

Problem komunikacji federatów w symulacji opartej o HLA

Krzysztof PANUFNIK*

1. Wprowadzenie

Standard HLA (ang. *High Level Architecture*) powstał z myślą o łączeniu ze sobą różnorodnych symulatorów [1], zwanych w standardzie federatami, bez względu na technologię, w której zostały wykonane. Dzięki połączeniu federatów za pomocą HLA możliwe jest tworzenie rozbudowanych symulacji (zwanych federacjami), w których symulatory współdzielą informacje o obiektach symulacyjnych. HLA jest standardem, który realizowany jest za pomocą RTI (ang. *Run Time Infrastructure*).

Wraz ze wzrostem liczby federatów podłączonych do HLA oraz symulowanych obiektów wzrasta problem dystrybucji informacji o zmianach w tych obiektach do federatów. HLA definiuje dwie usługi, których można użyć w zależności od skali symulacji. Mają one za zadanie ograniczenie liczby wysyłanych zmian.

Dużej liczbie zmian obiektów towarzyszy problem sposobu wykorzystania zasobów sieciowych do dystrybucji zmian. Implementacje HLA do dystrybucji danych w sieci lokalnej wykorzystują grupy multicast protokołu IP. Liczba takich grup może okazać się niewystarczająca w przypadku dużych eksperymentów symulacyjnych. Potrzebny jest wtedy algorytm podziału federatów na grupy, który zapewni wydajną komunikację.

2. Usługi filtrowania danych HLA

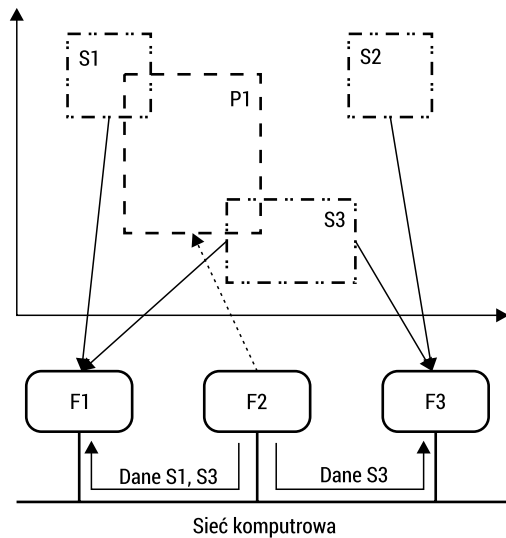
HLA przewiduje istnienie dwóch usług, których celem jest ograniczenie liczby przesyłanych informacji. Są to Zarządzanie Deklaracjami (ang. *Declaration Management*) oraz Zarządzanie Dystrybucją Danych (ang. *Data Distribution Management*) [2]. Obie te usługi działają, wykorzystując wzorzec publikuj–subskrybuj. Federaty mogą zadeklarować, jakie obiekty zamierzają zmieniać lub o zmianach w jakich obiektach chcą być informowane.

* Wojskowa Akademia Techniczna

Usługa DM pozwala na definiowanie zainteresowania poprzez wskazanie typu obiektu, np. pojazdu, czołgu czy motocykla. W połączeniu z możliwościami modelowania obiektów symulacyjnych (wprowadzania relacji „jest” między obiektami) możliwe jest zadeklarowanie zainteresowania jedną grupą obiektów, co przełoży się na zainteresowanie grupami obiektów, które są związane relacją „jest” ze wskazaną grupą.

Usługa DDM daje dużo większe możliwości wyrażania zainteresowań [3]. Jest to możliwe dzięki wprowadzeniu filtrowania na poziomie atrybutów obiektów. Podczas projektowania modelu obiektów symulacyjnych definiowane są atrybuty, po których może odbywać się filtrowanie, oraz wartości, jakie atrybuty te mogą przyjmować. Atrybut taki nazywany jest wymiarem, a zbiór jego wartości tłumaczony jest na dodatnie wartości liczbowe. Po zdefiniowaniu wszystkich atrybutów filtrujących możliwe jest stworzenie wielowymiarowej przestrzeni (o liczbie wymiarów równej liczbie atrybutów filtrujących). W tak zdefiniowanej przestrzeni wprowadza się następujące pojęcia:

- zakres – przedział będący podzbiorem wartości wymiaru,
- obszar – wielowymiarowa bryła powstała przez zdefiniowanie niepustych zakresów dla każdego z wymiarów.



RYS. 1. Działanie usługi DDM

FIG. 1. DDM service

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Działanie usługi DDM przebiega w czterech fazach:

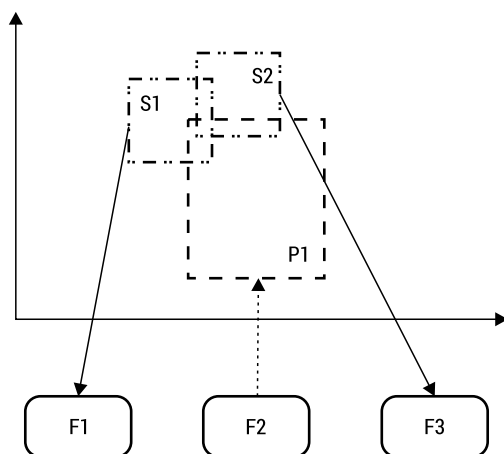
- deklarowanie – federaci deklarują zainteresowanie danymi poprzez wskazanie jednego lub więcej obszarów wraz ze wskazaniem, jakiego typu jest do obszar (związany z subskrypcją lub publikacją);

- wyszukiwanie przecięć – poszukiwanie części wspólnych między obszarami publikującymi a obszarami subskrybującymi,
- połączenie federatów – wskazanie kanałów komunikacji, którymi federaty mają się posługiwać w celu wymiany danych związanych z poszczególnymi obszarami danych;
- wymiana danych – przesłanie danych, zgodne z wcześniej ustalonymi kanałami komunikacji.

Na rysunku 1 przedstawiono przykład działania usługi DDM dla federacji, w skład której wchodzi trzy federaty. Zdefiniowano dwa atrybuty filtrujące, co spowodowało stworzenie dwuwymiarowej przestrzeni. W przestrzeni tej zostały zadeklarowane następujące regiony:

- subskrypcji (S1, S2 oraz S3) – federaty F1 i F3 są zainteresowane zmianami obiektów znajdujących się odpowiednio w obszarach S1 i S3 dla federata F1 oraz S2 i S3 dla federata F3;
- publikacji (P1) – federat F2 zadeklarował zmienianie obiektów znajdujących się w tym obszarze.

Region P1 przecina się z regionami S1 i S3. W rezultacie federat F2 będzie wysyłał dane o zmianach w regionie P1 (a więc i części regionów S1 i S3) do federatów F1 i F3. W ten sposób federaty F1 i F3 otrzymają wszystkie zmiany, którymi były zainteresowane. Federat F3 nie otrzyma informacji o zmianach w obszarze S2, gdyż nie ma federata, który by zmieniał dany obszar.



RYS. 2. Przecięcie obszarów zainteresowań

FIG. 2. Regions intersection

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Należy wspomnieć o pewnych ograniczeniach, które zostały poczynione przy tworzeniu wymagań dla usługi DDM. Przyjęto, że zakresy są przedziałem dopuszczalnych wartości wymiaru. To w połączeniu ze sposobem definiowania obszarów, tj. przez podanie zakresów, które je definiują, skutkuje tym, że obszary mogą przyjmować jedynie kształt hiperprostokątów. Często zdarza się, że przecięcia obszarów nie są hiperprostokątami. Taką sytuację przedstawia rysunek 2.

Można wyróżnić istnienie trzech przecięć, których zmiany powinny być rozsyłane do federatów:

$$I1 = S1 \cap P1 \setminus S2$$

$$I2 = S2 \cap P1 \setminus S1$$

$$I3 = S1 \cap S2 \cap P1$$

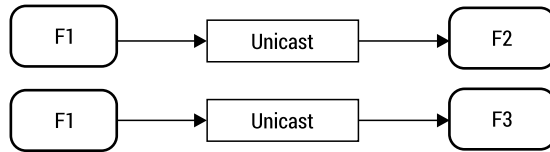
Do federata F1 powinny zostać wysłane zmiany z przecięć I1 oraz I3, a do federata F2 – I2 oraz I3. Ze względu na ograniczenia przyjęte w standardzie HLA tak się nie stanie, a oba federaty otrzymają informacje o wszystkich zmianach w obszarze P1, nawet tych, którymi nie byli zainteresowani. Ograniczenie to zostało najprawdopodobniej wprowadzone, aby ułatwić poszukiwanie przecięć między obszarami.

Powyższy problem można częściowo rozwiązać poprzez zdefiniowanie bardzo wielu małych regionów publikacji i subskrypcji. Miałyby to jednak negatywny wpływ na proces odnajdywania przecięć między obszarami. W dalszej części rozważać się będzie obszary i przecięcia o dowolnych kształtach.

Deklaracje obszarów mogą zmieniać się w czasie. W przypadku zmiany wymagane jest ponowne ustalenie kanałów do komunikacji federatów.

3. Kanały komunikacji w sieci komputerowej

Najczęstszą metodą poprawy wydajności jest wykorzystanie sieciowych grup multicast. Jest to jedna z trzech metod komunikacji w sieci komputerowej. Pozostałe dwie to unicast, czyli połączenie „jeden do jeden”, oraz broadcast, czyli połączenie „wszyscy do wszystkich”. Połączenia typu multicast pozwalają na wysyłanie danych tylko do komputerów będących członkami grupy multicast. W efekcie komputer wysyłający dane może wysłać je jednokrotnie. Alternatywą dla tego rozwiązania byłoby wysyłanie do każdego z odbiorców danych za pomocą połączenia unicast, co skutkowałoby wielokrotnym przesyłaniem tych samych informacji. Dzięki połączeniom multicast możliwe jest więc realizowanie komunikacji typu „jeden do wielu” oraz „wiele do wielu”. Na rysunkach 3, 4 oraz 5 przedstawiono możliwe sposoby wykorzystania kanałów multicast i unicast do komunikacji.

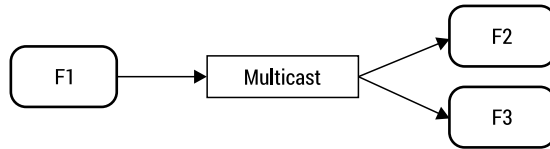


RYS. 3. Komunikacja z wykorzystaniem kanałów unicast

FIG. 3. Unicast communication

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

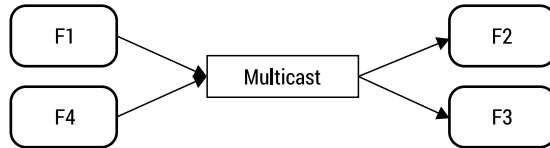


RYS. 4. Komunikacja typu „jeden do wielu” z wykorzystaniem kanału multicast

FIG. 4. One to many multicast communication

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.



RYS. 5. Komunikacja typu „wiele do wielu” z wykorzystaniem kanału multicast

FIG. 5. Many to many multicast communication

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Choć komunikacja za pomocą kanałów multicast wydaje się odpowiadać na potrzeby dystrybucji danych w symulatorach opartych o HLA, nie jest ona pozbawiona wad. Główną jest ograniczona liczba grup multicast, które można utworzyć. Choć istnieją urządzenia, które pozwalają na zdefiniowanie 1024-2000 grup multicast [4][5], nierzadko spotyka się wartości 256 lub mniejsze [6, 7]. Wartości te mogą być niewystarczające dla złożonych symulacji o dużej liczbie obszarów.

Tworzenie nowych grup multicast wiąże się ze zbudowaniem drzewa routingu między nadawcą a odbiorcami. W zależności od wybranego algorytmu komunikacji multicast koszt stworzenia drzewa, jak i jego wpływ na ogólną wydajność sieci, może być różny [8]. Zdefiniowanie dużej liczby grup multicast stanowi więc obciążenie dla sieci, nawet gdy grupa nie jest wykorzystywana. Czas potrzebny na dołączenie

lub opuszczenie grupy multicast również może być znaczący (od kilkuset milisekund do kilku sekund [9]). W związku z tym wykorzystanie grup multicast sprawdza się najlepiej w przypadku symulacji, w której nie występuje wiele zmian obszarów.

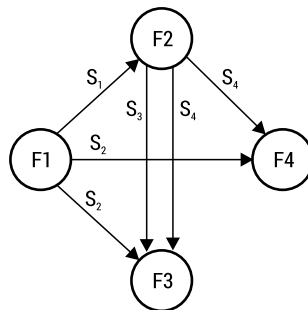
4. Dotychczasowe rozwiązania

Dotychczasowe metody rozwiązania problemu podziału federatów na grupy multicast można podzielić na bazujące na:

- regionach – sprawdzane są wszystkie przecięcia regionów publikujących ze wszystkimi regionami subskrybującymi, a grupy multicast są przypisywane w przypadku wystąpienia przecięcia;
- siatce – przestrzeń dzieli się dodatkowo za pomocą siatki o stałym lub zmiennym rozmiarze komórki; następnie danej komórce przypisuje się grupę multicast, w zależności od przyjętej metody zawsze lub tylko wtedy, gdy w danej komórce występują przecinające się regiony.

Do rozwiązań opartych o regiony należą algorytmy LOC (ang. *Largest Outgoing Connection*), IRLOC (ang. *Input-Restricted LOC*) [10] oraz ACPS (ang. *Adaptive Communication Protocol Selection*).

Dla każdego regionu publikującego, dla którego zachodzi przecięcie z regionem subskrybującym, definiuje się strumień zmian. Federaty subskrybujące, których regiony przecięły się z regionem publikującym, są odbiorcami strumienia zmian. Zależności między federatami definiowane przez strumienie można przedstawić za pomocą grafu strumieni zmian (rys. 6).



RYS. 6. Graf strumieni zmian
FIG. 6. A change stream graph

ŹRÓDŁO: opracowanie własne.
SOURCE: own elaboration.

Ze strumieniem zmian skojarzone są więc: federat, waga strumienia będąca sumą rozmiarów wszystkich zmian wysłanych przez federata z danego obszaru oraz zbiór federatów, do których należy wysłać zmiany. Zbiór strumieni zmian dla grafu z rysunku 6 to:

$$S = \left\{ \langle f_1, w_1, \{f_2\} \rangle, \langle f_1, w_2, \{f_3, f_4\} \rangle, \langle f_2, w_3, \{f_3\} \rangle, \langle f_2, w_4, \{f_3, f_4\} \rangle \right\}$$

Mając dany zbiór strumieni zmian S , algorytmy dzielą strumień zmian między m grup multicastowych tak, aby czas przesłania wszystkich zmian był jak najmniejszy, nie większy niż t_{\max} . Mając następujące dane:

- zbiór federatów:

$$F = \{f_1, f_2, \dots, f_n\}, f_i \in N$$

- macierz przepustowości łącza między federatami:

$$C = [c_{i,j}] c_{i,j} \in N, i, j \in F$$

- funkcję opisującą liczbę zmian wysłaną przez federata:

$$gs(x) \in N, x \in F$$

- czas wysyłki pojedynczej zmiany przez federata:

$$ts_i \in N, i \in F$$

- funkcję opisującą liczbę zmian odebraną przez federata:

$$gr(x) \in N, x \in F$$

- czas odbioru pojedynczej zmiany przez federata:

$$tr_i \in N, i \in F$$

- funkcję opisującą liczbę bajtów przesłanych od federata x_1 do federata x_2 :

$$gt(x_1, x_2) \in N, x_1, x_2 \in F$$

- czas przesłania wszystkich wiadomości daje się oszacować jako:

$$\max \left\{ gs(i) \cdot ts_i + \frac{gt(i,j)}{c_{i,j}} + gr(j) \cdot tr_j : i, j \in F \right\}$$

Algorytm LOC sortuje listę strumieni na podstawie ich wag. Poczynając od strumienia o największej wadze algorytm sprawdza, czy możliwe jest dodanie nadawcy i odbiorców strumienia do grupy multicast. Jeżeli dodanie strumienia do grupy spowodowałoby przekroczenie t_{\max} , sprawdzana jest możliwość dodania strumienia do kolejnej grupy. W przypadku, gdy nie ma więcej grup, strumień jest rozsyłany do wszystkich odbiorców za pomocą połączeń unicast. Wadą algorytmu jest to, że skupia się on wyłącznie na minimalizacji obciążenia federatów wysyłających.

Powyższy problem rozwiązuje algorytm IRLOC. Wprowadza on dodatkową miarę strumienia, jaką jest zysk z użycia połączenia multicast:

$$(k_i - 1) \cdot w_i$$

gdzie k_i jest liczbą federatów odbierających i -tego strumienia. Miara opisuje, ile bajtów mniej zostanie wysłanych w przypadku użycia połączenia multicast zamiast rozsyłania wiadomości wielokrotnie połączeniem unicast. Dołączenie strumienia do grupy multicast może przynieść negatywny skutek w postaci zwiększenia się liczby nadmiarowych wiadomości u federatów:

- będących już członkami grupy multicast, ze względu na to, że nie są odbiorcami dodawanego strumienia;
- będących odbiorcami dodawanego strumienia, ale nie będących odbiorcami jednego bądź większej liczby strumieni już dodanych do grupy.

Algorytm IRLOC wprowadza dwie zmiany w stosunku do algorytmu LOC:

- strumienie są sortowane nie ze względu na wagę, ale ze względu na zysk z użycia połączenia multicast;
- strumień jest dodawany do grupy tylko wtedy, gdy jego zysk z użycia połączenia multicast przewyższa negatywny skutek.

Wadą algorytmu IRLOC jest to, że nie uwzględnia on faktu, że strumienie mogą mieć podobną grupę federatów odbierających.

Problem ten rozwiązuje algorytm ACPS. Wprowadza on miarę odległości między strumieniem a grupą multicast. Miarą tą jest negatywny skutek dodania strumienia do grupy. W stosunku do algorytmu IRLOC zmienia się wybór grupy, do której należy dodać strumień. W algorytmie IRLOC grupy były sprawdzane po kolei, natomiast w algorytmie ACPS wybierana jest ta grupa, dla której odległość strumienia jest najmniejsza.

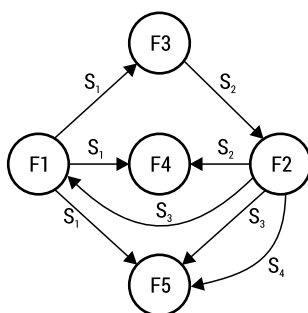
5. Koncepcja rozwiązania problemu

Opisane algorytmy podejmowały decyzję o dodaniu strumienia do grupy multicast na podstawie jednego bądź obu kryteriów minimalizacji obciążenia federatów wysyłających i odbierających. Oba te kryteria wpływają na czas przesłania wszystkich wiadomości. Omówione algorytmy nie biorą jednak pod uwagę, że federaty nie są sobie równe. Dzieje się tak dlatego, że podstawą podejmowania decyzji są strumienie zmian. Czas transmisji jest ograniczony od góry przez parę federatów najbardziej obciążonych. Traktowanie podziału z punktu widzenia strumieni zmian nie daje możliwości odciążenia federatów z par najbardziej obciążonych. W celu rozwiązania tego problemu proponuje się wprowadzenie następujących zmian:

- podział strumieni na mniejsze poprzez rozdzielenie federatów odbierających;

- inny sposób sortowania strumieni, który zapewni najbardziej obciążonym federatom pierwszeństwo w przydziale do grup multicast;
- wybór grupy, do której należy dodać strumień powinien być motywowany minimalizacją obciążenia najbardziej obciążonych par federatów, co doprowadziłoby do zmniejszenia całkowitego czasu przesłania.

Przykład eksperymentu symulacyjnego, dla którego zaproponowane zmiany mogą poprawić wydajność komunikacji, przedstawiono na rysunku 7. Parametry strumieni zostały natomiast zaprezentowane w tabeli 1.



RYS. 7. Graf strumieni zmian
FIG. 7. A change stream graph

ŹRÓDŁO: opracowanie własne.
SOURCE: own elaboration.

Zakłada się następujące parametry:

- czas wysłania i odebrania pojedynczej zmiany przez federata jest taki sam dla wszystkich federatów i wynosi 1ms,
- czas przesłania zmiany między dowolną parą federatów wynosi 2ms,
- dostępna liczba grup multicast równa jest 2.

TAB. 1. Parametry strumieni zmian
TAB. 1. Change streams parameters

Strumień	Nadawca	Odbiorcy	w_i	$(k_i - 1) \cdot w_i$
S_1	1	3,4,5	5	10
S_2	2	3,4	5	5
S_3	2	1,5	4	4
S_4	2	5	3	0

ŹRÓDŁO: opracowanie własne.
SOURCE: own elaboration.

TAB. 2. Przepisanie strumieni zmian do kanałów komunikacji
 TAB. 2. Assignment of change streams to communication channels

Algorytm	Grupa 1	Grupa 2	Połączenia bezpośrednie
IRLOC	$\langle 1, S_1, \{3,4,5\} \rangle$ $\langle 2, S_2, \{3,4\} \rangle$	$\langle 2, S_3, \{1,5\} \rangle$	$\langle 2, S_4, \{5\} \rangle$
ACPS	$\langle 1, S_1, \{3,4,5\} \rangle$	$\langle 2, S_2, \{3,4\} \rangle$	$\langle 2, S_3, \{1\} \rangle$ $\langle 2, S_3, \{5\} \rangle$ $\langle 2, S_4, \{5\} \rangle$
Proponowany algorytm	$\langle 1, S_1, \{3,4\} \rangle$ $\langle 2, S_2, \{3,4\} \rangle$	$\langle 2, S_3, \{1,5\} \rangle$	$\{1, S_1, \{5\} \rangle$ $\langle 2, S_4, \{5\} \rangle$

ŹRÓDŁO: opracowanie własne.
 SOURCE: own elaboration.

TAB. 3. Liczba zmian wysyłanych i odbieranych przez federaty oraz czas przesłania
 TAB. 3. Number of changes send and received by federates and total transmission time

Federaty	Wysyłające		Odbierające				Czas
	1	2	1	3	4	5	
Minimum	5	12	4	10	10	12	48
IRLOC	5	12	4	10	10	17	53
ACPS	5	16	4	10	10	12	60
Proponowany algorytm	10	12	4	10	10	12	48

ŹRÓDŁO: opracowanie własne.
 SOURCE: own elaboration.

Przypisanie strumieni zmian do kanałów komunikacyjnych, będące wynikiem działania poszczególnych algorytmów, przedstawiono w tabeli 2. Przypisanie ma postać trójki:

(nadawca, strumień, zbiór odbiorców)

Z analizy grafu strumieni zmian wynika, że federat 2 jest tym, który musi rozesłać największą liczbę zmian. Z federatów odbierających federat 5 jest natomiast tym, który musi odebrać największą liczbę zmian. W tabeli 3 przedstawiono wyniki dla poszczególnych przypisań. Algorytm IRLOC spowodował dodatkowe obciążenie federata 5. Algorytm ACPS minimalizuje nadmiary po stronie federatów odbierających, czego efektem było dodatkowe obciążenie dla federata 2. Zaproponowany algorytm nie powoduje dodatkowego obciążenia dla najwolniejszej pary federatów 2 i 5. W efekcie

działania algorytmu dodatkowo obciążony zostaje federat 1. Liczba zmian, które federat miał rozesłać nie jest duża, więc dodatkowe obciążenie nie powoduje pogorszenia całkowitego czasu przesłania zmian.

6. Podsumowanie

W opracowaniu przedstawiono podstawy problemu komunikacji federatów w środowisku HLA oraz główne ograniczenia sieciowe, które wpływają na dobór sposobu komunikacji. Jednocześnie zwrócono uwagę na istotność doboru strategii komunikacji na wydajność eksperymentu symulacyjnego. Zauważono ograniczenia przedstawionych algorytmów i przedstawiono koncepcje zmiany algorytmów celem odciążenia najbardziej obciążonych federatów. Pokazano istnienie eksperymentów symulacyjnych, potwierdzających, że zmiana ta powinna pozytywnie wpłynąć na czas trwania eksperymentu.

Literatura

1. "Introduction to IEEE Std 1516-2010, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules", IEEE 2010.
2. "IEEE Std 1516-2010, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules", IEEE 2010.
3. "IEEE Std 1516-2010, IEEE Standard for Modeling and Simulation (M&S) High Level architecture (HLA) Federate interface specification", IEEE2010.
4. https://extremenetworks-ua.com/assets/files/SummitX650/DSSumX650_1442.pdf.
5. https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/multicast-limiting-group-joins-igmp.html.
6. <https://www.manualslib.com/manual/232384/3com-5500g-Ei.html?page=247>.
7. https://gtacknowledge.extremenetworks.com/articles/Q_A/What-is-the-maximum-number-of-IPv4-Multicast-groups-supported-on-Extreme-devices.
8. Billhartz T, Cain J, Farrey-Goudreau B et al. Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols, *IEEE Journal on Selected Areas in Communications*. 1997; 15(3): 304-315.
9. <https://www.networkworld.com/article/2241579/multicast-group-capacity-extreme-comes-out-on-top.html>.
10. Morse K, Zyda M. Multicast Grouping for Data Distribution Management. *Simulation Practice and Theory*. 2002; 9.
11. Wang J, Zheng T. A hybrid multicast-unicast assignment approach for data distribution management in HLA. *Simulation Modelling Practice and Theory*. 2014; 40(0): 39-63.

Streszczenie

W pracy przedstawiono problem komunikacji między federatami symulatora opartego o standard HLA. Omówiono zagadnienia sieciowe związane z komunikacją federatów oraz główne ograniczenia aspektów komunikacji sieciowej. Przedstawiono wybrane algorytmy wyboru strategii komunikacji wraz z opisem ich wad. Zaproponowano zmiany w doczasowych algorytmach, które mogą pozytywnie wpłynąć na wydajność komunikacji federatów.

Słowa kluczowe: symulacja, HLA, Data Distribution Management, Interest Management

Summary

Federate communication problem in HLA-based simulation

The paper presents communication problem between federated in HLA-based simulation. The networking background related to federates communication with consideration of limits of such communication was presented. Selected communication strategy algorithms and their weaknesses have been described. Improvements to those algorithms, which should improve communication performance, were proposed.

Keywords: Simulation, HLA, Data Distribution Service, Interest Management