УДК 004.75

Шкарупило Вадим Вікторович,
к.т.н., доцент, доцент кафедри комп'ютерних систем і мереж, Національний університет біоресурсів і природокористування України
shkarupylo.vadym@gmail.com
Кудерметов Равіль Камілович,
к.т.н., доцент, завідувач кафедри комп'ютерних систем та мереж, Запорізький національний технічний університет України
kudermetov@gmail.com
Тіменко Артур Валентинович,
асистент кафедри комп'ютерних систем та мереж, Запорізький національний технічний університет України
timenko.artur@gmail.com
Польська Ольга Володимирівна,
старший викладач кафедри комп'ютерних систем та мереж, Запорізький національний технічний університет України
ol.polsk@gmail.com

ПІДХІД ДО ПЕРЕВІРКИ ВІРТУАЛЬНОЇ ТОПОЛОГІЇ ПРОГРАМНО-КОНФІГУРОВАНОЇ МЕРЕЖІ

Анотація. *У наш час концепція програмно-конфігурованих мереж широко розглядається у якості однієї з основоположних технологій подальшого розвитку Інтернету – у вигляді Інтернету речей. При цьому, спираючись на специфіку названих мереж, оперують поняттям віртуальної топології мережі. Це дає змогу гнучким та оперативним чином реагувати на зміну вимог до розподілених програмних систем, побудованих на основі мережі. При цьому постає проблема перевірки узгодженості компонентів системи. З цією метою у статті проаонується підхід до перевірки віртуальної топології програмно-конфігурованої мережі. Запропонований підхід ґрунтується на використанні темпоральної логіки дій (Temporal Logic of Actions, TLA). В основі запропонованого підходу – використання концепції «дії» (action). Для представлення специфіки запропонованого підходу використано структуру Кріпке. Відмінна риса запропонованого підходу – гнучкість реконфігурування специфікації (формальної моделі) системи з віртуальною топологією. Перевірку топології запропоновано виконувати шляхом автоматизованої формальної верифікації методом перевірки на моделі. З метою поліпшення автоматизації, застосовано реалізацію методу перевірки на моделі TLA Checker (TLC), що ґрунтується на обході простору станів системи переходів в основі формальної моделі методом обходу в ширину (Breadth-first Search, BFS).*

Ключові слова: *ad-hoc, Quality of Services, верифікація, віртуальна топологія, Інтернет речей, перевірка на моделі, програмно-конфігурована мережа, специфікація.*

Протягом еволюційного розвитку мережних технологій пропонувалися різноманітні пріоритети, призначені до врахування при побудові відповідних систем. Прикладом таких пріоритетів було, наприклад, залучення вже існуючої інфраструктури – телефонні мережі. У наш час, враховуючи той фактор, що мережею циркулює трафік різного характеру (голосові, відеодані тощо), а до мережних транзакцій залучені різноманітні зацікавлені сторони (організації, користувачі), що передбачає різноманітні сценарії взаємодії, одним з першорядних питань постає забезпечення задовільного рівня якості надання послуг (Quality of Services, QoS). Безпосередньо мережа при цьому є системою з динамічною природою, коли сценарії обміну даними між учасниками взаємодії мережею носять ad-hoc-характер. Окреслені позиції дає змогу реалізувати використання концепції програмно-конфігурованих мереж (Software-defined Networks, SDN).

У наш час принципи побудови SDN-мереж широко розглядаються у якості основоположних засад та платформи для реалізації концепції Інтернету речей (Internet of Things, IoT), що ґрунтується на ідеї застосування програмної складової забезпечення централізованого координування мережі – контролера (controller) [1]. Наприклад, з точки зору забезпечення механізму автоматизованої взаємодії різноманітних пристроїв (Machine-

to-Machine, M2M) між собою, існує потреба у створення масштабованих, реконфігурованих, а також придатних до здійснення централізованого контролю, рішень, які б оперативним чином відповідали стрімким змінам (ad-hoc) вимог оточуючого середовища, виражених у зміні бізнес-процесів. Це може бути досягнуто шляхом впровадження основоположних засад SDN-мереж (Software-defined Networks, SDN). Відповідними представниками є, зокрема, наступні складові [2]:

– програмованість – за рахунок виділення рівнів керування і даних;

– відкритість – впровадження відкритих стандартів;

– гнучкість – придатність до динамічного реконфігурування, контролю, обслуговування.

Спираючись на трирівневу архітектуру SDN-мереж [3], що включає верхній рівень додатків (Application Layer), середній рівень керування (Control Layer) та нижній рівень даних/маршрутизації (Data/Forwarding Layer), можна виділити три наступні складові побудови топології SDN-мереж:

– контролер – програмна система, на яку покладено функції координування компонентів SDN-мережі;

– комутатори (switches), що можуть бути реалізовані як програмно, так і апаратно, та призначені для пересилання трафіку мережею;

– хости (hosts) – кінцеві вузли топології.

Вищеназвані концепти у повній мірі представлені у поширеному емуляторі SDN-мереж – Mininet [4].

З позиції ad-hoc-сценаріїв взаємодії, розглянемо топологію SDN-мережі як динамічну структуру на основі контролера, множини комутаторів та множини хостів. Для забезпечення успішного функціонування окресленої системи необхідно упевнитися, що компоненти системи взаємодіють між собою узгоджено. Цього можна досягти шляхом залучення до процесу проектування названої системи математичного апарату сімейства методів перевірки на моделі, що дозволить відповісти на питання щодо узгодженості взаємодії компонентів системи шляхом автоматизованої формальної верифікації – перевірки на моделі (специфікації). У якості засобу створення відповідної моделі, як правило, використовується певна темпоральна логіка.

У даному випадку обрано темпоральну логіку дій (Temporal Logic of Actions, TLA) та відповідний формалізм TLA+ [5]. Даний крок базується на наступних факторах:

– формалізм TLA+, завдяки математичній строгості, дозволяє компактним та однозначним чином представити властивості системи, призначені до автоматизованої перевірки;

– використання концепту дії (action) дозволить зручним чином створювати модульні та придатні до оперативного реконфігурування формальні моделі, що дозволить адекватним чином реагувати на зміну ad-hoc-сценаріїв функціонування системи. Дію можна розглядати як специфікацію переходу між двома станами системи – поточним і наступним. При цьому, розглядаючи SDN-сумісні комутатори як основоположні складові маршрутів просування пакетів від хоста-джерела – до цільового хоста, специфікації безпосередньо маршрутів можна представляти ланцюжками специфікацій дій.

У загальному випадку, шлях, сформований комутаторами, який подолали певні пакети, може бути іншим для інших пакетів. Більше того, зміна політик і сценаріїв функціонування IoT-системи може супроводжуватися зміною топології SDN-мережі в її основі. При цьому, беручи до уваги програмованість програмно-конфігурованих мереж, доречно оперувати поняттям віртуальної топології мережі поверх фізичної топології, враховуючи також підтримку заданого рівня QoS-характеристик. Отже, враховуючи ad-hoc-природу сценаріїв функціонування відповідних IoT-систем, постає проблема оперативного реконфігурування топології SDN-мережі в основі IoT-системи згідно актуальних вимог, з позиції підтримки належного рівня QoS-характеристик. Окреслена проблема набуває

особливої актуальності з точки зору інтенсивних сценаріїв використання IoT-системи, коли перелік доступних ресурсів (обчислювальних, збереження даних, програмних) є обмеженим.

У якості засобу, що сприяє вирішенню названої проблеми, пропонується підхід, що полягає у здійсненні автоматизованої формальної верифікації при проектуванні системи. У зв'язку із цим формулюється нижченаведена мета роботи.

Мета роботи: сприяти вирішенню проблеми оперативного реконфігурування топології SDN-мережі в основі IoT-системи згідно заданих вимог.

Для досягнення сформульованої мети в роботі пропонується підхід, що ґрунтується на використанні темпоральної логіки TLA. Названа темпоральна логіка, а також відповідна реалізація методу перевірки на моделі TLC (TLA Checker), широко використовуються в різноманітних предметних областях, наприклад, при перевірці проектних рішень веб-сервісів Amazon (Amazon Web Services, AWS) [6].

Метод TLC дозволяє здійснювати перевірку формальних моделей в автоматизованому режимі та допускає два шляхи застосування: за рахунок обходу станів системи переходів в основі формальної моделі методом обходу в ширину (Breadth-first Search, BFS) та в глибину (Depth-first search, DFS). Було показано, що застосування DFS-реалізації потребує суттєво менших часових витрат, у порівнянні з альтернативною BFS-реалізацією, однак, при цьому потрібно зазначати глибину обходу [7]. Цей фактор можна розцінювати як істотну перепону з позиції автоматизації. У зв'язку із цим, в запропонованому підході використовуватиметься BFS-реалізація методу TLC.

Постановка задачі. Розглянемо топологію деревовидної структури (рис. 1).

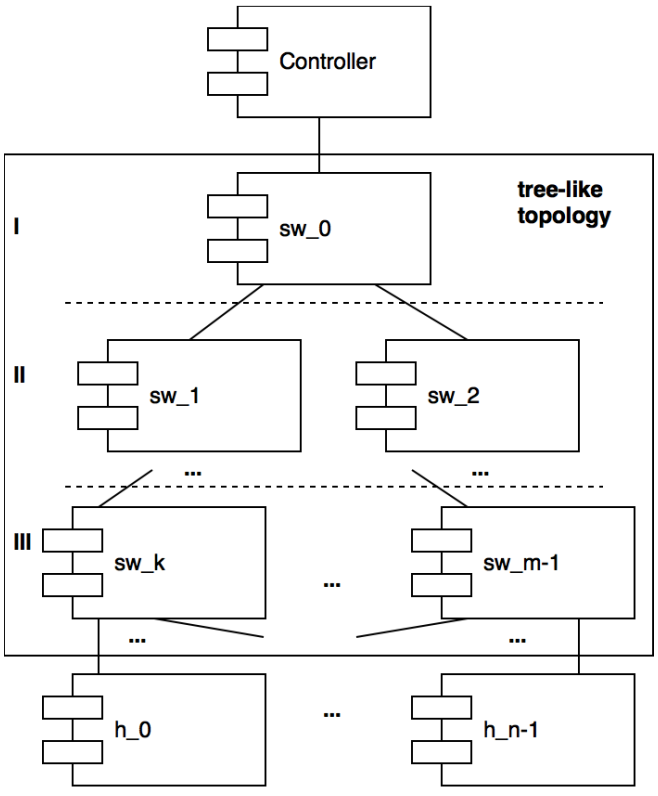

Рис.1. SDN-мережа деревовидної топології

На рис. 1 подано структуру SDN-мережі деревовидної топології, де «Controller» – спеціалізоване програмне забезпечення, розгорнуте на певному вузлі мережі та призначене для централізованого координування просування пакетів мережею; sw_0 – SDN-сумісний комутатор верхнього рівня (за аналогією організації комутаторів у ЦОД – центрах обробки даних); $sw_1, sw_2, ..., sw_{k-1} (k \in N)$ – SDN-сумісні комутатори проміжного рівня, що, власне,

формують деревовидну структуру; $sw_k, sw_{k+1},..., sw_{m-1}(m \in N)$ – термінальні комутатори, до яких безпосередньо підключаються хости $h_0, h_1...,h_{n-1}(n \in N)$.

Отже, множину комутаторів можна представити наступним чином:

$$SW = \{sw_0\} \cup SW' \cup SW^*, \tag{1}$$

де – $SW' = \{sw_1, sw_2,..., sw_{k-1}\}$, $SW^* = \{sw_k, sw_{k+1},..., sw_{m-1}\}$.

Для створення формальної моделі топології потрібно також визначити множину хостів:

$$H = \{h_0, h_1,..., h_{n-1}\}. \tag{2}$$

Аналітичне представлення формальної моделі топології задамо структурою Кріпке на множині атомарних виразів AP [8]:

$$M = \langle S, \{s_0\}, R, L \rangle, \tag{3}$$

де S – кінцева множина станів системи переходів; $s_0 \in S$ – початковий стан; $R \subseteq S^2$ – множина переходів між станами; $L: S \to 2^{AP}$ – функція розмітки станів.

Множину змінних станів сформуємо на основі множин (1) і (2), перепозначивши відповідні елементи:

$$V = SW \cup H : V = \{v_0, v_1,..., v_i,..., v_{m+n-1}\}, \tag{4}$$

де $v_i \in V$ – елемент множини змінних станів, що може представляти як комутатор, так і хост.

Для формування множини AP попередньо сформуємо множину значень, які можуть приймати змінні станів:

$$D = \{d_0, d_1\}, \tag{5}$$

де $d_0 = 0 \in D$ означає, що пакет даних ще не досяг вузла, представленого змінною $v_i \in V$, $d_1 = 1 \in D$ – пакет досяг вузла $v_i \in V$.

Множину AP сформуємо наступним чином:

$$AP = V \times D. \tag{6}$$

Для постановки задачі перевірки узгодженості взаємодії компонентів системи представимо результуючу формальну модель у вигляді темпоральної формули φ. Тоді вирішувана в роботі задача полягає в наступному:

$$M, s \models \varphi, \tag{7}$$

що означає, що формула φ має виконуватися $\forall s \in S$ структури (3).

Отже, проведення перевірки віртуальної топології SDN-мережі має на меті з'ясувати, що результуюча темпоральна формула, задана формальною моделлю, виконується $\forall s \in S$ структури (3).

Опис підходу. Запропонований підхід розглянемо на прикладі. Нехай деревовидна структура (рис. 1) сформована на основі наступної множини SDN-сумісних комутаторів (1): $SW = \{sw_0, sw_1, ..., sw_6\}$. Учасниками взаємодії є два хости (2): $H = \{h_0, h_1\}$. Отже, маємо наступну множину змінних станів (4): $V = \{v_0, v_1, ..., v_8\}$, де $v_0 \in V$ представляє $h_0 \in H$, $v_1 \in V ... v_7 \in V$ – $sw_0 \in SW ... sw_6 \in SW$, відповідно, а $v_8 \in V$ – $h_1 \in H$.

Запропонований підхід полягає в наступному.

На основі сформованої множини V задамо специфікацію початкового стану системи переходів, яку представимо як кон'юнкцію:

$$Init \equiv (v_0 = 0) \wedge (v_1 = 0) \wedge ... \wedge (v_8 = 0), \tag{8}$$

де вираз *Init* сформовано на основі відповідної розмітки початкового стану $L(s_0) = \{(v_0,0),(v_1,0),...,(v_8,0)\}$.

Для досягнення наступних станів скористаємося концепцією «події», яку формалізуємо як імплікацію, модифіковану темпоральним оператором X (neXt) [7]:

$$Event_i \equiv \neg(v_i = 0) \vee X(v_i = 1), \tag{9}$$

що означає, що в наступний момент модельного часу змінна $v_i \in V$ змінить своє значення. У даному випадку розмітку $L(s)$ певного поточного стану $s \in S$ можна розглядати у якості передумови до виникнення події (3), а розмітку наступного стану $s' = R(s) \in S$ – наслідку виникнення події. Передумову і наслідок зручно представляти у вигляді симетричної різниці:

$$L(s) \Delta L(s') = \{(v_i,0),(v_i,1)\}. \tag{10}$$

Перехід $(s, s') \in R$ розглядається у якості базового блоку побудови формальної моделі.

Для формування специфікації переходу за основу береться вираз (9). При цьому враховуються також наступні позиції:

– значення змінних множини $V' = V \setminus \{v_i\}$ при переході мають залишатися незмінними;

– специфікація переходу має включати в собі специфікацію передумови до здійснення переходу.

Специфікація переходу, що враховує вищезазначені зауваження, будується на основі конструкції IF-THEN-ELSE та має наступний вигляд:

$$\begin{aligned} Action_i \equiv (v_i' = IF\ (Condition)\ THEN\ \neg v_i\ ELSE\ v_i) \wedge \\ \wedge UNCHANGED(v_0, v_1, ..., v_{i-1}, v_{i+1}, ..., v_{m+n-1}) \end{aligned}, \tag{11}$$

де оператор *UNCHANGED* враховує перше зауваження, а змінна з модифікатором $v_i' \in V'$ представляє змінну станів у наступний момент модельного часу; у якості передумови до здійснення переходу виступає або специфікація *Init* початкового стану $s_0 \in S$, або специфікація попереднього переходу.

Ланцюжки переходів (11), що представляють собою просування пакету даних мережею від $h_0 \in H$ до $h_1 \in H$, мають вигляд кон'юнкції:

$$Trace_j \equiv Action_0 \wedge Action_1 \wedge ... \wedge Action_{m+n-1}. \tag{12}$$

З позиції спрощення, вираз $Trace_j$ можна розглядати у якості представлення сценарію функціонування системи згідно j-ї віртуальної топології.

У випадку єдиного сценарію, результуюча темпоральна формула φ (7), істинність якої має бути встановлена $\forall s \in S$ (3), матиме наступний вигляд:

$$Fi \equiv Init \wedge G(Trace_j), \tag{13}$$

де G – темпоральний оператор «Globally».

Для випадку декількох альтернативних сценаріїв маємо наступний вираз:

$$Fi \equiv Init \wedge G(Trace_0 \vee Trace_1 \vee ... \vee Trace_j \vee ... \vee Trace_{q-1}), \tag{14}$$

де q – кількість альтернативних сценаріїв функціонування системи.

Гнучкість запропонованого підходу полягає в оперуванні концептами (11) і (12) при створенні формальної моделі.

Результати проведених експериментальних досліджень показали, що перевірка віртуальної топології згідно запропонованого підходу супроводжується прийнятними часовими витратами (близько 1,3 с – при виконанні BFS-обходу), а одержувані при цьому формальні моделі є компактними та структурованими, що відповідає специфіці SDN-мереж.

Висновки. Таким чином, в роботі запропоновано підхід до перевірки віртуальної топології програмно-конфігурованої мережі, що ґрунтується на використанні темпоральної логіки TLA, відповідного формалізму TLA+ та методу автоматизованої перевірки на моделі TLC. Було одержано наступні результати:

1. Показано, що формалізм TLA+ може слугувати прийнятним засобом створення формальних моделей систем на основі програмно-конфігурованих мереж, з метою здійснення автоматизованої перевірки узгодженості взаємодії компонентів таких систем.
2. Показано, що аналітичне представлення формальної специфікації віртуальної топології на основі структури Кріпке можна використовувати у якості придатного засобу аналізу системи переходів в основі формальної специфікації віртуальної топології, що ґрунтується на залученні концепції «дії» темпоральної логіки TLA.
3. Показано, що, з позиції автоматизації, використання реалізації методу TLC, що базується на обході простору станів системи переходів методом обходу в ширину (BFS), є прийнятним кроком з позиції автоматизації.

Подальша робота напрямлена на автоматизацію процедури синтезу формальної моделі системи.

Подяки. Роботу виконано у межах проекту Erasmus+ Internet of Things: Emerging Curriculum for Industry and Human Applications ALIOT Project (reference number 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP).

Список використаних джерел

1. Atzori L., Iera A., Morabito G. The Internet of Things: A survey. *Computer Networks*. 2010. Vol. 54, No. 15. P. 2787–2805.
2. Chin W. H., Fan Z., Haines R. Emerging technologies and research challenges for 5G wireless networks. *IEEE Wireless Communications*. 2014. Vol. 21, No. 2. P. 106–112.
3. Scott-Hayward S., O'Callaghan G., Sezer S. SDN Security: A Survey. *SDN for Future Networks and Services (SDN4FNS)*: proc. 2013 IEEE conference (Trento, Italy, 11–13 Nov. 2013). 2013. P. 1–7.

4. Oliveira R. L. S., Schweitzer C. M., Shinoda A. A., Prete L. R. Using Mininet for Emulation and Prototyping Software-Defined Networks. *Communications and Computing (COLCOM)*: proc. 2014 IEEE Colombian Conference (Bogota, Colombia, 4-6 June 2014). 2014. P. 1–6.
5. Lamport L. Specifying Systems: The TLA+ language and tools for hardware and software engineers. Boston : Addison-Wesley, 2002.
6. Newcombe C., Rath T., Zhang F., Munteanu B., Brooker M., Deardeuff M. How Amazon web services uses formal methods. *Communications of the ACM*. 2015. Vol. 58, No. 4, P. 66–73.
7. Shkarupylo V. V., Tomicic I., Kasian K. M. The investigation of TLC model checker properties. *Journal of Information and Organizational Sciences*. 2016. Vol. 40, No. 1. P. 145–152.
8. Clarke E. M., Grumberg O., Peled D.A. Model checking. Cambridge, Massachusetts : MIT Press, 2001.

References

1. Atzori, L., Iera, A., Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54 (15), 2787–2805. doi: 10.1016/j.comnet.2010.05.010
2. Chin, W. H., Fan, Z., Haines, R. (2014). Emerging technologies and research challenges for 5G wireless networks. IEEE Wireless Communications, 21 (2), 106–112. doi: 10.1109/MWC.2014.6812298
3. Scott-Hayward, S., O'Callaghan, G., Sezer, S. (2013). SDN Security: A Survey. Proceedings of 2013 IEEE SDN for Future Networks and Services (SDN4FNS). Trento (Italy), 1–7. doi: 10.1109/SDN4FNS.2013.6702553
4. Oliveira, R. L. S., Schweitzer, C. M., Shinoda, A. A., Prete, L. R. (2014). Using Mininet for Emulation and Prototyping Software-Defined Networks. Proceedings of 2014 IEEE Colombian Conference on Communications and Computing (COLCOM). Bogota (Colombia), 1–6. doi: 10.1109/ColComCon.2014.6860404
5. Lamport, L. (2002). Specifying Systems: The TLA+ language and tools for hardware and software engineers. Boston: Addison-Wesley.
6. Newcombe, C., Rath, T., Zhang, F., Munteanu, B., Brooker, M., Deardeuff, M. (2015). How Amazon web services uses formal methods. Communications of the ACM, 58 (4), 66–73. doi: 10.1145/2699417
7. Shkarupylo, V. V., Tomicic, I., Kasian, K. M. (2016). The investigation of TLC model checker properties. Journal of Information and Organizational Sciences, 40 (1), 145–152. doi: 10.31341/jios.40.1.7
8. Clarke, E. M., Grumberg, O., Peled, D. A. (2001). Model checking. Cambridge, Massachusetts: MIT Press.

Шкарупило Вадим Викторович,
к.т.н., доцент, доцент кафедры компьютерных систем и сетей, Национальный университет биоресурсов и природопользования Украины
shkarupylo.vadym@gmail.com
Кудерметов Равиль Камилович,
к.т.н., доцент, заведующий кафедрой компьютерных систем и сетей, Запорожский национальный технический университет Украины
kudermetov@gmail.com
Тименко Артур Валентинович,
ассистент кафедры компьютерных систем и сетей, Запорожский национальный технический университет Украины
timenko.artur@gmail.com
Польская Ольга Владимировна,
старший преподаватель кафедры компьютерных систем и сетей, Запорожский национальный технический университет Украины
ol.polsk@gmail.com

ПОДХОД К ПРОВЕРКЕ ВИРТУАЛЬНОЙ ТОПОЛОГИИ ПРОГРАММНО-КОНФИГУРИРУЕМОЙ СЕТИ

В настоящее время концепция программно-конфигурируемых сетей широко рассматривается в качестве одной из основополагающих технологий, способствующих дальнейшему развитию Интернета – в виде Интернета вещей. При этом, основываясь на специфике названных сетей, оперируют понятием виртуальной топологии сети. Это дает возможность гибким и оперативным образом реагировать на изменение требований к распределенным программным системам, созданным на основе сети. При этом возникает проблема проверки согласованности взаимодействия компонентов системы. С этой целью в статье предложен подход к проверке виртуальной топологии программно-конфигурируемой сети. Предложенный подход основывается на использовании темпоральной логики действий (Temporal Logic of Actions, TLA). В основе предложенного подхода состоит использование концепции «действия» (action). Для представления специфики предложенного подхода использована структура Крипке. Отличительной особенностью предложенного подхода является гибкость реконфигурирования спецификации (формальной модели) системы с виртуальной топологией. Проверку топологии предложено выполнять путем автоматизированной формальной верификации методом проверки на модели. С целью способствования автоматизации, использована реализация метода проверки на модели TLA Checker (TLC), которая основывается на обходе пространства состояний системы переходов в основе формальной модели методом обхода в ширину (Breadth-first Search, BFS).

***Ключевые слова**: ad-hoc, Quality of Services, верификация, виртуальная топология, Интернет вещей, проверка на модели, программно-конфигурируемая сеть, спецификация.*

Vadym Shkarupylo,
Candidate of Technical Sciences, Docent, Associate Professor of the Department of Computer Systems and Networks, National University of Life and Environmental Sciences of Ukraine
shkarupylo.vadym@gmail.com
Ravil Kudermetov,
Candidate of Technical Sciences, Docent, Head of the *Department of Computer Systems and Networks, Zaporizhzhia National Technical University of Ukraine*
kudermetov@gmail.com
Artur Timenko,
Assistant of the Department of Computer Systems and Networks, Zaporizhzhia National Technical University of Ukraine
timenko.artur@gmail.com
Olga Polska,
Senior Lecturer of the Department of Computer Systems and Networks, Zaporizhzhia National Technical University of Ukraine
ol.polsk@gmail.com

APPROACH TO VIRTUAL TOPOLOGY OF SOFTWARE-DEFINED NETWORK CHECKING

At present, the concept of Software-defined Networks (SDN) is broadly considered to be among the defining technologies prompting further development of the Internet – as the Internet of Things (SDN). At the same time, taking into consideration the peculiarities of named networks, the concept of virtual topology is utilized. This allows react on the changes of the requirements to distributed software systems, built upon the network, in a flexible and agile manner. The problem of checking the consistency of resulting software solution arises here. With this statement in mind, the approach to checking the consistency of SDN-network virtual topology has been proposed. The proposed approach is grounded on the Temporal Logic of Actions (TLA) and corresponding TLA+ formalism usage. The approach is built over the concept of "action" – specification of transition between the states of transition system, defined by the TLA+ specification. To analytically represent the peculiarities of the proposed approach, the Kripke structure is used. The distinctive feature of the proposed approach is the flexibility – the formal model (specification) of the system can be operatively reconfigured with respect to the change of virtual topology of the network. To check the specification in an

automated manner, the TLC (TLA Checker) implementation of model checking method is utilized. The implementation is based on traversing the states of transition system by way of Breadth-first Search (BFS).

***Keywords**: ad-hoc, Quality of Services, verification, virtual topology, Internet of Things, Model Checking, Software-defined Network, specification.*