

**Michał DYK, Andrzej NAJGEBAUER,
Dariusz PIERZCHAŁA**

Wojskowa Akademia Techniczna, Wydział Cybernetyki,
ul. Kaliskiego 2, 00-908 Warszawa
E-mail: mdyk@wat.edu.pl, anajgebauer@wat.edu.pl,
dariusz.pierzchala@wat.edu.pl

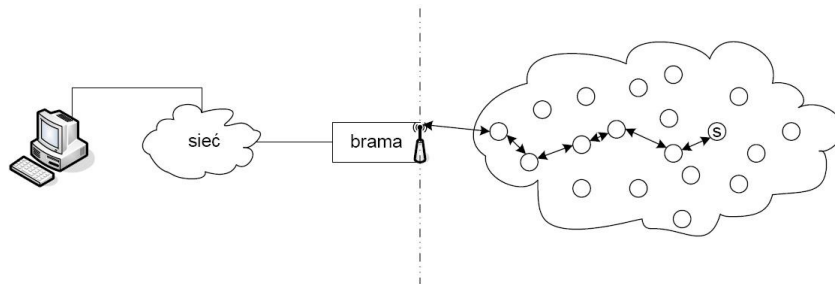
Model symulacyjny sieci sensorowej na potrzeby systemów fuzji danych

1 Wstęp

Zagadnienie fuzji danych sensorycznych jest obecnie obiektem wciąż rosnącego zainteresowania. Zgromadzenie danych z wielu źródeł i zasilenie nimi mechanizmów agregacji i wnioskowania pozwala uzyskać dużo więcej informacji, niż niosą ze sobą „surowe” odczyty z pojedynczych sensorów. Dzięki temu możliwe jest dotarcie do wiedzy, która nie zawsze jest widoczna wprost. Jednak, aby ten proces przebiegał poprawnie, konieczny jest dostęp do dużej ilości różnorodnych danych, na których system może się „uczyć” i wnioskować. Ostatnie lata przyniosły ogromny postęp w rozwoju urządzeń mobilnych, w szczególności bezprzewodowych, które stając się „wszechobecne”, mają możliwość zaspokoić to zapotrzebowanie. Rodząca się koncepcja Internetu Rzeczy (ang. *Internet of Things*) pozwala twierdzić, że w niedługim czasie dostęp do danych sensorycznych będzie powszechny. Pomimo tego, w wielu przypadkach, przygotowanie środowiska zawierającego rozlokowane rzeczywiste sensory jest bardzo kosztowne i czasochłonne. Rozwiązaniem dla potrzeb badań jest przygotowanie systemu symulacyjnego zapewniającego dane sensoryczne na potrzeby ich fuzji. Niniejszy artykuł prezentuje model matematyczny oraz architekturę oprogramowania spełniające te oczekiwania.

2 Sieci sensorowe i Internet Rzeczy

Sieć sensorową najogólniej można określić jako sieć złożoną z wielu - najczęściej więcej niż kilkunastu - urządzeń, które rozlokowane na ustalonym obszarze realizują wspólne zadanie [1]. Podstawowym elementem jest sensor (stąd nazwa sieci), nazywany także węzłem. Każdy z nich ma takie same możliwości komunikacyjne i żaden nie jest wyróżniony w momencie tworzenia sieci. Przyjmuje się, że sieć sensorowa posiada „bramę” (nazywaną też „ujściem”), czyli element, do którego przesyłane są zgromadzone dane lub z którego wysyłane są zapytania do sieci. Ogólny schemat bezprzewodowej sieci sensorowej przedstawia rysunek 1. Przesyłanie danych pomiędzy bramą a sensorami (lub w razie potrzeby tylko pomiędzy węzłami) odbywa się najczęściej metodą „skoków”. Jeśli węzeł będący źródłem komunikatu nie ma bezpośredniego połączenia z bramą, przesyła dane do węzła sąsiedniego, który z kolei przekazuje komunikat następnym sensorom. Sieci sensorowe mają wiele zastosowań - wykorzystywane są między innymi do monitorowania zjawisk przyrodniczych, wczesnego ostrzegania przed lawinami lub pożarami (np. lasów). Oczywiście, mają także zastosowania militarne, np. w celu wykrywania lub śledzenia przeciwnika.



Rys. 1. Ogólny schemat sieci sensorowej. Źródło: Wikipedia

Fig. 1. The general scheme of sensor network. Source: Wikipedia

W ciągu ostatnich kilku lat coraz prężniej rozwija się koncepcja Internetu Rzeczy (ang. *Internet of Things*, w skrócie IoT), która powstała w badaniach i projektach związanych z sieciami sensorowymi. Głównym postulatem jest podłączenie jak największej liczby urządzeń elektronicznych do sieci Internet. W efekcie każde z nich otrzyma własny adres internetowy, za pośrednictwem którego możliwe będzie pozyskiwanie danych lub sterowanie nim. Daje to ogromne możliwości - począwszy od regulowania ogrzewania we własnym domu, po zaawansowane monitorowanie i sterowanie procesem produkcyjnym za pośrednictwem jednego komputera, laptopa czy telefonu. Co równie istotne, dostęp do różnego rodzaju danych sensorycznych staje się łatwy i powszechny.

3 Fuzja danych

Proces gromadzenia informacji z wielu źródeł i ich łączenia/analizy w celu uzyskania wniosków niemożliwych do otrzymania z pojedynczych obserwacji, rozumiany pod pojęciem fuzji danych, nie jest nowym zagadnieniem [2]. Jako ludzie w naturalny sposób dokonujemy fuzji danych z naszych zmysłów (np. określenie czy owoc jest świeży i nadaje się do spożycia, wymaga zaangażowania większości z nich). W obszarze zagadnień technologicznych bardzo istotny wpływ na rozwój fuzji danych miały (i nadal mają) rozwiązania militarne, w szczególności zadania śledzenia celów (ang. *target tracking*). Wykorzystywano do tego najczęściej dane z wielu stacji radarowych, co powodowało konieczność określenia, czy dotyczą one jednego czy wielu różnych obiektów.

Pojawienie się i upowszechnienie sieci sensorowych spowodowało wydzielenie oddzielnej gałęzi fuzji danych, nazywanej multisensorową fuzją danych (ang. *multisensor data fusion*). Jej istotne zalety to [2]:

- wykorzystanie wielu sensorów tego samego typu powoduje większą dokładność prowadzonych obserwacji – szacuje się, że zwiększenie liczby sensorów o N powoduje zwiększenie dokładności estymacji położenia celu o współczynnik proporcjonalny do $N^{1/2}$;
- środowisko wielosensorowe wykorzystywane jest dla nawigacji typu VOR; w tym celu wyznaczane są położenia obiektu względem sensorów – wyklucza to konieczność stosowania na przykład systemów GPS;

- wykorzystanie danych z heterogenicznych sensorów powoduje poprawienie zdolności obserwacyjnych – uzupełniające się dane z różnych rodzajów źródeł mogą umożliwić zdobycie bardziej dokładnej lub zupełnie nowej informacji.

W zależności od potrzeb oraz zastosowanych sensorów wyróżnić można trzy zasadnicze sposoby dokonywania fuzji danych pochodzących ze środowiska multisensorowego:

- 1) bezpośrednia fuzja danych sensorycznych,
- 2) reprezentacja danych sensorycznych poprzez wektor cech (ang. *feature vector*), które następnie poddawane są fuzji,
- 3) sensory dokonują przetwarzania do wysokiego poziomu, czyli wniosków lub wręcz decyzji – to one są przedmiotem dalszej fuzji.

Pierwszy z przedstawionych sposobów jest najczęściej stosowany w przypadkach, kiedy sensory obserwują to samo zjawisko, a zbierane przez nie dane są możliwe do porównania (np. pomiar temperatury w dużym budynku lub zliczanie wolnych miejsc w parkingu podziemnym). Pozostałe sposoby stosowane są w przypadkach, kiedy dane odczytywane z sensorów nie są wprost porównywalne. Fuzja na poziomie wektorów cech zakłada, że sensory samodzielnie dokonują ekstrakcji reprezentatywnych cech z zebranych przez siebie danych a następnie organizują je w strukturę wektora, który przesyłany jest do serwera dokonującego fuzji. Fuzja na poziomie decyzji zakłada, że każdy z sensorów samodzielnie, na podstawie własnych obserwacji, podejmuje decyzję na przykład o położeniu obserwowanego obiektu, jego tożsamości itp.

Na potrzeby standaryzacji procesów fuzji danych, w połowie lat osiemdziesiątych ubiegłego wieku, opracowany został model referencyjny pod nazwą JDL [2]. Powstał on w wyniku pracy grupy nazywanej Joint Directors of Laboratories Data Fusion Working Group. Zakłada on, że gromadzone przez sensory dane będą przetwarzane na czterech etapach, z których każdy realizuje wnioskowanie na coraz wyższym poziomie, tj.: ekstrakcja sygnałów, identyfikacja obiektu, identyfikacja sytuacji, identyfikacja zagrożenia.

4 Model symulacyjny sieci sensorowej

W niniejszym punkcie przedstawione zostaną główne założenia modelu symulacyjnego sieci sensorowej. W oparciu o ten model zaprojektowany został symulator, który umożliwi generowanie danych na potrzeby systemów fuzji danych. Podobnie jak inne tego typu modele [4], również prezentowany w pracy oparto na teorii grafów.

W dalszych rozważaniach przyjęto następujące oznaczenia:

- $S \in Z_+^N$ - niepusty zbiór identyfikatorów sensorów (węzłów sieci),
- SN - zbiór wszystkich możliwych sieci,
- $L_s(t) \in \psi$ - lokalizacja sensora n w chwili t wyrażona jako współrzędne w kartezjańskim układzie współrzędnych,
- $d(L_i(t), L_j(t))$ - odległość euklidesowa pomiędzy lokalizacjami sensorów i oraz j w chwili t ,
- $r_i \in Z_+^N$ - zasięg radiowy węzła i ,
- $V_i \in Z_+^N$ - prędkość węzła i ,
- $\psi \in (R \times R)^N$ - zbiór wszystkich możliwych lokalizacji.

Proponowany model zapewnia możliwość symulowania heterogenicznej sieci sensorowej działającej na zadanym obszarze z możliwością przemieszczania się węzłów. Sieć modelowana jest jako graf prosty, w którym gałąź reprezentuje możliwość komunikacji pomiędzy dwoma węzłami (1):

$$SN(t) = \langle S, E(t), b \rangle, \quad (1)$$

gdzie b to funkcja zdefiniowana na krawędziach grafu, określająca przepustowość łącza wyrażaną w bitach na jednostkę czasu:

$$b: S \times S \rightarrow Z_+^N. \quad (2)$$

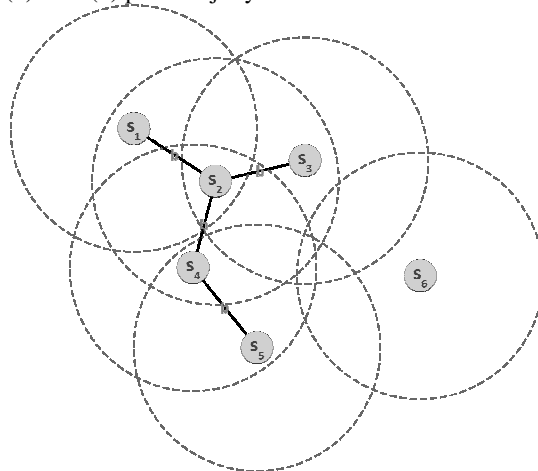
Dzięki temu dla każdej pary sensorów, które są sąsiadami, możliwe jest określenie przepustowości ich łącza, co jest kluczowe dla procesu komunikacji (opisany w dalszej części). $E(t)$ to zbiór krawędzi grafu określony jako:

$$E(t) = \{ \langle x, y \rangle : x, y \in S \wedge x \neq y \wedge x \in N_y(t) \wedge y \in N_x(t) \}. \quad (3)$$

$N_x(t)$ i $N_y(t)$ to zbiory identyfikatorów węzłów, które są sąsiadami odpowiednio sensora x i y . Sąsiedztwo definiowane jest następująco:

$$N_i(t) \in \{ x \in S : d(L_x(t), L_i(t)) \leq \min\{r_i, r_x\} \}. \quad (4)$$

Z zależności (4) wynika, że sąsiadem węzła i -tego są wszystkie te sensory, z którymi znajduje się on w zasięgu radiowym. W przypadku różnych zasięgów uwzględniany jest najmniejszy z nich. Wynika z tego, że sąsiedztwo jest równoważne możliwości dwukierunkowej komunikacji. Przykład sieci zbudowanej przy uwzględnieniu zależności (1), (2), (3) oraz (4) prezentuje rysunek 2.



Rys. 2. Sieć sensorów zgodna z modelem symulatora. Źródło: Opracowanie własne
 Fig. 2. Sensor network compatible with the model of the simulator
 Source: Own preparation

Łatwo zauważyć, że w przypadku kiedy zasięg radiowy wszystkich węzłów jest identyczny, graf sprowadza się do grafu typu Unit Disc Graph [6].

Sieć w proponowanym modelu nie jest stacjonarna w czasie i jej struktura może się zmieniać. Wpływ na nią mogą mieć: ruch węzłów lub zdarzenia symulacyjne, których wynikiem może być na przykład uszkodzenie lub zniszczenie sensorów.

Sensory, z których składa się sieć, definiowane są jako zbiór ośmiu cech zgodnie z zależnością (5):

$$s(t) = \langle L(t); V; r; A_s; O(t); P; R(t); SNT(t) \rangle, \quad (5)$$

gdzie:

- $A_s = \langle a_1, a_2, \dots, a_i \rangle$ – zbiór identyfikatorów zdolności sensora s , którymi może on dokonywać akwizycji danych (np. odczyt temperatury, wykrycie ruchu, pomiar ciśnienia krwi itp.),
- $O(t) = \{o_a(t) = \langle a, R^{N \times M}, t \rangle; a \in A; N, M \in \mathbb{Z}_+^N; t \in T\}$ – zbiór obserwacji sensora w chwili czasu t ,
- P – zbiór programów wykonywanych przez sensor w trakcie działania,
- $R(t) = \langle r_1(t), r_2(t), \dots, r_i(t) \rangle$ – zbiór wartości określających ilość dostępnych zasobów (np. energii, pamięci operacyjnej, obciążenie procesora itp.) w chwili czasu t ,
- $SNT(t)$ – podsieć sieci SN zawierająca węzły, o których sensor ma pełną wiedzę (tj. lokalizacja, prędkość, zdolności).

Podobnie jak sieć, także sensory nie są stacjonarne w czasie. Na zmianę ich stanu wpływa ruch, wytracanie zasobów oraz zjawiska zachodzące w otaczającym je środowisku. Sensor obserwuje zjawiska przy pomocy swoich zdolności do akwizycji (zbiór A_s). Może to być na przykład pożar, trzęsienie ziemi, ruch (ludzi lub pojazdów) itp. Podobnie jak sensory, także zjawiska opisywane są przez szereg cech i mogą być dowolnie konfigurowane na potrzeby eksperymentu:

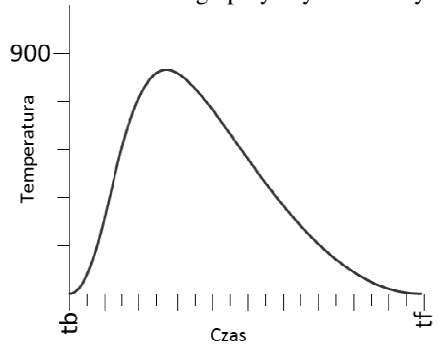
$$e(t) = \langle R(t), EA(t), t_b, t_f, p(t) \rangle \wedge t_b \leq t \leq t_f, \quad (6)$$

przy czym:

- $R(t) \in \psi$ – obszar (zbiór lokalizacji), na którym zaszło zjawisko,
- t_b – chwila czasu rozpoczęcia zjawiska,
- t_f – chwila czasu zakończenia zjawiska,
- $p(t)$ – prawdopodobieństwo zniszczenia węzłów znajdujących się w obszarze zjawiska,
- $EA(t) = \{\langle a, s_a(t) \rangle; a \in A; (s_a(t): T \rightarrow P(R^{N \times M}); N, M \in \mathbb{Z}_+^N)\}$.

Kluczową częścią definicji zjawiska jest zbiór $EA(t)$. Zawiera on elementy, które są dwójkami, składającymi się z indeksu zdolności sensora do akwizycji danych oraz funkcji $s_a(t)$. Funkcja ta opisuje, w jak sposób zjawisko rozwija się w czasie, jednak przez pryzmat określonej zdolności do jego akwizycji. Ściślej rzecz ujmując, jest to multifunkcja z dziedziną, którą jest aktywny czas symulacyjny, oraz wartościami ze zbioru P , będącego rodziną wszystkich podzbiorów zbioru $R^{N \times M}$. Dla przykładu zjawisko pożaru może być obserwowane przez sensory ze zdolnością do odczytu temperatury bądź przez kamery optyczne [3]. W takim przypadku zbiór EA będzie się

składał z dwóch elementów. Pierwszy będzie dotyczył pomiaru temperatury i funkcja $s_a(t)$ będzie przyjmowała wartości ze zbioru $P(R^{1 \times 1})$, czyli upraszczając – z R . Wynika to faktu, iż temperaturę w każdej chwili czasu można wyrazić w postaci pojedynczej liczby rzeczywistej. Rysunek 3 pokazuje, jak mogłaby wyglądać funkcja $s_a(t)$ dla pomiaru temperatury. Drugi element zbioru EA w rozważanym przykładzie odpowiada za obserwację zjawiska przy użyciu cyfrowej akwizycji obrazu. W takim przypadku funkcja $s_a(t)$ przyjmuje wartości ze zbioru $P(R^{N \times M})$, gdzie N oraz M to maksymalna rozdzielczość kamery wyrażana w liczbie pikseli. Jest to następstwo faktu, że obraz cyfrowy można zapisać w postaci uporządkowanego zbioru liczb rzeczywistych o mocy $N \times M$, przy czym każdy z jego elementów opisuje pojedynczy piksel (wartość liczby zależy od sposobu kodowania). Rysunek 3 przedstawia przykład funkcji $s_a(t)$ dla zjawiska obserwowanego przy użyciu kamery cyfrowej.



Rys. 3. Funkcja S_a dla pomiaru temperatury. Źródło: Opracowanie własne

Fig. 3. The S_a function for temperature acquisition. Source: Own preparation

Zjawisko obserwowane jest przez te węzły sieci, które znajdują się w obszarze jego działania. Tak długo, jak długo spełniony jest ten warunek, uzupełniany jest zbiór $O(t)$ obserwacji sensora. W ten sposób generowane są dane sensoryczne, które mogą zasilać systemy fuzji danych. Jednocześnie symulator pozwoli na badanie różnych algorytmów trasowania i wymiany komunikatów pomiędzy sensorami, jak również możliwości odczytywania danych z sieci jako całości (makroprogramowanie).

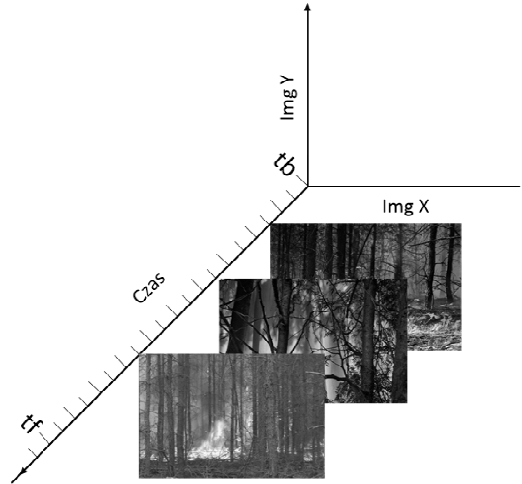
Zjawiska układane są w scenariusz SC , który jest rodziną zbiorów Ev zdefiniowaną jako:

$$SC = \langle Ev_{\tau_1}, Ev_{\tau_2}, \dots, Ev_{\tau_n} \rangle, \tau_1 < \tau_2 \dots < \tau_n, \tau_{1..n} \in T. \quad (7)$$

Zbiór Ev_{τ_k} definiowany jest jako:

$$Ev_{\tau_k} = \{e \in EV : t_{b_e} = \tau_k\}, t_{b_e}, \tau_k \in T, \quad (8)$$

przy czym EV to zbiór wszystkich zjawisk w scenariuszu, natomiast t_{b_e} to chwila rozpoczęcia zjawiska e . Wynika stąd, że scenariusz to rodzina zbiorów grupujących zjawiska rozpoczynające się w tej samej chwili czasu symulacyjnego. Dodatkowo zbiór SC jest uporządkowany rosnąco według wartości czasu rozpoczęcia zjawisk.



Rys. 4. Funkcja S_a dla cyfrowej akwizycji obrazu. Źródło: Opracowanie własne

Fig. 4. The S_a function for digital image acquisition. Source: Own preparation

Sensory mogą przesyłać pomiędzy sobą komunikaty w celu przekazywania pozyskanych obserwacji. W tym kontekście istnieją dwa rodzaje komunikatów:

- komunikaty z zapytaniem o dane,
- komunikaty z danymi.

Komunikat z zapytaniem o dane wysłany od węzła s_s do s_d ma postać:

$$mq_{s,d} = \langle A_q, s_s, s_d \rangle, \quad (9)$$

gdzie A_q określa zdolność, której dane mają zostać pobrane (np. odczyt temperatury). Musi zachodzić $A_q \in A_{s_d}$, czyli zapytanie może dotyczyć tylko tych zdolności, które są dostępne u odbiorcy komunikatu. W odpowiedzi na zapytanie o dane przesyłany jest komunikat z danymi postaci:

$$mq_{s,d} = \langle O(\tau), s_s, s_d \rangle, \quad (10)$$

gdzie $O(\tau)$ to zbiór obserwacji nadawcy w chwili czasu τ .

Przesłanie komunikatu od nadawcy do odbiorcy zajmuje pewien czas, który zależy od rozmiaru komunikatu (w szczególności danych przesyłanych w zbiorze $O(\tau)$) oraz przepustowości łącza wyznaczonej przez funkcję b zdefiniowaną na krawędzi grafu pomiędzy węzłami zaangażowanymi w komunikację. Czas wyznaczany jest przez funkcję $tm: M \times Z \rightarrow T$, gdzie M to zbiór wszystkich komunikatów. Komunikacja jest procesem, który rozpoczyna się w pewnej chwili czasu symulacyjnego τ i kończy najpóźniej w chwili $\tau + tm(m, b(s_s, s_d))$. Proces komunikacji ma trzy możliwe stany:

- w trakcie: 0,
- sukces: 1,
- porażka: -1.

Przyjmując t jako aktualny czas symulacyjny oraz tm_{s_s, s_d} jako czas potrzeby na przesłanie komunikatu pomiędzy węzłem s_s a s_d , warunki przejść pomiędzy stanami są następujące:

- $W_{0,0}: t < \tau + tm_{s_s, s_d} \wedge s_d \in N_{s_s}(t)$,
- $W_{0,1}: t > \tau + tm_{s_s, s_d} \wedge s_d \in N_{s_s}(t)$,
- $W_{0,-1}: t < \tau + tm_{s_s, s_d} \wedge s_d \notin N_{s_s}(t)$.

W przypadku kiedy $s_d \notin N_{s_s}$, czyli odbiorca nie jest sąsiadem nadawcy, komunikat przesyłany jest na zasadzie skoków. Trasa między nadawcą a odbiorcą definiowana jest jako:

$$R_{s_s, s_d}(t) = \langle s_s, s_1, \dots, s_k, s_d \rangle, s_s, s_1, \dots, s_k, s_d \in S, s_1 \in N_{s_s}, s_k \in N_{s_d}, \\ \forall s_l (s_{l-1} \in N_{s_l}, s_{l+1} \in N_{s_l}), l = 2, \dots, k-1. \quad (11)$$

Długość trasy wyznaczana jest jako moc zbioru $R_{s_s, s_d}(t)$ pomniejszona o węzły nadawcy i odbiorcy:

$$RL_{s_s, s_d}(t) = |R_{s_s, s_d}| - 2. \quad (12)$$

W przypadku komunikacji z wykorzystaniem skoków każdy węzeł podejmuje decyzję, do kogo przesłać komunikat w następnym kroku. Zmienną decyzyjną jest więc zbiór sąsiadów węzła wysyłającego komunikat:

$$X = \{s \in N_{s_s}\}. \quad (13)$$

Dane, na podstawie których podejmowana jest decyzja, to podzbiór węzłów sieci, o których nadawca ma pełną wiedzę:

$$D = SNT_{s_s}(t). \quad (14)$$

Kryteria, na podstawie których wybierane są kolejne węzły do wykonania skoku, zależą od zastosowanego algorytmu trasowania. W tym kontekście algorytm trasowania jest definiowany przez:

- zbiór kryteriów $W(D, X)$,
- funkcję wyznaczającą zbiór sensorów sąsiadujących, do których ma zostać przesłany komunikat:

$$rt_{s_s, s_d}(s_s, s_d, a, W(D, X)) \rightarrow \{s \in S\}. \quad (15)$$

Uwzględniając powyższe rozważania, należy stwierdzić, że rozszerzenie symulatora o nowy algorytm trasowania wymaga zdefiniowania:

- kryteriów: $W(D, X)$,
- postaci funkcji: $sn(s_s, t) \rightarrow SNT_{s_s}(t)$, której celem jest wyznaczenie podzbioru węzłów sieci, o których nadawca ma pełną wiedzę,
- postaci funkcji: rt_{s_s, s_d} ,
- programu $p \in P$, który realizuje trasowanie w każdym z węzłów.

5 Architektura symulatora

Symulator zaprojektowany został w architekturze rozproszonej z wykorzystaniem przetwarzania w tzw. *chmurze*. Najnowsze rozwiązania mechanizmów zarządzania czasem i przebiegiem symulacji odchodzą od wielowątkowości na rzecz *mechanizmu kontynuacji*. Mechanizm ten ma niebagatelna zaletę – pozwala na zapis/odczyt stanu symulatora w różnych chwilach czasowych. Praktyczne zastosowania koncepcji kontynuacji znaleźć można między innymi w serwerze kontynuacyjnym Seaside, a także Apache Commons Javaflow Component. Proponuje się w prezentowanym rozwiązaniu wdrożenie komponentów Javaflow w warstwie zarządzania symulacją, która ulokowana jest na serwerze aplikacyjnym lokalnej chmury. Przyjęcie modelu przetwarzania w chmurze daje szereg możliwości, m.in.: instalowania, konfigurowania i uruchamiania aplikacji symulacyjnych poprzez interfejs przeglądarki, współdzielenia aplikacji przez wielu użytkowników, dostępności wyników ich działania z przenośnych lub stacjonarnych komputerów/smartfonów, czy wreszcie uniezależnienia od platformy uruchomieniowej, co realnie zapewnia dostęp do aplikacji z dowolnego urządzenia posiadającego dostęp do Internetu oraz obsługę języka Java. Takie podejście umożliwi również efektywne symulowanie sieci, w której węzłami są urządzenia działające pod kontrolą systemu operacyjnego Android. Moc obliczeniowa dostępna w chmurze pozwoli na to, aby każdy sensor posiadał indywidualną instancję systemu operacyjnego, na którym uruchamiane będą aplikacje wykorzystywane w eksperymencie. Dzięki temu aplikacje uruchamiane w węzłach sieci będą pełnoprawnymi programami, co powinno istotnie ograniczyć wysiłek wdrożenia ich w sieci składającej się z rzeczywistych urządzeń. Inną istotną cechą architektoniczną jest zastosowanie wtyczek, których dołączenie rozbudowywać będzie możliwości symulatora. Wtyczki dostarczać mogą nowych algorytmów ruchu sensorów, trasowania oraz mechanizmów makroprogramowania sieci.

6 Podsumowanie

W niniejszym opracowaniu przedstawiony został model sieci sensorowej, który jest podstawą do budowy symulatora. Obecnie istniejące rozwiązania w tej dziedzinie skupiają się przede wszystkim na modelowaniu aspektów komunikacyjnych pomiędzy sensorami, przez co użycie ich na potrzeby systemów fuzji danych jest trudne lub wręcz niemożliwe. Proponowany model w mniejszym stopniu skupia się na bezprzewodowej komunikacji pomiędzy sensorami, a przede wszystkim pozwala na umieszczenie sieci w środowisku, w którym zachodzą zjawiska obserwowane przez sensory. Mogą być one dowolnie konfigurowane, jak również sensory mogą posiadać wiele zdolności do akwizycji danych, również nieznanymi w chwili obecnej.

Symulator oparty na prezentowanym modelu zaprojektowany został w sposób umożliwiający jego łatwą konfigurację i rozszerzenie o nowe algorytmy trasowania bądź ruchu węzłów. Proponowane rozwiązanie, wykorzystujące moc obliczeniową chmury dla symulowania węzłów, powoduje, że z każdym z nich może być związana maszyna wirtualna, na której uruchomiony jest rzeczywisty system operacyjny. Daje to możliwość uruchamiania przez sensory programów w trakcie działania symulacji. W konsekwencji proponowany symulator stanowić będzie wsparcie dla systemów stosujących różne architektury fuzji danych, w tym fuzję wektorów cech, jak i fuzję decyzji wypracowanych przez sensory.

Literatura

1. Najgebauer A., Dyk M.: *Sieci sensorowe dla potrzeb pozyskiwania danych w symulacji wielorozdzielczej*. Sensors, 2010
2. Hall D. L., Llinas J.: *Handbook of Multisensor Fusion: Theory and Practice - Second Edition*, ed. Martin Liggins, CRC Press, Boca Raton, FL 2009
3. Šerić L., Stipaničev D., Štula M.: Agent Based Sensor and Data Fusion in Forest Fire Observer .In: *Sensor and Data Fusion*, Dr. ir. Nada Milisavljević (Ed.), In-Teh, 2009, ISBN 978-3-902613-52-3
4. Schmid S., Wattenhofer R.: Modeling sensor networks. In: *Algorithms and Protocols for Wireless Sensor Networks*, Boukerche, A., John Wiley & Sons, 2008
5. Al-Karaki J. N., Kamal A. E.: Routing techniques in wireless sensor networks: a survey. In: *Wireless Communications*, IEEE, vol.11, no.6, pp. 6 {28, doi: 10.1109/MWC.2004.1368893, 2004
6. Clark B. N., Colbourn C. J., Johnson D. S.: Unit disk graphs. In: *Discrete Mathematics* 86 (13): 165177, 1990

Streszczenie

W pracy przedstawiono model sieci sensorowej będący podstawą dla budowy symulatora, który może być wykorzystany do generowania sytuacji oraz danych na potrzeby systemów fuzji danych. Model, a w konsekwencji symulator, pozwala na zbudowanie heterogenicznej sieci (lub wielu sieci), której sensory obserwują zmieniające się w czasie zjawiska. Zaletami proponowanego modelu są: możliwość szerokiej konfiguracji węzłów, które posiadać mogą wiele zdolności do obserwacji środowiska, również uznawanych obecnie za futurystyczne, oraz łatwa konfigurowalność algorytmów trasowania i komunikacji bezprzewodowej. Budowany w oparciu o przedstawiony model symulator będzie również umożliwiał modyfikację zachowania sensorów poprzez ich makroprogramowanie.

Słowa kluczowe: sieci sensorowe, fuzja danych, symulacja dyskretna zdarzeniowa

Simulation model of sensor network for data fusion

Summary

Paper presents wireless sensor network model which is base for desining and implementation of a simulator, which can be used by data fusion systems. The model, and in consequence the simulator, allows user to build a heterogenous, wireless network. Its sesors can observe phenomena, which may change during simulation process. Proposed model has some advantages. First of all, sensors can be widely configured. They can have multiple observaional capabilities, also those that are considered futuristic.

Keywords: sensor networks, data fusion, discrete event simulation