукції – віртуальних інфраструктур міст і хмарних сервісів – 70%.

Права інтелектуальної власності: заявки на отримання патентів: 1) Модель автоматного моніторингу та управління дорожнім рухом на основі використання віртуального вуличного світлофора, відображуваного на екрані автомобільного монітора. 2) Автоматна модель кібер-фізичної системи, що створює взаємодію реальної і віртуальної інфраструктури дорожнього руху, які формують сервіс фізичного моніторингу, хмарного управління і прокладання квазіоптимальних маршрутів. 3) Модель взаємодії квантоподібних структур даних з віртуальним процесором для паралельної обробки інформації з метою управління світлофорами і рухом автомобілів за вибраним маршрутом в реальному часі. 4) Модель розумного світлофора, організуючого, спільно з автомобільними комп'ютерами або гаджетами, обчислювальну мережу для квазіоптимального, с позиції мінімізації сумарного времени, переїзду перехрестя всіма учасниками дорожнього руху.

Створення математичного кібер-шару проекту в формі моделей і методів, Big Data структур даних, нових віртуальних хмарних (Memory-Transaction) МТ-обчислювачів і засобів анімаційного моделювання кіберпростору управління транспортом. Розробка і реалізація алгоритмів квазіоптимального управління світлофором, оптимізації маршрутів руху з використанням історії. Створення алгоритмів ергономічної анімації з метою

візуалізації процесів моніторингу та управління дорожніх ситуацій, пов'язаних з перехрестями на стороні хмари (сервера). Розробка алгоритмів ергономічної візуалізації керуючої інфраструктури на існуючих картах для виведення графічних даних на монітори автомобілів. Програмна імплементація алгоритмів в хмарні сервіси транспортного шару кіберпростору. Розробка технологічного кібер-шару проекту в формі взаємодії програмно-апаратних компонентів кібер-фізичної системи, що інтегрує існуючі і нові розробки: картографію, світлофори, автомобільні комп'ютери в єдину автоматну модель управління транспортом в реальному часі. Проектування інформаційного, методичного та лінгвістичного забезпечень кіберсистеми. Тестування і верифікація віртуальної частини кіберсистеми. Створення технічного і організаційного кібер-шарів проекту для подальшого виконання заходів по інтегруванню реальної керуючої інфраструктури доріг та автомобілів з віртуальним кіберпростору. Розробка глибокої мультиверсної системи захисту всіх програмно-технічних компонентів хмарного сервісу управління транспортом і подальша верифікація промислового зразка системи хмарного управління трафіком на фрагменті міської інфраструктури з використанням спеціально оснащених автомобілів і розумних хмарних світлофорів, синхронізованих з реальними. Створення дата актуалізації картографічного шару шляхом віртуального нанесення на електронну карту інфраструктури світлофорів і їх синхронізація з реальними пристроями управління перехрестями на перехідний період. Створення Інтернет-системи банківської підтримки для акумулювання постійних, періодичних орендних і разових платежів. Проектування документації супроводу проекту, рекламна кампанія за пріоритетами кіберсистеми хмарного управління дорожнім рухом з використанням прес-релізів, телебачення, Інтернету, міжнародних конференцій.

Д 13 Завдання, що підлягають вирішенню для реалізації TCS-хмари

- 1) Впровадження технології радіочастотного цифрового позиціонування транспортного засобу (гаджета) з точністю до $2 - 0,2$ метрів.
- 2) Створення нової системи мнемонічних і супроводжуючих звукових сигналів на моніторі, що становлять хмарні правила дорожнього руху.
- 3) Розробка операційного та керуючого автоматів, які об'єднують всі компоненти TCS в єдину хмарну систему, що має входом і виходом гаджету $G(k)$ (комп'ютер автомобіля), як єдиний інтерфейс зв'язку з хмарою, що позиціонується в реальному масштабі часу точними координатами k на карті M інфраструктури зі світлофорами L для управління пересуванням учасника дорожнього руху.
- 4) Проектування масштабованої системи TCS для світлофора, проспекту, району, міста, країни, планети.
- 5) Розробка сервісів (програмних додатків) на стороні клієнта (автомобіль, пішохід, велосипедист, мотоцикліст) для замовлення опцій квазіоптимального управління виконанням маршрутів пересування, включаючи систему відео- і аудіо-сигналів, прив'язаних до інфраструктури і доставляються з TCS в реальному масштабі часу.
- 6) Проектування сервісів на стороні хмари (сервера) для вирішення завдань оптимізації при прокладці маршрутів, управлінні світлофором, модифікації хмарної і реальної інфраструктури, а також сервісів оперативного управління хмарними світлофорами на основі моніторингу дорожнього руху.
- 7) Розробка кубітних структур даних і «квантових» матричних процесорів (обчислювачів) на основі технологій Big Data для одночасного і паралельного сервісного обслуговування користувачів TCS-хмари в реальному масштабі часу, число яких в межі має бути рівною кількості жителів планети.
- 8) Створення інтелектуальних моделей, методів синтезу та аналізу віртуальної інфраструктури для оцінювання якості дорожнього руху,

моделювання та візуалізації трафіку, пропозиції оптимального маршруту з урахуванням технічних, кліматичних, соціальних факторів, якості доріг, кількості світлофорів, лівих поворотів з метою створення нових і реконструкції існуючих інфраструктур дорожнього руху.

9) Розробка хмарних сервісів для автотранспортних підприємств з метою підвищення якості обслуговування пасажирів, перевезення вантажів, оптимізації часових і матеріальних витрат.

10) Створення хмарних сервісів для водія з метою підвищення якості проїзду за заданим маршрутом і оптимізації часових і матеріальних витрат.

11) Збір статистичної інформації (інтелектуалізація глобальної, корпоративної та персональної інфраструктури) шляхом накопичення історії трафіку, зміни його параметрів у часі і в просторі для прокладання квазіоптимальних маршрутів майбутніх поїздок.

12) Створення засобів захисту інформації та санкціонованого доступу до персональних і корпоративних даних в хмарі. Кожен користувач бачить на карті тільки свій автомобіль в хмарі і знеособлені транспортні потоки. Всі ідентифікатори транспорту можуть бути доступні за рішенням суду або постановою слідчих органів тільки спеціальним державним службам.

13) Створення специфікації TCS-хмари як start-up проекту для його подальшої пропозиції або продажу компаніям Apple і Google.

14) Пошук валідних партнерів і заможних інвесторів в розвинених країнах (Росія, США, Німеччина, Естонія, Іран, Саудівська Аравія, Польща, Франція, Швеція, Норвегія) для реалізації спільного проекту Smart Cloud Traffic Control шляхом отримання грантів в рамках європейських та національних програм на виконання масштабованого прототипу системи в межах міста, країни.

15) Патентування в Україні, США та інших країнах віртуальних: технології та інфраструктури управління дорожнім рухом на основі використання в реальному часі хмарних світлофорів, що відображаються на дисплеях гаджетів, одночасно використовуваних як сенсори і актюатори.

ДОДАТОК Е

ЛІСТИНГИ ПРОГРАМ

Е1. HDL-модель пристрою Sherlock (car monitoring)

```

                                dwt9_7.v
`timescale 1 ns / 10 ps
module dwt9_7 ( clk, en, rst, data_in, done, hh, hl, lh, ll, level, tile_size );
parameter [31:0] n_data =9;
parameter [31:0] n_koef =8;
parameter [31:0] tile =8;
parameter [31:0] adr_size =8;

    input clk;
    input en;
    input rst;
    input [(n_data - 1):0] data_in ;
    output [3:0] done ;
    output [(n_data - 1):0] hh ;
    output [(n_data - 1):0] hl ;
    output [(n_data - 1):0] lh ;
    output [(n_data - 1):0] ll ;
    output [2:0] level ;
    output [2:0] tile_size ;
    parameter [2:0] t=tile;
    wire net2903;
    wire select_data;
    wire [3:0] done_buf;
    wire [(n_data - 1):0] ll_buf;
    wire [(n_data - 1):0] ll_in;
    wire [3:0] done;
    wire [(n_data - 1):0] hh;
    wire [(n_data - 1):0] hl;
    wire [(n_data - 1):0] lh;
    wire [(n_data - 1):0] ll;
    wire [2:0] level;
    wire [2:0] tile_size;
    assign {tile_size}=tile;
    fifoclr_cc #(n(n_data),.size(((tile * tile) / 4)),.adr_size(adr_size))
u1(.clock(clk), .fifo_gsr(rst), .read_enable(select_data),
.write_data_in(ll_buf[(n_data - 1):0]), .write_enable(done_buf[0]),
.empty_out(/*open */), .full_out(/*open */), .read_data_out(ll_in[(n_data -
1):0]));
    //Parameter Order: adr_size,n,size
    dwt_block u2(.clk(clk), .data_in(data_in[(n_data - 1):0]),
.ll_in(ll_in[(n_data - 1):0]), .rst(rst), .en(en), .done_buf(done_buf),
.hh(hh[(n_data - 1):0]), .hl(hl[(n_data - 1):0]), .lh(lh[(n_data - 1):0]),
.ll_buf(ll_buf[(n_data - 1):0]), .level(level), .sel(select_data));
    //Parameter Order: n_data,n_koef,size,adr_size,tile
    assign {done}=done_buf;
    assign {ll}=ll_buf[(n_data - 1):0];
endmodule

                                dwt_block.v
`timescale 1 ns / 10 ps
module dwt_block ( clk, rst, en, data_in, ll_in, sel, done_buf, hh, hl, lh,
ll_buf, level );
parameter [31:0] n_data =9;
parameter [31:0] n_koef =8;
parameter [31:0] size =16;
parameter [31:0] adr_size =8;
parameter [31:0] tile =8;
    input clk;
    input rst;
    input en;
    input [(n_data - 1):0] data_in ;
    input [(n_data - 1):0] ll_in ;

```

```

output sel;
output [3:0] done_buf ;
output [(n_data - 1):0] hh ;
output [(n_data - 1):0] hl ;
output [(n_data - 1):0] lh ;
output [(n_data - 1):0] ll_buf ;
output [2:0] level ;
wire data_select;
wire [(n_data - 1):0] c0_h;
wire [(n_data - 1):0] c0_l;
wire [(n_data - 1):0] c1_h;
wire [(n_data - 1):0] c1_l;
reg [(n_data - 1):0] c2_h;
reg [(n_data - 1):0] c2_l;
wire [4:0] c_alu_column;
wire [1:0] c_alu_raw;
wire [(n_data - 1):0] d0_h;
wire [(n_data - 1):0] d0_in;
wire [(n_data - 1):0] d0_l;
wire [(n_data - 1):0] d1_h;
wire [(n_data - 1):0] d1_in;
wire [(n_data - 1):0] d1_l;
wire [(n_data - 1):0] data;
wire [3:0] done;
wire [4:0] enable_h;
wire [4:0] enable_l;
wire [(n_data - 1):0] h;
wire [(n_data - 1):0] h2;
wire [(n_data - 1):0] l;
wire [(n_data - 1):0] l2;
wire [4:0] pop_h;
wire [4:0] pop_l;
wire [(n_data - 1):0] s_h;
wire [(n_data - 1):0] s_in;
wire [(n_data - 1):0] s_l;
wire [(n_data - 1):0] x0;
wire [(n_data - 1):0] x1;
wire [(n_data - 1):0] x2;
wire sel;
wire [3:0] done_buf;
wire [(n_data - 1):0] hh;
wire [(n_data - 1):0] hl;
wire [(n_data - 1):0] lh;
wire [(n_data - 1):0] ll_buf;
wire [2:0] level;
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        c2_l <= {(n_data - 1)-0+1- 0}{1'b0};
    else
        c2_l <= 1;
end// always
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        c2_h <= {(n_data - 1)-0+1- 0}{1'b0};
    else
        c2_h <= h;
end// always
assign {data}=(data_select == 1'b1) ?
        data_in
        : ll_in;
    column h_mem(.clk(clk), .in_data(h[(n_data - 1):0]), .rst(rst),
.d0_in(d0_in[(n_data - 1):0]), .d1_in(d1_in[(n_data - 1):0]), .pop(pop_h),
.push(enable_h), .s_in(s_in[(n_data - 1):0]), .c0(c0_h[(n_data - 1):0]),
.c1(c1_h[(n_data - 1):0]), .d0(d0_h[(n_data - 1):0]), .d1(d1_h[(n_data - 1):0]),
.s(s_h[(n_data - 1):0]));
//Parameter Order: n_data,n_koef,tile,adr_size
    column l_mem(.clk(clk), .in_data(l[(n_data - 1):0]), .rst(rst),
.d0_in(d0_in[(n_data - 1):0]), .d1_in(d1_in[(n_data - 1):0]), .pop(pop_l),
.push(enable_l), .s_in(s_in[(n_data - 1):0]), .c0(c0_l[(n_data - 1):0]),

```

```

.cl1(c1_l[(n_data - 1):0]), .d0(d0_l[(n_data - 1):0]), .d1(d1_l[(n_data - 1):0]),
.s(s_l[(n_data - 1):0]));
//Parameter Order: n_data,n_koef,tile,adr_size
row_buffer u1(.clk(clk), .rst(rst), .data(data[(n_data - 1):0]), .en(en),
.x0(x0[(n_data - 1):0]), .x1(x1[(n_data - 1):0]), .x2(x2[(n_data - 1):0]));
//Parameter Order: n_data
control_block #(.adr_size(adr_size),.size(tile)) u3(.clk(clk), .rst(rst),
.en(en), .c_alu_column(c_alu_column), .c_alu_row(c_alu_row), .enable_h(enable_h),
.enable_l(enable_l), .level(level), .data_select(data_select), .done(done),
.pop_h(pop_h), .pop_l(pop_l));
//Parameter Order: adr_size,size
arith_for_row u4(.c_alu_raw(c_alu_raw), .clk(clk), .rst(rst), .en(en),
.x0(x0[(n_data - 1):0]), .x1(x1[(n_data - 1):0]), .x2(x2[(n_data - 1):0]),
.h(h[(n_data - 1):0]), .l(l[(n_data - 1):0]));
//Parameter Order: n_data,n_koef
arith_for_column u5(.c0_h(c0_h[(n_data - 1):0]), .c0_l(c0_l[(n_data - 1):0]),
.c1_h(c1_h[(n_data - 1):0]), .c1_l(c1_l[(n_data - 1):0]), .c2_h(c2_h[(n_data -
1):0]), .c2_l(c2_l[(n_data - 1):0]), .c_alu_column(c_alu_column), .clk(clk),
.rst(rst), .d0_h(d0_h[(n_data - 1):0]), .d0_l(d0_l[(n_data - 1):0]),
.d1_h(d1_h[(n_data - 1):0]), .d1_l(d1_l[(n_data - 1):0]), .en(en),
.s_h(s_h[(n_data - 1):0]), .s_l(s_l[(n_data - 1):0]), .h(h2[(n_data - 1):0]),
.l(l2[(n_data - 1):0]), .d0_out(d0_in[(n_data - 1):0]), .d1_out(s_in[(n_data -
1):0]), .s_out(d1_in[(n_data - 1):0]));
//Parameter Order: n_data,n_koef
out_buf #(.m(n_data),.m_in(n_data)) u6(.done(done), .h(h2[(n_data - 1):0]),
.l(l2[(n_data - 1):0]), .clk(clk), .en(en), .rst(rst), .done_buf(done_buf),
.hh(hh[(n_data - 1):0]), .hl(hl[(n_data - 1):0]), .lh(lh[(n_data - 1):0]),
.ll_buf(ll_buf[(n_data - 1):0]));
//Parameter Order: m,m_in
assign {sel}=data_select;
endmodule

```

Row_buffer.sherlock

```

`timescale 1 ns / 10 ps
module row_buffer ( data, clk, rst, en, x0, x1, x2 );
parameter [31:0] n_data =9;
input [(n_data - 1):0] data ;
input clk;
input rst;
input en;
output [(n_data - 1):0] x0 ;
output [(n_data - 1):0] x1 ;
output [(n_data - 1):0] x2 ;
reg [(n_data - 1):0] x0_in;
reg [(n_data - 1):0] x1_in;
wire [(n_data - 1):0] x0;
wire [(n_data - 1):0] x1;
wire [(n_data - 1):0] x2;
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        begin
            x0_in <= {((n_data - 1)-0+1- 0){1'b0}};
            x1_in <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                x0_in <= x1_in;
                x1_in <= data;
            end
        end// always
    assign {x0}=x0_in;
    assign {x1}=x1_in;
    assign {x2}=data;
endmodule

```

Arith_for_row.sherlock

```

`timescale 1 ns / 10 ps
module arith_for_row ( x0, x1, x2, clk, rst, en, c_alu_raw, l, h );
parameter [31:0] n_data =9;
parameter [31:0] n_koef =8;
parameter [31:0] n =9;

```

```

input [(n_data - 1):0] x0 ;
input [(n_data - 1):0] x1 ;
input [(n_data - 1):0] x2 ;
input clk;
input rst;
input en;
input [1:0] c_alu_raw ;
output [(n_data - 1):0] l ;
output [(n_data - 1):0] h ;
reg [n_data:0] l1__temp1;
reg [(n_data + n_koef) - 1):0] l_out__temp;
parameter [31:0] p1_const=32'b11100101100000110011100110011110;
parameter [31:0] u1_const=32'b10000011011001000000011010111001;
parameter [31:0] p2_const=32'b00111000100000011001110101110101;
parameter [31:0] u2_const=32'b00011100011000100110101010010000;
parameter [31:0] k0_const=32'b00110100000001100111000010100100;
parameter [31:0] k1_const=32'b00100111010111011001011000010110;
parameter [(n_koef - 1):0] p1={p1_const[31],(~ (p1_const[30:((31 - n_koef) +
1])) + 1'b1)};
parameter [(n_koef - 1):0] u1={u1_const[31],(~ (u1_const[30:((31 - n_koef) +
1])) + 1'b1)};
parameter [(n_koef - 1):0] p2=p2_const[31:((31 - n_koef) + 1)];
parameter [(n_koef - 1):0] u2=u2_const[31:((31 - n_koef) + 1)];
parameter [(n_koef - 1):0] k0=k0_const[31:((31 - n_koef) + 1)];
parameter [(n_koef - 1):0] k1=k1_const[31:((31 - n_koef) + 1)];
wire first;
wire last;
reg first0;
reg last0;
reg first1;
reg last1;
reg first2;
reg last2;
reg first_del;
reg last_del;
reg [(n_data - 1):0] x0_del;
wire [(n_data - 1):0] x2_in;
reg [(n_data - 1):0] d0_1;
reg [(n_data - 1):0] d0_0;
reg [(n_data - 1):0] d0_0_del;
wire [(n_data - 1):0] d0_01;
wire [(n_data - 1):0] d0_03;
reg [(n_data - 1):0] s0_1;
reg [(n_data - 1):0] d0_02;
wire [(n_data - 1):0] s0_11;
reg [(n_data - 1):0] s0_0_del;
reg [(n_data - 1):0] s0_0;
reg [(n_data - 1):0] s0_01;
reg [(n_data - 1):0] d1_1;
wire [(n_data - 1):0] d1_01;
reg [(n_data - 1):0] d1_0_del;
reg [(n_data - 1):0] d1_0;
reg [(n_data - 1):0] s1;
reg [(n_data - 1):0] l;
reg [(n_data - 1):0] h;
assign {first}=c_alu_raw[1];
assign {last}=c_alu_raw[0];
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            first0 <= 1'b0;
            last0 <= 1'b0;
            first1 <= 1'b0;
            last1 <= 1'b0;
            first2 <= 1'b0;
            last2 <= 1'b0;
            first_del <= 1'b0;
            last_del <= 1'b0;
        end
end

```

```

else
    if ((en == 1'b1))
        begin
            first0 <= first;
            last0 <= last;
            first1 <= first0;
            last1 <= last0;
            first2 <= first1;
            last2 <= last1;
            first_del <= first2;
            last_del <= last2;
        end
// end always
assign {x2_in}=(last == 1'b1) ?
        x0
        : x2;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            x0_del <= {((n_data - 1)-0+1- 0){1'b0}};
            d0_1 <= {((n_data - 1)-0+1- 0){1'b0}};
            d0_0_del <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                x0_del <= x0;
                d0_1 <= f_arith__0(x0, x1, x2_in, p1);
                d0_0_del <= d0_0;
            end
// end always
assign {d0_01}=(first == 1'b0) ?
        d0_0
        : d0_1;
assign {d0_03}=(last2 == 1'b0) ?
        d0_01
        : d0_0;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            d0_0 <= {((n_data - 1)-0+1- 0){1'b0}};
            d0_02 <= {((n_data - 1)-0+1- 0){1'b0}};
            s0_1 <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                d0_0 <= d0_0_del;
                s0_1 <= f_arith__1(d0_01, x0_del, d0_1, u1);
                d0_02 <= d0_03;
            end
// end always
assign {s0_11}=(last_del == 1'b1) ?
        s0_1
        : s0_0;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            s0_0_del <= {((n_data - 1)-0+1- 0){1'b0}};
            s0_0 <= {((n_data - 1)-0+1- 0){1'b0}};
            d1_1 <= {((n_data - 1)-0+1- 0){1'b0}};
            s0_01 <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                s0_0_del <= s0_1;
                s0_0 <= s0_0_del;
                s0_01 <= s0_0;
            end

```

```

        d1_1 <= f_arith__2(s0_0, d0_02, s0_11, p2);
    end
// end always
    assign {d1_01}=(first_del == 1'b1) ?
        d1_1
        : d1_0;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            d1_0_del <= {((n_data - 1)-0+1- 0){1'b0}};
            d1_0 <= {((n_data - 1)-0+1- 0){1'b0}};
            s1 <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                d1_0_del <= d1_1;
                d1_0 <= d1_0_del;
                s1 <= f_arith__3(d1_01, s0_01, d1_1, u2);
            end

// end always
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            h <= {((n_data - 1)-0+1- 0){1'b0}};
            l <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                l_out__temp = (k1 * d1_1);
                h <= {l_out__temp[((n_data + n_koef) -
1)],l_out__temp[((n_data + n_koef) - 2):n_koef]};
                l_out__temp = (k0 * s1);
                l <= {l_out__temp[((n_data + n_koef) -
1)],l_out__temp[((n_data + n_koef) - 2):n_koef]};
            end
// end always

function [(n - 1):0] f_arith__0;
    input [(n_data - 1):0] x0;
    input [(n_data - 1):0] x1;
    input [(n_data - 1):0] x2;
    input [(n_koef - 1):0] koef;
    parameter n = ((n_data - 1)-0+1); //constant
    parameter m = ((n_koef - 1)-0+1); //constant
    reg [n:0] temp1;
    reg [(n + m):0] temp2;
    reg [(n + m):0] temp3;
    reg [n:0] x0_t;
    reg [n:0] x2_t;
    reg [(n + m):0] x1_t;
    reg [(n - 1):0] temp4;
    begin : Function
        begin
            x1_t = {((n + m)-0+1- 0){1'b0}};
            x0_t = {x0[(n - 1)],x0[(n - 1):0]};
            x2_t = {x2[(n - 1)],x2[(n - 1):0]};
            temp1 = (x0_t + x2_t);
            temp2 = (koef * temp1);
            x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]},x1[(n - 1):0]};
            temp2[(n + m)] = temp2[((n + m) - 1)];
            temp3 = (x1_t + temp2[(n + m):0]);
            temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
        end
    end
endfunction

```

```

        f_arith__0=temp4;
        disable Function;
    end
end
end // Function
endfunction
function [(n - 1):0] f_arith__1;
    input [(n_data - 1):0] x0;
    input [(n_data - 1):0] x1;
    input [(n_data - 1):0] x2;
    input [(n_koef - 1):0] koef;
parameter n = ((n_data - 1)-0+1);//constant
parameter m = ((n_koef - 1)-0+1);//constant
    reg [n:0] temp1;
    reg [(n + m):0] temp2;
    reg [(n + m):0] temp3;
    reg [n:0] x0_t;
    reg [n:0] x2_t;
    reg [(n + m):0] x1_t;
    reg [(n - 1):0] temp4;
begin : Function
    begin
        x1_t = {((n + m)-0+1- 0){1'b0}};
        x0_t = {x0[(n - 1)],x0[(n - 1):0]};
        x2_t = {x2[(n - 1)],x2[(n - 1):0]};
        temp1 = (x0_t + x2_t);
        temp2 = (koef * temp1);
        x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]},x1[(n - 1):0]};
        temp2[(n + m)] = temp2[((n + m) - 1)];
        temp3 = (x1_t + temp2[(n + m):0]);
        temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
    begin
        f_arith__1=temp4;
        disable Function;
    end
    end
end // Function
endfunction
function [(n - 1):0] f_arith__2;
    input [(n_data - 1):0] x0;
    input [(n_data - 1):0] x1;
    input [(n_data - 1):0] x2;
    input [(n_koef - 1):0] koef;
parameter n = ((n_data - 1)-0+1);//constant
parameter m = ((n_koef - 1)-0+1);//constant
    reg [n:0] temp1;
    reg [(n + m):0] temp2;
    reg [(n + m):0] temp3;
    reg [n:0] x0_t;
    reg [n:0] x2_t;
    reg [(n + m):0] x1_t;
    reg [(n - 1):0] temp4;
begin : Function
    begin
        x1_t = {((n + m)-0+1- 0){1'b0}};
        x0_t = {x0[(n - 1)],x0[(n - 1):0]};
        x2_t = {x2[(n - 1)],x2[(n - 1):0]};
        temp1 = (x0_t + x2_t);
        temp2 = (koef * temp1);
        x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]},x1[(n - 1):0]};
        temp2[(n + m)] = temp2[((n + m) - 1)];
        temp3 = (x1_t + temp2[(n + m):0]);
        temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
    begin
        f_arith__2=temp4;
        disable Function;
    end
    end
end
end
end

```



```

        end // Function
    endfunction
    function [(n - 1):0] f_arith__3;
        input [(n_data - 1):0] x0;
        input [(n_data - 1):0] x1;
        input [(n_data - 1):0] x2;
        input [(n_koef - 1):0] koef;
        parameter n = ((n_data - 1)-0+1); //constant
        parameter m = ((n_koef - 1)-0+1); //constant
        reg [n:0] temp1;
        reg [(n + m):0] temp2;
        reg [(n + m):0] temp3;
        reg [n:0] x0_t;
        reg [n:0] x2_t;
        reg [(n + m):0] x1_t;
        reg [(n - 1):0] temp4;
        begin : Function
            begin
                x1_t = {(n + m)-0+1- 0}{1'b0}};
                x0_t = {x0[(n - 1)],x0[(n - 1):0]};
                x2_t = {x2[(n - 1)],x2[(n - 1):0]};
                temp1 = (x0_t + x2_t);
                temp2 = (koef * temp1);
                x1_t[(n + m):(m - 2)] = {{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]]},x1[(n - 1):0]};
                temp2[(n + m)] = temp2[((n + m) - 1)];
                temp3 = (x1_t + temp2[(n + m):0]);
                temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
            begin
                f_arith__3=temp4;
            disable Function;
            end
        end
    end // Function
endfunction
endmodule

```

Arith_for_column.sherlock

```

`timescale 1 ns / 10 ps
module arith_for_column ( c0_l, c1_l, c2_l, d0_l, s_l, d1_l, c0_h, c1_h, c2_h,
d0_h, s_h, d1_h, clk, rst, en, c_alu_column, d0_out, s_out, d1_out, h, l );
parameter [31:0] n_data =9;
parameter [31:0] n_koef =8;
parameter n = ((n_data - 1)-0+1);

    input [(n_data - 1):0] c0_l ;
    input [(n_data - 1):0] c1_l ;
    input [(n_data - 1):0] c2_l ;
    input [(n_data - 1):0] d0_l ;
    input [(n_data - 1):0] s_l ;
    input [(n_data - 1):0] d1_l ;
    input [(n_data - 1):0] c0_h ;
    input [(n_data - 1):0] c1_h ;
    input [(n_data - 1):0] c2_h ;
    input [(n_data - 1):0] d0_h ;
    input [(n_data - 1):0] s_h ;
    input [(n_data - 1):0] d1_h ;
    input clk;
    input rst;
    input en;
    input [4:0] c_alu_column ;
    output [(n_data - 1):0] d0_out ;
    output [(n_data - 1):0] s_out ;
    output [(n_data - 1):0] d1_out ;
    output [(n_data - 1):0] h ;
    output [(n_data - 1):0] l ;
    reg [(n_data + n_koef) - 1]:0] l_out__temp;
    parameter [31:0] p1_const=32'b11100101100000110011100110011110;
    parameter [31:0] u1_const=32'b10000011011001000000011010111001;
    parameter [31:0] p2_const=32'b00111000100000011001110101110101;
    parameter [31:0] u2_const=32'b00011100011000100110101010010000;

```

```

parameter [31:0] k0_const=32'b00110100000001100111000010100100;
parameter [31:0] k1_const=32'b00100111010111011001011000010110;
parameter [(n_koef - 1):0] p1={p1_const[31],(~ (p1_const[30:((31 - n_koef) +
1)]) + 1'b1)};
parameter [(n_koef - 1):0] u1={u1_const[31],(~ (u1_const[30:((31 - n_koef) +
1)]) + 1'b1)};
parameter [(n_koef - 1):0] p2=p2_const[31:((31 - n_koef) + 1)];
parameter [(n_koef - 1):0] u2=u2_const[31:((31 - n_koef) + 1)];
parameter [(n_koef - 1):0] k0=k0_const[31:((31 - n_koef) + 1)];
parameter [(n_koef - 1):0] k1=k1_const[31:((31 - n_koef) + 1)];
wire first1;
wire last1;
wire first2;
wire last2;
wire sel;
wire last1_del;
wire last2_del;
reg first1_del;
reg first2_del;
wire [(n_data - 1):0] c0;
wire [(n_data - 1):0] c1;
wire [(n_data - 1):0] c2;
wire [(n_data - 1):0] c2_in;
wire [(n_data - 1):0] d0;
wire [(n_data - 1):0] s0;
wire [(n_data - 1):0] d1;
reg [(n_data - 1):0] c0_1;
reg [(n_data - 1):0] d0_0;
reg [(n_data - 1):0] d0_1;
reg [(n_data - 1):0] s0_0;
reg [(n_data - 1):0] s0_0t;
reg [(n_data - 1):0] d1_1;
reg [(n_data - 1):0] d1_1t0;
reg [(n_data - 1):0] d1_1t1;
wire [(n_data - 1):0] d0_01;
reg [(n_data - 1):0] d0_02;
reg [(n_data - 1):0] s0_1;
wire [(n_data - 1):0] s0_01;
reg [(n_data - 1):0] s0_2;
wire [(n_data - 1):0] d1_10;
reg [(n_data - 1):0] d1_11;
reg [(n_data - 1):0] s1;
wire [(n_data - 1):0] d1_12;
wire [(n_data - 1):0] d0_out;
wire [(n_data - 1):0] s_out;
wire [(n_data - 1):0] d1_out;
reg [(n_data - 1):0] h;
reg [(n_data - 1):0] l;
assign {sel}=c_alu_column[4];
assign {first1}=c_alu_column[3];
assign {last1}=c_alu_column[2];
assign {first2}=c_alu_column[1];
assign {last2}=c_alu_column[0];
shiftrgluts u01(.a(4'b0110), .ce(en), .clk(clk), .data(last1), .q(last1_del));
shiftrgluts u02(.a(4'b1000), .ce(en), .clk(clk), .data(last2), .q(last2_del));
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            first1_del <= 1'b0;
            first2_del <= 1'b0;
        end

    else
        begin
            first1_del <= first1;
            first2_del <= first2;
        end
end
// end always
assign {c0}=(sel == 1'b1) ?
    c0_1

```

```

        : c0_h;
assign {c1}=(sel == 1'b1) ?
        c1_l
        : c1_h;
assign {c2}=(sel == 1'b1) ?
        c2_l
        : c2_h;

assign {d0}=(sel == 1'b1) ?
        d0_l
        : d0_h;
assign {s0}=(sel == 1'b1) ?
        s_l
        : s_h;
    assign {d1}=(sel == 1'b1) ?
        d1_l
        : d1_h;
assign {c2_in}=(last1_del == 1'b1) ?
        c0
        : c2;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            c0_1 <= {((n_data - 1)-0+1- 0){1'b0}};
            d0_0 <= {((n_data - 1)-0+1- 0){1'b0}};
            s0_0t <= {((n_data - 1)-0+1- 0){1'b0}};
            s0_0 <= {((n_data - 1)-0+1- 0){1'b0}};
            d1_1t0 <= {((n_data - 1)-0+1- 0){1'b0}};
            d1_1t1 <= {((n_data - 1)-0+1- 0){1'b0}};
            d1_1 <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                c0_1 <= c0;
                d0_0 <= d0;
                s0_0 <= s0;
                d0_1 <= f_arith__0(c0, c1, c2_in, p1);
                s0_0t <= s0_0;
                s0_0 <= s0_0t;
                d1_1t0 <= d1;
                d1_1t1 <= d1_1t0;
                d1_1 <= d1_1t1;
            end
// end always
    assign {d0_out}=d0_1;
    assign {d0_01}=(first1_del == 1'b1) ?
        d0_1
        : d0_0;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            d0_02 <= {((n_data - 1)-0+1- 0){1'b0}};
            s0_1 <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                d0_02 <= d0_0;
                s0_1 <= f_arith__1(d0_01, c0_1, d0_1, u1);
            end
// end always
    assign {s_out}=s0_1;
    assign {s0_01}=(last2_del == 1'b1) ?
        s0_0
        : s0_1;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        s0_2 <= {((n_data - 1)-0+1- 0){1'b0}};

```

```

else
    if ((en == 1'b1))
        begin
            s0_2 <= s0_0;
            d1_11 <= f_arith__2(s0_0, d0_02, s0_01, p2);
        end
// end always
assign {d1_out}=d1_11;
assign {d1_12}=(first2_del == 1'b1) ?
    d1_11
    : d1_1;
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        s1 <= {((n_data - 1)-0+1- 0){1'b0}};

    else
        if ((en == 1'b1))
            s1 <= f_arith__3(d1_12, s0_2, d1_11, u2);
// end always
always @ (posedge clk or posedge rst ) // begin
    if ((rst == 1'b1))
        begin
            h <= {((n_data - 1)-0+1- 0){1'b0}};
            l <= {((n_data - 1)-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                l_out__temp = (k1 * d1_11);
                h <= {l_out__temp[:(n_data + n_koef) -
1]],l_out__temp[:(n_data + n_koef) - 2):n_koef]};
                l_out__temp = (k0 * s1);
                l <= {l_out__temp[:(n_data + n_koef) -
1]],l_out__temp[:(n_data + n_koef) - 2):n_koef]};
            end
// end always
function [(n - 1):0] f_arith__0;
    input [(n_data - 1):0] x0;
    input [(n_data - 1):0] x1;
    input [(n_data - 1):0] x2;
    input [(n_koef - 1):0] koef;
    parameter n = ((n_data - 1)-0+1); //constant
    parameter m = ((n_koef - 1)-0+1); //constant
    reg [n:0] temp1;
    reg [(n + m):0] temp2;
    reg [(n + m):0] temp3;
    reg [n:0] x0_t;
    reg [n:0] x2_t;
    reg [(n + m):0] x1_t;
    reg [(n - 1):0] temp4;
    begin : Function
        begin
            x1_t = {((n + m)-0+1- 0){1'b0}};
            x0_t = {x0[(n - 1)],x0[(n - 1):0]};
            x2_t = {x2[(n - 1)],x2[(n - 1):0]};
            temp1 = (x0_t + x2_t);
            temp2 = (koef * temp1);
            x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]},x1[(n - 1):0]};
            temp2[(n + m)] = temp2[:(n + m) - 1)];
            temp3 = (x1_t + temp2[(n + m):0]);
            temp4 = {temp3[(n + m)],temp3[:(n + m) - 4):(m - 2)]};
        end
        f_arith__0=temp4;
        disable Function;
    end
end
end // Function
endfunction
function [(n - 1):0] f_arith__1;

```

```

input [(n_data - 1):0] x0;
input [(n_data - 1):0] x1;
input [(n_data - 1):0] x2;
input [(n_koef - 1):0] koef;
parameter n = ((n_data - 1)-0+1);//constant
parameter m = ((n_koef - 1)-0+1);//constant
reg [n:0] temp1;
reg [(n + m):0] temp2;
reg [(n + m):0] temp3;
reg [n:0] x0_t;
reg [n:0] x2_t;
reg [(n + m):0] x1_t;
reg [(n - 1):0] temp4;
begin : Function
begin
x1_t = {((n + m)-0+1- 0){1'b0}};
x0_t = {x0[(n - 1)],x0[(n - 1):0]};
x2_t = {x2[(n - 1)],x2[(n - 1):0]};
temp1 = (x0_t + x2_t);
temp2 = (koef * temp1);
x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]},x1[(n - 1):0]};
temp2[(n + m)] = temp2[((n + m) - 1)];
temp3 = (x1_t + temp2[(n + m):0]);
temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
begin
f_arith__1=temp4;
disable Function;
end
end
end // Function
endfunction
function [(n - 1):0] f_arith__2;
input [(n_data - 1):0] x0;
input [(n_data - 1):0] x1;
input [(n_data - 1):0] x2;
input [(n_koef - 1):0] koef;
parameter n = ((n_data - 1)-0+1);//constant
parameter m = ((n_koef - 1)-0+1);//constant
reg [n:0] temp1;
reg [(n + m):0] temp2;
reg [(n + m):0] temp3;
reg [n:0] x0_t;
reg [n:0] x2_t;
reg [(n + m):0] x1_t;
reg [(n - 1):0] temp4;
begin : Function
begin
x1_t = {((n + m)-0+1- 0){1'b0}};
x0_t = {x0[(n - 1)],x0[(n - 1):0]};
x2_t = {x2[(n - 1)],x2[(n - 1):0]};
temp1 = (x0_t + x2_t);
temp2 = (koef * temp1);
x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]},x1[(n - 1):0]};
temp2[(n + m)] = temp2[((n + m) - 1)];
temp3 = (x1_t + temp2[(n + m):0]);
temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
begin
f_arith__2=temp4;
disable Function;
end
end
end // Function
endfunction
function [(n - 1):0] f_arith__3;
input [(n_data - 1):0] x0;
input [(n_data - 1):0] x1;
input [(n_data - 1):0] x2;
input [(n_koef - 1):0] koef;

```

```

parameter n = ((n_data - 1)-0+1);//constant
parameter m = ((n_koef - 1)-0+1);//constant
    reg [n:0] temp1;
    reg [(n + m):0] temp2;
    reg [(n + m):0] temp3;
    reg [n:0] x0_t;
    reg [n:0] x2_t;
    reg [(n + m):0] x1_t;
    reg [(n - 1):0] temp4;
    begin : Function
        begin
            x1_t = {(n + m)-0+1- 0}{1'b0}};
            x0_t = {x0[(n - 1)],x0[(n - 1):0]};
            x2_t = {x2[(n - 1)],x2[(n - 1):0]};
            temp1 = (x0_t + x2_t);
            temp2 = (koef * temp1);
            x1_t[(n + m):(m - 2)] = {{{x1[(n - 1)],x1[(n - 1)]},x1[(n -
1)]]},x1[(n - 1):0]};
            temp2[(n + m)] = temp2[((n + m) - 1)];
            temp3 = (x1_t + temp2[(n + m):0]);
            temp4 = {temp3[(n + m)],temp3[((n + m) - 4):(m - 2)]};
            begin
                f_arith__3=temp4;
            disable Function;
            end
        end
    end // Function
endfunction
endmodule

```

Control_block.streetlight

```

`timescale 1 ns / 10 ps
module control_block ( clk, rst, en, data_select, done, level, c_alu_row,
enable_l, pop_l, enable_h, pop_h, c_alu_column );
parameter [31:0] size =32;
parameter [31:0] adr_size =4;

    input clk;
    input rst;
    input en;
    output data_select;
    output [3:0] done ;
    output [2:0] level ;
    output [1:0] c_alu_row ;
    output [4:0] enable_l ;
    output [4:0] pop_l ;
    output [4:0] enable_h ;
    output [4:0] pop_h ;
    output [4:0] c_alu_column ;
    reg [adr_size:0] counter1;
    reg [adr_size:0] counter2;
    reg [adr_size:0] tile;
    reg [2:0] level_in;
    wire [4:0] pop_h_in;
    reg sel;
    wire en_c2;
    wire first_r;
    wire last_r;
    wire sel_c;
    wire first1_c;
    wire last1_c;
    wire first2_c;
    wire last2_c;
    wire [0:0] temp1;
    wire [0:0] temp2;
    wire [0:0] temp3;
    wire [0:0] temp4;
    wire enable_l_c0;
    wire enable_l_c1;
    wire enable_l_d0;
    wire enable_l_s;

```

```

wire enable_l_d1;
wire enable_h_c0;
wire enable_h_c1;
wire enable_h_d0;
wire enable_h_s;
wire enable_h_d1;
wire pop_c0;
wire pop_c1;
wire pop_d0;
wire pop_s;
wire pop_d1;
reg done_ll;
reg done_lh;
reg done_hl;
reg done_hh;
wire data_select;
wire [3:0] done;
wire [2:0] level;
wire [1:0] c_alu_row;
wire [4:0] enable_l;
reg [4:0] pop_l;
wire [4:0] enable_h;
wire [4:0] pop_h;
wire [4:0] c_alu_column;
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        counter1 <= {(adr_size-0+1- 0){1'b0}};
    else
        if ((en == 1'b1))
            begin
                if ((counter1 < (tile + 2)))
                    counter1 <= (counter1 + 1);
                else
                    begin
                        counter1 <= {(adr_size-0+1- 0){1'b0}};
                        counter1[1:0] <= 2'b11;
                    end
            end
        end
end// always
assign {en_c2}=(counter1 == (tile + 2)) ?
    1'b1
    : 1'b0;
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        counter2 <= {(adr_size-0+1- 0){1'b0}};
    else
        if (((en == 1'b1) & (en_c2 == 1'b1)))
            begin
                if ((counter2 <= (tile + 1)))
                    counter2 <= (counter2 + 1);
                else
                    counter2 <= {(adr_size-0+1- 0){1'b0}};
            end
        end
end// always
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        sel <= 1'b0;
    else
        if ((en == 1'b1))
            begin
                if (((counter1 == tile) & (counter2 == (tile + 2))))
                    sel <= 1'b1;
                else if (((counter1 == tile) & ((counter2 == tile) |
(counter2 == (tile + 1)))))
                    sel <= 1'b0;
            end
        end
end// always
assign {data_select}=sel;

```

```

always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        begin
            tile <= size;
            level_in <= {(2-0+1- 0){1'b0}};
        end
    else
        if ((en == 1'b1))
            begin
                if (((counter1 == (tile + 2)) & (counter2 == (tile +
2))))
                    begin
                        tile <= {1'b0,tile[adr_size:1]};
                        level_in <= (level_in + 1);
                    end
                end
            end// always

assign {first_r}=(counter1 == 3) ?
    1'b1
    : 1'b0;
assign {last_r}=(counter1 == tile) ?
    1'b1
    : 1'b0;
assign {c_alu_row}={first_r,last_r};

assign {sel_c}=(counter1[0] == 1'b1) ?
    1'b1
    : 1'b0;
    assign {first1_c}=( ((counter2 == 2) & (counter1 >= 11)) | ((counter2 == 3) &
(counter1 < 12))) ?
        1'b1
        : 1'b0;
assign {last1_c}=(counter2 == tile) ?
    1'b1
    : 1'b0;
    assign {first2_c}=( ((counter2 == 4) & (counter1 > 12)) | ((counter2 == 5) &
(counter1 <= 12))) ?
        1'b1
        : 1'b0;
    assign {last2_c}=( (counter2 == (tile + 2)) | (counter2 == 0)) ?
        1'b1
        : 1'b0;
assign {c_alu_column}={{sel_c,first1_c},last1_c},first2_c},last2_c};
assign {temp1}=( (((counter1 >= 9) & (counter2[0] == 1'b0)) | ((counter1 < 9) &
(counter2[0] == 1'b1))) & (counter1[0] == 1'b1));
assign {temp2}=( (((counter1 >= 9) & (counter2[0] == 1'b1)) | ((counter1 < 9) &
(counter2[0] == 1'b0))) & (counter1[0] == 1'b1));
assign {enable_l_c0}=temp1 ?
    1'b1
    : 1'b0;
assign {enable_l_c1}=(temp2 & (counter2 > 0)) ?
    1'b1
    : 1'b0;
assign {enable_l_d0}=(temp1 & (counter2 > 1)) ?
    1'b1
    : 1'b0;
assign {enable_l_s}=(temp1 & (counter2 > 1)) ?
    1'b1
    : 1'b0;
assign {enable_l_d1}=(temp1 & (counter2 > 3)) ?
    1'b1
    : 1'b0;

assign
{enable_l}={{enable_l_c0,enable_l_c1},enable_l_d0},enable_l_s},enable_l_d1};
    assign {temp3}=( (((counter1 >= 8) & (counter2[0] == 1'b0)) | ((counter1 < 8) &
(counter2[0] == 1'b1))) & (counter1[0] == 1'b0));
    assign {temp4}=( (((counter1 >= 8) & (counter2[0] == 1'b1)) | ((counter1 < 8) &
(counter2[0] == 1'b0))) & (counter1[0] == 1'b0));

```



```

assign {enable_h_c0}=temp3 ?
        1'b1
        : 1'b0;
assign {enable_h_c1}=(temp4 & (counter2 > 0)) ?
        1'b1
        : 1'b0;
assign {enable_h_d0}=(temp3 & (counter2 > 1)) ?
        1'b1
        : 1'b0;
assign {enable_h_s}=(temp3 & (counter2 > 1)) ?
        1'b1
        : 1'b0;
assign {enable_h_d1}=(temp3 & (counter2 > 3)) ?
        1'b1
        : 1'b0;

assign
{enable_h}={{{{enable_h_c0,enable_h_c1},enable_h_d0},enable_h_s},enable_h_d1};
assign {pop_c0}=(temp3 & (counter2 > 1)) ?
        1'b1
        : 1'b0;
assign {pop_c1}=(temp3 & (counter2 > 1)) ?
        1'b1
        : 1'b0;
assign {pop_d0}=(temp3 & (counter2 > 3)) ?
        1'b1
        : (((counter1 < 8) & (counter2 == 0)) & (counter1[0] == 1'b0))
?
        1'b1
        : 1'b0;
assign {pop_s}=(temp3 & (counter2 > 3)) ?
        1'b1
        : (((counter1 < 8) & (counter2 == 0)) & (counter1[0] == 1'b0))
?
        1'b1
        : 1'b0;
assign {pop_d1}=(temp3 & (counter2 > 5)) ?
        1'b1
        : (((counter1 < 8) & (counter2 == 0)) & (counter1[0] == 1'b0))
?
        1'b1
        : 1'b0;
assign {pop_h_in}={{{{pop_c0,pop_c1},pop_d0},pop_s},pop_d1};
assign {pop_h}=pop_h_in;
always @ (posedge clk ) begin
    pop_l <= pop_h_in;
end// always
always @ (posedge clk ) begin
    begin
        done_lh <= enable_l_d1;
        done_hh <= enable_h_d1;
        done_ll <= done_lh;
        done_hl <= done_hl;
    end

    end// always

    assign {done}={{{{done_ll,done_lh},done_hl},done_hh}};
endmodule

```

Out_buf. streetlight

```

`timescale 1 ns / 10 ps
module out_buf ( l, h, clk, rst, en, done, ll_buf, lh, hl, hh, done_buf );
parameter [31:0] m =8;
parameter [31:0] m_in =8;
    input [(m_in - 1):0] l ;
    input [(m_in - 1):0] h ;
    input clk;
    input rst;
    input en;
    input [3:0] done ;

```

```

output [(m - 1):0] ll_buf ;
output [(m - 1):0] lh ;
output [(m - 1):0] hl ;
output [(m - 1):0] hh ;
output [3:0] done_buf ;
wire done_ll;
wire done_lh;
wire done_hl;
wire done_hh;
reg [(m - 1):0] ll_buf;
reg [(m - 1):0] lh;
reg [(m - 1):0] hl;
reg [(m - 1):0] hh;
wire [3:0] done_buf;
assign {done_ll}=done[3];
assign {done_lh}=done[2];
assign {done_hl}=done[1];
assign {done_hh}=done[0];
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        ll_buf <= {(m - 1)-0+1- 0}{1'b0}};
    else
        if (((en == 1'b1) & (done_ll == 1'b1)))
            ll_buf <= l[(m - 1):0];
end// always
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        lh <= {(m - 1)-0+1- 0}{1'b0}};
    else
        if (((en == 1'b1) & (done_lh == 1'b1)))
            lh <= h[(m - 1):0];
end// always
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        hl <= {(m - 1)-0+1- 0}{1'b0}};
    else
        if (((en == 1'b1) & (done_hl == 1'b1)))
            hl <= l[(m - 1):0];
end// always
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        hh <= {(m - 1)-0+1- 0}{1'b0}};
    else
        if (((en == 1'b1) & (done_hh == 1'b1)))
            hh <= h[(m - 1):0];
end// always
endmodule

```

Column.streetlight

```

`timescale 1 ns / 10 ps
module column ( clk, rst, in_data, d0_in, d1_in, pop, push, s_in, c0, c1, d0, d1,
s );
parameter [31:0] n_data =9;
parameter [31:0] n_koef =8;
parameter [31:0] tile =8;
parameter [31:0] adr_size =8;
input clk;
input rst;
input [(n_data - 1):0] in_data ;
input [(n_data - 1):0] d0_in ;
input [(n_data - 1):0] d1_in ;
input [4:0] pop ;
input [4:0] push ;
input [(n_data - 1):0] s_in ;
output [(n_data - 1):0] c0 ;
output [(n_data - 1):0] c1 ;
output [(n_data - 1):0] d0 ;
output [(n_data - 1):0] d1 ;
output [(n_data - 1):0] s ;
wire ce;
wire pop_d0;

```

```

wire pop_d1;
wire pop_s;
wire [3:0] a1;
wire [3:0] a2;
wire [3:0] a3;
wire [(n_data - 1):0] c0_out;
wire [(n_data - 1):0] c1_out;
wire [(n_data - 1):0] d0_out;
wire [(n_data - 1):0] d1_out;
wire [(n_data - 1):0] s_out;
reg [(n_data - 1):0] c0;
reg [(n_data - 1):0] c1;
reg [(n_data - 1):0] d0;
reg [(n_data - 1):0] d1;
reg [(n_data - 1):0] s;
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        c0 <= {((n_data - 1)-0+1- 0){1'b0}};
    else
        c0 <= c0_out;
end// always

always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        d0 <= {((n_data - 1)-0+1- 0){1'b0}};
    else
        d0 <= d0_out;
end// always
always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        c1 <= {((n_data - 1)-0+1- 0){1'b0}};
    else
        c1 <= c1_out;
end// always

always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        s <= {((n_data - 1)-0+1- 0){1'b0}};
    else
        s <= s_out;
end// always

always @ (posedge clk or posedge rst ) begin
    if ((rst == 1'b1))
        d1 <= {((n_data - 1)-0+1- 0){1'b0}};
    else
        d1 <= d1_out;
end// always
assign {ce}=1'b1;
assign {a1}=4'b0010;
assign {a2}=4'b0011;
assign {a3}=4'b0100;
fifoclr_cc #(.n(n_data),.size(tile),.adr_size(adr_size))
c0_label(.clock(clk), .fifo_gsr(rst), .read_enable(push[4]),
.write_data_in(in_data[(n_data - 1):0]), .write_enable(pop[4]), .empty_out(/*open
*/), .full_out(/*open */), .read_data_out(c0_out[(n_data - 1):0]));
//Parameter Order: adr_size,n,size
fifoclr_cc #(.n(n_data),.size(tile),.adr_size(adr_size))
c1_label(.clock(clk), .fifo_gsr(rst), .read_enable(push[3]),
.write_data_in(in_data[(n_data - 1):0]), .write_enable(pop[3]), .empty_out(/*open
*/), .full_out(/*open */), .read_data_out(c1_out[(n_data - 1):0]));
//Parameter Order: adr_size,n,size
shiftregluts u3(.a(a1), .ce(ce), .clk(clk), .data(pop[2]), .q(pop_d0));
shiftregluts u4(.a(a2), .ce(ce), .clk(clk), .data(pop[1]), .q(pop_s));
shiftregluts u5(.a(a3), .ce(ce), .clk(clk), .data(pop[0]), .q(pop_d1));
fifoclr_cc #(.n(n_data) /* 16 */ /* 4 */ ) d1_label(.clock(clk),
.fifo_gsr(rst), .read_enable(push[0]), .write_data_in(d1_in[(n_data - 1):0]),
.write_enable(pop_d1), .empty_out(/*open */), .full_out(/*open */),
.read_data_out(d1_out[(n_data - 1):0]));
//Parameter Order: adr_size,n,size

```

```

    fifoctlr_cc #(.n(n_data) /* 16 */ /* 4 */ ) d_label(.clock(clk),
.fifo_gsr(rst), .read_enable(push[2]), .write_data_in(d0_in[(n_data - 1):0]),
.write_enable(pop_d0), .empty_out(/*open */), .full_out(/*open */),
.read_data_out(d0_out[(n_data - 1):0]));
    //Parameter Order: adr_size,n,size
    fifoctlr_cc #(.n(n_data) /* 16 */ /* 4 */ ) s_label(.clock(clk),
.fifo_gsr(rst), .read_enable(push[1]), .write_data_in(s_in[(n_data - 1):0]),
.write_enable(pop_s), .empty_out(/*open */), .full_out(/*open */),
.read_data_out(s_out[(n_data - 1):0]));
    //Parameter Order: adr_size,n,size
Endmodule

                                fifoctlr_cc.streetlight

`timescale 1 ns / 10 ps
module fifoctlr_cc ( clock, read_enable, write_enable, write_data_in, fifo_gsr,
read_data_out, full_out, empty_out );
parameter [31:0] n =9;
parameter [31:0] size =16;
parameter [31:0] adr_size =4;
    //type strength
    parameter strn_x0l__strength=4'd0;
    parameter strn_x0h__strength=4'd1;
    parameter strn_x1l__strength=4'd2;
    parameter strn_x0z__strength=4'd3;
    parameter strn_xz1__strength=4'd4;
    parameter strn_wlh__strength=4'd5;
    parameter strn_wlz__strength=4'd6;
    parameter strn_wzh__strength=4'd7;
    parameter strn_w0h__strength=4'd8;
    parameter strn_w1l__strength=4'd9;
    input clock;
    input read_enable;
    input write_enable;
    input [(n - 1):0] write_data_in ;
    input fifo_gsr;
    output [(n - 1):0] read_data_out ;
    output full_out;
    output empty_out;
    reg full;
    reg empty;
    reg [(adr_size - 1):0] read_addr;
    reg [(adr_size - 1):0] write_addr;
    reg [(adr_size - 1):0] fcounter;
    reg read_allow;
    reg write_allow;
    wire fcnt_allow;
    wire ra_or_fcnt0;
    wire wa_or_fcnt0;
    wire emptyg;
    wire fullg;
    wire [7:0] addra;
    wire [7:0] addrb;
    wire [15:0] doa;
    wire [15:0] dib;
    wire [(n - 1):0] read_data_out;
    wire full_out;
    wire empty_out;
    assign {full_out}=full;
    assign {empty_out}=empty;
    generate
        if ((n == 9) & (adr_size == 4))begin//generate if

            assign {addra}={4'b0000,read_addr};
            assign {addrb}={4'b0000,write_addr};
            assign {dib}={7'b0000000,write_data_in};
            assign {read_data_out}=doa[8:0];
        end//generate if
    endgenerate
    generate
        if ((n == 9) & (adr_size == 5))begin//generate if

```

```

        assign {addra}={3'b000,read_addr};

        assign {addrb}={3'b000,write_addr};

        assign {dib}={7'b0000000,write_data_in};

        assign {read_data_out}=doa[8:0];

    end//generate if
endgenerate
generate
    if ((n == 9) & (adr_size == 6))begin//generate if

        assign {addra}={2'b00,read_addr};
        assign {addrb}={2'b00,write_addr};
        assign {dib}={7'b0000000,write_data_in};
        assign {read_data_out}=doa[8:0];
    end//generate if
endgenerate
generate
    if ((n == 9) & (adr_size == 7))begin//generate if
        assign {addra}={1'b0,read_addr};
        assign {addrb}={1'b0,write_addr};
        assign {dib}={7'b0000000,write_data_in};
        assign {read_data_out}=doa[8:0];
    end//generate if
endgenerate
generate
    if ((n == 9) & (adr_size == 7))begin//generate if
        assign {addra}=read_addr;
        assign {addrb}=write_addr;
        assign {dib}={7'b0000000,write_data_in};
        assign {read_data_out}=doa[8:0];
    end//generate if
endgenerate
    RAMB4_S16_S16 bram1(.DOA(doa), .DOB(/*open */), .ADDRA(addra), .ADDRB(addrb),
.CLKA(clock), .CLKB(clock), .DIA(dib), .DIB(dib), .ENA(read_allow),
.ENB(write_allow), .RSTA(1'b0), .RSTB(1'b0), .WEA(1'b0), .WEB(write_allow));
    always @ (posedge clock or posedge fifo_gsr ) // begin
        if ((fifo_gsr == 1'b1))
            read_allow <= 1'b0;
        else
            read_allow <= (read_enable & ~ ((~ (or_reduce__0(fcounter[(adr_size
- 1):1])) & ~ (write_allow))));
        // end always
    always @ (posedge clock or posedge fifo_gsr ) // begin
        if ((fifo_gsr == 1'b1))
            write_allow <= 1'b0;

        else
            write_allow <= (write_enable & ~ ((and_reduce__1(fcounter[(adr_size
- 1):1]) & ~ (read_allow))));

        // end always
    assign {fcnt_allow}=(write_allow ^ read_allow);
    assign {ra_or_fcnt0}=(read_allow | ~ (fcounter[0]));
    assign {emptyg}=(~ (or_reduce__2(fcounter[(adr_size - 1):1])) & ra_or_fcnt0) &
~ (write_allow));
    always @ (posedge clock or posedge fifo_gsr ) // begin
        if ((fifo_gsr == 1'b1))
            empty <= 1'b1;

        else
            empty <= emptyg;
        // end always
    assign {wa_or_fcnt0}=(write_allow | fcounter[0]);
    assign {fullg}=(~ (and_reduce__3(fcounter[(adr_size - 1):1]) & wa_or_fcnt0) & ~
(read_allow));

```

```

always @ (posedge clock or posedge fifo_gsr ) // begin
    if ((fifo_gsr == 1'b1))
        full <= 1'b1;
    else
        full <= fullg;
// end always
always @ (posedge clock or posedge fifo_gsr ) // begin
    if ((fifo_gsr == 1'b1))
        read_addr <= {((adr_size - 1)-0+1- 0){1'b0}};
    else
        if ((read_allow == 1'b1))
            read_addr <= (read_addr + 1'b1);
// end always
always @ (posedge clock or posedge fifo_gsr ) // begin
    if ((fifo_gsr == 1'b1))
        write_addr <= {((adr_size - 1)-0+1- 0){1'b0}};
    else
        if ((write_allow == 1'b1))
            write_addr <= (write_addr + 1'b1);
// end always
always @ (posedge clock or posedge fifo_gsr ) // begin
    if ((fifo_gsr == 1'b1))
        fcounter <= {((adr_size - 1)-0+1- 0){1'b0}};
    else

        if ((fcnt_allow == 1'b1))
            begin
                if ((read_allow == 1'b0))
                    fcounter <= (fcounter + 1'b1);
                else
                    fcounter <= (fcounter - 1'b1);
            end
// end always
function [0:0] or_reduce__0;
    input [(adr_size - 1):1] arg;
    reg result;
    begin : Function
        begin
            result = 1'b0;
            begin :Block_Name_1
                integer i;
                for (i=(adr_size - 1);i>=1;i=i-1) begin
                    result = (result | arg[i]);
                end //for
            end //end Block
        begin
            or_reduce__0=result;
            disable Function;
        end
    end
end // Function
endfunction
function [0:0] and_reduce__1;
    input [(adr_size - 1):1] arg;
    reg result;
    begin : Function
        begin
            result = 1'b1;
            begin :Block_Name_2
                integer i;
                for (i=(adr_size - 1);i>=1;i=i-1) begin
                    result = (result & arg[i]);
                end //for
            end //end Block
        begin
            and_reduce__1=result;
            disable Function;
        end
    end
end

```

```

        end
    end // Function
endfunction
function [0:0] or_reduce__2;
    input [(adr_size - 1):1] arg;
    reg result;
    begin : Function
        begin
            result = 1'b0;
            begin :Block_Name_3
                integer i;
                for (i=(adr_size - 1);i>=1;i=i-1) begin
                    result = (result | arg[i]);
                end //for
            end //end Block
        begin
            or_reduce__2=result;
            disable Function;
        end
    end
end // Function
endfunction
function [0:0] and_reduce__3;
    input [(adr_size - 1):1] arg;
    reg result;
    begin : Function
        begin
            result = 1'b1;
            begin :Block_Name_4
                integer i;
                for (i=(adr_size - 1);i>=1;i=i-1) begin
                    result = (result & arg[i]);
                end //for
            end //end Block
        begin
            and_reduce__3=result;
            disable Function;
        end
    end
end // Function
endfunction
endmodule

                                shiftregluts. streetlight

`timescale 1 ns / 10 ps
module shiftregluts ( clk, data, ce, a, q );
    input clk;
    input data;
    input ce;
    input [3:0] a ;
    output q;
    parameter depth_width = 16;//constant
    reg [0:(depth_width - 1)] srl_sig ;
    wire q;
    always @ (posedge clk ) // begin

        if ((ce == 1'b1))
            srl_sig <=
{data,srl_sig[0],srl_sig[1],srl_sig[2],srl_sig[3],srl_sig[4],srl_sig[5],srl_sig[6]
,srl_sig[7],srl_sig[8],srl_sig[9],srl_sig[10],srl_sig[11],srl_sig[12],srl_sig[13],
srl_sig[14]};
    // end always
    assign {q}=srl_sig[a];
endmodule

```

E2. Інфраструктура верифікації streetlight and sherlock

```

                                paket11.sv
`timescale 1 ns / 10 ps
package Data_pack1;
class Etalon1;
// int X[8][8];
  int temp, i, j;
  int tmp;
  const int tile = 256;
  const real a1 = -1.586134342059924;
  const real a2 = -0.052980118572961;
  const real a3 = 0.882911075530934;
  const real a4 = 0.443506852043971;
  const real K = 1.230174104914001;
  const real K0 = 1/K;
  const real K1 = K/2;
  task dwt_etalon(input int X[64],output int HH1[16],output int
HL1[16],output int LH1[16],output int LL1[16]);
    int Z[8][8], T[8][8], L1[8][4], H1[8][4], HH[4][4], HL[4][4], LH[4][4],
LL[4][4];
    /*size_block=size(X);
    X_W=[X(2) X X(size_block(2)-1)];*/
    Z[0]=X[0:7];
    Z[1]=X[8:15];
    Z[2]=X[16:23];
    Z[3]=X[24:31];
    Z[4]=X[32:39];
    Z[5]=X[40:47];
    Z[6]=X[48:55];
    Z[7]=X[56:63];
    for(int i=0; i<8;i++)
      for(int j=0; j<8;j++)
        begin
          if (j==0)
            Z[i][j-1]=Z[i][j+1];
          else if (j==7)
            Z[i][j+1]=Z[i][j-1];
            Z[i][j]=Z[i][j]+(Z[i][j-1]+Z[i][j+1])*a1;
          end
        for(int i=0; i<8;i++)
          for(int j=0; j<8;j=j+2)
            begin
              if (j==0)
                Z[i][j-1]=Z[i][j+1];
                if (j==15)
                  Z[i][j+1]=Z[i][j-1];
                  Z[i][j]=Z[i][j]+(Z[i][j-1]+Z[i][j+1])*a2;
                end
              for(int i=0; i<8;i++)
                for(int j=0; j<8;j=j+2)
                  begin
                    if (j==0)
                      Z[i][j-1]=Z[i][j+1];
                      if (j==15)
                        Z[i][j+1]=Z[i][j-1];
                        Z[i][j]=Z[i][j]+(Z[i][j-1]+Z[i][j+1])*a3;
                      end
                    for(int i=0; i<8;i++)
                      for(int j=0; j<8;j=j+2)
                        begin
                          L1[i][j/2]=Z[i][j];
                        end
                      end
                    /*for(int i=0; i<16;i+2)
                      for(int j=0; j<8;j++)
                        LL[i/2][j]=L[i][j];*/

```



```

    for(int i = 0; i<8;++i)
    for (int j = 0; j<4;++j)
    begin
        tmp = L1[i][j];
        L1[i][j] = L1[j][i];
        L1[j][i] = tmp;
end
    for(int j=0; j<8;j++)
begin
    temp=0;
    for(int k=0;k<8;k++)
        temp=temp+L1[k][j/2];
        if (temp>=0)
            X[j*8+i]=temp/256 + temp/128-(temp/256)*2;
        else begin
            temp=-temp;
            temp=temp/256 + temp/128-(temp/256)*2;
            X[j*8+i]=-temp;
        end
    for(int i=0; i<8;i++)
        for(int j=0; j<8;j=j+2)
        begin
            if (j==0)
                Z[i][j-1]=Z[i][j+1];
                if (j==15)
                    Z[i][j+1]=Z[i][j-1];
                Z[i][j]=Z[i][j]+(Z[i][j-1]+Z[i][j+1])*a4;
            end
        for(int i=0; i<8;i++)
            for(int j=0; j<8;j=j+2)
                H1[i][j/2]=Z[i][j];

            for(int i=0; i<8;i++)
            for(int j=0; j<8;j++)
            begin
                temp=0;
                for(int k=0;k<8;k++)
                    temp=temp+H1[k][j/2];
                    if (temp>=0)
                        X[j*8+i]=temp/256 + temp/128-(temp/256)*2;
                    else begin
                        temp=-temp;
                        temp=temp/256 + temp/128-(temp/256)*2;
                        X[j*8+i]=-temp;
                    end
            end
            for(int i = 0; i<8;++i)
            for (int j = 0; j<4;++j)
            begin
                tmp = H1[i][j];
                H1[i][j] = H1[j][i];
                H1[j][i] = tmp;
end
        for(int i=0; i<8;i++)
        for(int j=0; j<4;j++)
        begin
            if (j==0)
                L1[i][j-1]=L1[i][j+1];
            else if (j==3)
                L1[i][j+1]=L1[i][j-1];
                L1[i][j]=L1[i][j]+(L1[i][j-1]+L1[i][j+1])*a1;
            end
        for(int i=0; i<8;i++)
        for(int j=0; j<4;j=j+2)
        begin
            if (j==0)

```

```

        L1[i][j-1]=L1[i][j+1];
        if (j==15)
            L1[i][j+1]=L1[i][j-1];
        L1[i][j]=L1[i][j]+(L1[i][j-1]+L1[i][j+1])*a2;
    end
for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
begin
    if (j==0)
        L1[i][j-1]=L1[i][j+1];
        if (j==15)
            L1[i][j+1]=L1[i][j-1];
        L1[i][j]=L1[i][j]+(L1[i][j-1]+L1[i][j+1])*a3;
    end
for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
begin
    LL[i/2][j/2]=L1[i][j];
end
/*for(int i=0; i<16;i+2)
for(int j=0; j<8;j++)
    LL[i/2][j]=L1[i][j];*/
for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
begin
    if (j==0)
        L1[i][j-1]=L1[i][j+1];
        if (j==15)
            L1[i][j+1]=L1[i][j-1];
        L1[i][j]=L1[i][j]+(L1[i][j-1]+L1[i][j+1])*a4;
    end
for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
    LH[i/2][j/2]=L1[i][j];
for(int i=0; i<8;i++)
for(int j=0; j<4;j++)
begin
    if (j==0)
        H1[i][j-1]=H1[i][j+1];
    else if (j==3)
        H1[i][j+1]=H1[i][j-1];
        H1[i][j]=H1[i][j]+(H1[i][j-1]+H1[i][j+1])*a1;
    end
end
for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
begin
    if (j==0)
        H1[i][j-1]=H1[i][j+1];
        if (j==15)
            H1[i][j+1]=H1[i][j-1];
        H1[i][j]=H1[i][j]+(H1[i][j-1]+H1[i][j+1])*a2;
    end

for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
begin
    if (j==0)
        H1[i][j-1]=H1[i][j+1];
        if (j==15)
            H1[i][j+1]=H1[i][j-1];
        H1[i][j]=H1[i][j]+(H1[i][j-1]+H1[i][j+1])*a3;
    end
for(int i=0; i<8;i++)
for(int j=0; j<4;j=j+2)
begin
    HL[i/2][j/2]=H1[i][j];

```

```

end
/*for(int i=0; i<16;i+2)
  for(int j=0; j<8;j++)
    LL[i/2][j/2]=L[i][j];*/
for(int i=0; i<8;i++)
  for(int j=0; j<4;j=j+2)
    begin
      if (j==0)
        H1[i][j-1]=H1[i][j+1];
        if (j==15)
          H1[i][j+1]=H1[i][j-1];
        H1[i][j]=H1[i][j]+(H1[i][j-1]+H1[i][j+1])*a4;
      end
for(int i=0; i<8;i++)
  for(int j=0; j<4;j=j+2)
    HH[i/2][j/2]=H1[i][j];
for(int i=0; i<4;i++)
  for(int j=0; j<4;j++)
    begin
      temp=0;
      for(int k=0;k<4;k++)
        temp=HH[i][j];
      if (temp>=0)
        HHL[j*4]=temp/256 + temp/128-(temp/256)*2;
      else begin
        temp=-temp;
        temp=temp/256 + temp/128-(temp/256)*2;
        HHL[j*4]=-temp;
      end
    end
  for(int i=0; i<4;i++)
  for(int j=0; j<4;j++)
    begin
      temp=0;
      for(int k=0;k<4;k++)
        temp=LH[i][j];
      if (temp>=0)
        LHL[j*4]=temp/256 + temp/128-(temp/256)*2;
      else begin
        temp=-temp;
        temp=temp/256 + temp/128-(temp/256)*2;
        LHL[j*4]=-temp;
      end
    end
  for(int i=0; i<4;i++)
  for(int j=0; j<4;j++)
    begin
      temp=0;
      for(int k=0;k<4;k++)
        temp=HL[i][j];
      if (temp>=0)
        HLL[j*4]=temp/256 + temp/128-(temp/256)*2;
      else begin
        temp=-temp;
        temp=temp/256 + temp/128-(temp/256)*2;
        HLL[j*4]=-temp;
      end
    end
  for(int i=0; i<4;i++)
  for(int j=0; j<4;j++)
    begin
      temp=0;
      for(int k=0;k<4;k++)
        temp=LL[i][j];
      if (temp>=0)
        LLL[j*4]=temp/256 + temp/128-(temp/256)*2;
      else begin

```

```

        temp=-temp;
        temp=temp/256 + temp/128-(temp/256)*2;
        LL1[j*4]=-temp;
    end
end
endtask
endclass
endpackage

                                Interface1.sv

`timescale 1 ns / 10 ps
interface dipl;
logic CLK, RST, en;
parameter [31:0] n_data =9;
parameter [31:0] n_koef =8;
parameter [31:0] tile =64;
parameter [31:0] adr_size =8;
logic [(n_data - 1):0] data_in;
parameter[2:0] t=tile;
logic net2903;
logic select_data;
logic [3:0] done_buf;
logic [(n_data - 1):0] ll_buf;
logic [(n_data - 1):0] ll_in;
logic [3:0] done;
    logic [(n_data - 1):0] hh;
    logic [(n_data - 1):0] hl;
    logic [(n_data - 1):0] lh;
    logic [(n_data - 1):0] ll;
logic [2:0] level;
logic [2:0] tile_size;
    task DataSend(input int A[64] );
    foreach(A[i])
    begin
        @(posedge CLK);
        data_in=A[i];
    end
    endtask
        task DataRead1( output int B1[32]);
        foreach(B1[i])
        begin
            #4;
            @(posedge CLK )
            B1[i]=hh[i];
            // $cast(B[i],signed'(dct_2d));
        end
        endtask
            task DataRead2( output int B2[32]);
            foreach(B2[i])
            begin
                #5;
                @(posedge CLK )
                B2[i]=ll[i];
                // $cast(B2[i],signed'(dct_2d));
            end
            endtask
                modport TLM(import task DataSend(),task DataRead1(), task
DataRead2(),
                                output RST);
endinterface

                                Test1.sv

`timescale 1 ns / 10 ps
module Tester1(interface i);
import Data_pack1::*;
Etalon1 test_data1;
int k,m;
bit ER;
const int A[64]='{1,7,5,7,12,4,1,1,

```

```

        1,1,1,1,1,1,1,1,
        1,1,1,1,1,1,1,1,
        1,1,1,1,1,1,1,1,
        2,2,2,2,2,2,2,2,
        2,2,2,2,2,2,2,2,
        1,3,1,1,1,1,1,1,
        11,3,4,5,6,7,8,1};
int RES1[32], RES2[32],RES_et1[16],RES_et2[16],RES_et3[16],RES_et4[16];

initial begin:main
    i.RST=1'b1;
#64 i.RST=1'b0;
    fork
        i.DataSend(A);
        i.DataRead1(RES1);
        i.DataRead2(RES2);
    join
        test_data1=new;
        test_data1.dwt_etalon(A,RES_et1,RES_et2,RES_et3,RES_et4);
        foreach(RES_et1[k]) begin
            equal: assert (RES_et1[k]/2==RES1[k]/2)
            foreach(RES_et2[m]) begin
                equal: assert (RES_et2[m]/2==RES2[m]/2)
                $display("%m Good!!!!");
            else
                $warning("%m      Karaul!!!!      Res=%d,Res_et=%d",      RES_et1[k],
RES1[k],RES_et1[k], RES1[k]);
            end
        end
    end
endmodule
gener.sv
`timescale 1 ns / 10 ps
module Clock1 (interface Bus);
    always begin // clock
        #50 Bus.CLK = 0;
        #50 Bus.CLK = 1;
    end
endmodule : Clock1

gentest.sv

package Data;
    class Generate;
    rand byte K[64];
    endclas
endpackage

```