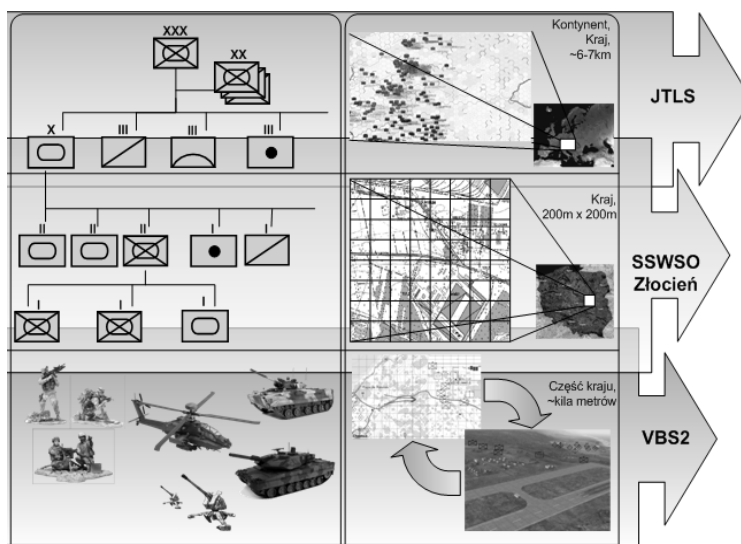


**Andrzej NAJGEBAUER, Michał DYK**  
Wojskowa Akademia Techniczna, Wydział Cybernetyki  
00-908 Warszawa, ul. Gen. Sylwestra Kaliskiego 2  
E-mail: anajgebauer@wat.edu.pl, mdyk@wat.edu.pl

## Sieci sensorowe dla potrzeb pozyskiwania danych w symulacji wielorozdzielczej

### 1 Wstęp

Współczesne systemy symulacyjne, w szczególności wojskowe, mają bardzo często strukturę hierarchiczną oraz cechują się różną rozdzielczością składowych symulatorów. Przykład środowiska symulacyjnego o takiej strukturze prezentuje poniższy diagram:



Rys. 1. Hierarchia symulatorów w środowisku wielorozdzielczym

Fig. 1. Example of simulators' hierarchy in multiresolution environment

Środowisko to jest obiektem badań w Instytucie Systemów Informatycznych Wydziału Cybernetyki WAT. Symulatory składowe mają różne poziomy rozdzielczości danych. Najniżej w hierarchii znajduje się VBS2, który symuluje działanie pojedynczych żołnierzy i pojazdów. Wygenerowane przez niego dane podlegają agregacji i przekazywane są „wyżej” do SSWSO Złocień, który w tym ujęciu ma niższą rozdzielczość danych. Na szczycie hierarchii znajduje się symulator JTLS, który w tym środowisku ma najniższą rozdzielczość danych.

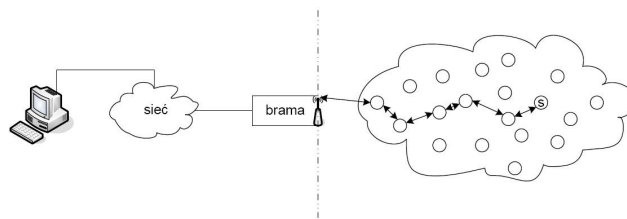
Symulowanie działań zagregowanych oddziałów (kompania, batalion, dywizja) odbywa się zazwyczaj przy użyciu stochastycznych algorytmów przemieszczania i wytracania wojsk, których adekwatność jest (lub może być) zweryfikowana w oparciu o statystyczne

dane rozegranych starć i konfliktów (jak np. model Lanchestera, którego poprawność została udowodniona m. in. w oparciu o dane z bitwy o wyspę Iwo Jima). W wyniku ich działania modyfikowany jest zagregowany stan oddziału jak liczność czy położenie. Dużo większym wyzwaniem jest symulowanie działań pojedynczych żołnierzy oraz pojazdów. Na komplikację wpływają kwestie wydajnościowe oraz modelowe. Symulowane obiekty są złożone (opisywane dużą liczbą parametrów i uczestniczą w wielu procesach) i zazwyczaj występują w dużej liczbie (np. żołnierze na polu walki). Stawia to wyzwania wydajnościowe, które pozwala rozwiązywać ciągle rosnąca moc obliczeniowa i rozwój systemów rozproszonych. Najtrudniejszym zagadnieniem jest poprawne zamodelowanie ludzkich zachowań, których nie da się łatwo opisać parametrami liczbowymi.

Rozwiązaniem problemów z symulowaniem pojedynczych obiektów może być przeprowadzenie symulacji rzeczywistej, w której udział biorą prawdziwi żołnierze. Powstaje jednak problem gromadzenia danych generowanych ramach takiej symulacji. Propozycją rozwiązania jest potraktowanie żołnierzy jako sieci sensorowej. W takim podejściu każdy z ćwiczących wyposażony jest w urządzenie komunikacyjne, które stanowi węzeł sieci. Gromadzenie danych na potrzeby symulatorów wyższych w hierarchii (mających niższą rozdzielczość) odbywa się poprzez zapytania do sieci, co oznacza, że każdy z węzłów odpytywany jest o aktualny stan. W przypadku wysyłania danych do sieci, np. przekazanie rozkazu z dowództwa kompanii, stosowny komunikat wysyłany jest do zainteresowanych węzłów.

## 2 Sieci sensorowe

Sieć sensorową najogólniej można określić jako sieć złożoną w wielu, najczęściej więcej niż kilkunastu, urządzeniach (zazwyczaj o małych rozmiarach), które rozmieszczone są na ustalonym obszarze i realizują wspólne zadanie. 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” (lub „ujście”), czyli element do którego przesyłane są zgromadzone dane lub z którego wysyłane są zapytania do sieci. Ogólny schemat sieci sensorowej przedstawia poniższy rysunek:



Rys.2. Przykład sieci sensorowej

Fig. 2. Example of sensor network

Przesyłanie danych pomiędzy bramą a sensorami (lub w razie potrzeby tylko pomiędzy węzłami) odbywa się zazwyczaj metodą „skoków”. Jeśli węzłem będący źródłem komunikatu nie ma bezpośredniego połączenia z bramą przesyła dane do węzła

sąsiedniego, który z kolei przekazuje komunikat kolejnym sensorom. W ten sposób dane „skaczą” pomiędzy węzłami aż dotrą do ujścia.

Pierwsze prace nad sieciami sensorowymi rozumianymi w przedstawiony powyżej sposób prowadziła agencja DARPA w latach 80-tych XX wieku. Ówczesne badania miały na celu zbudowanie sieci inteligentnych czujników, która mogłaby zastąpić systemy kontrolno-pomiarowe z węzłem centralnym wykonującym wszystkie obliczenia. Doświadczenia z tych prac zostały wykorzystane w projekcie SensIT [11] (również realizowanego przez DARPA), który był dużym krokiem naprzód w dziedzinie sieci sensorowych. W jego ramach opracowana została bezprzewodowa sieć, która posiadała możliwość samoorganizacji oraz miała zaimplementowane zaawansowane algorytmy trasowania, uwzględniające redundancję ścieżki przekazywania danych. Najważniejszym osiągnięciem było to, że powstała sieć stanowiła kompletny system, który autonomicznie pozyskiwał dane, dokonywał ich analizy i odpowiadał na zapytania. Całość funkcjonowała jedynie w oparciu o równorzędne sensory. Sieć zbudowana w ramach projektu SensIT była wykorzystywana do monitorowania i śledzenia ruchu na określonym obszarze.

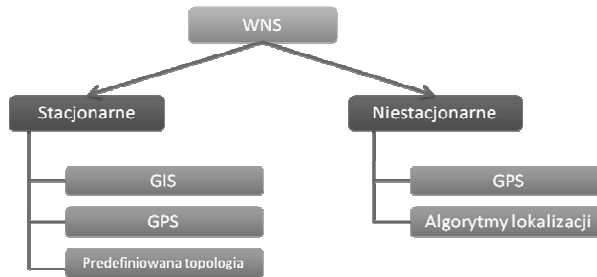
Obecnie (bezprzewodowe) sieci sensorowe mają bardzo szerokie spektrum zastosowań, z których jako przykłady można wymienić:

- Monitorowania zjawisk przyrodniczych,
- Projekt IceCube- sieć sensorów wykrywająca neutrina przelatujące przez naszą planetę (<http://icecube.wisc.edu/>),
- Projekt CodeBlue- monitorowanie stanu zdrowia osób zagrożonych chorobami układu krążenia,
- Monitorowanie powierzchni Marsa (projekt Tumbleweed) [8].

Jak w każdej sieci tak i w sieciach sensorowych dla poprawnego funkcjonowania konieczne jest zdefiniowanie algorytmu(ów) trasowania (routingu). W metodach wybierania trasy innych niż tzw. zalewanie sieci (często określane jako broadcast lub flooding) konieczna jest znajomość jej topologii. Oznacza to, że muszą być zdefiniowane metody, które pozwolą na wyznaczenie położenia węzłów i zbudowanie na tej podstawie topologii. Kolejne dwa rozdziały przedstawiają propozycje algorytmów lokalizacji węzłów oraz trasowania.

### 3 Autolokalizacja sensorów w oparciu o metodę Monte Carlo

Możliwości lokalizacji węzłów w sieciach sensorowych zależą od rodzaju sieci. Najistotniejszy w tym ujęciu jest podział sieci na stacjonarne (topologia nie zmienia się w czasie), oraz niestacjonarne (topologia może się zmieniać w czasie). Prezentuje to poniższy diagram



Rys.3. Klasyfikacja sieci sensorowych

Fig. 3. Classification of sensor networks

W przypadku sieci, których topologia nie zmienia się w czasie lokalizacja węzłów ustalana jest zazwyczaj przed rozpoczęciem działania. Można w tym celu wykorzystać systemy GIS lub GPS albo ustalić położenie każdego sensora z osobna w momencie ich rozmieszczania (predefiniowana topologia). W przypadku sieci niestacjonarnych węzły mogą poznać swoje położenie za pośrednictwem systemu GPS lub przy użyciu algorytmów autolokalizacji. Naturalnie sieć sensorowa złożona z ćwiczących żołnierzy należy do klasy sieci niestacjonarnych, stąd konieczność aktualizacji położenia węzłów. Dla zwiększenia autonomiczności i niezawodności sieci proponujemy zastosować w tym celu algorytmy autolokalizacji. Wśród licznych propozycji takich algorytmów ([3], [6], [7]) szczególnie interesujące wydają się być rozwiązania, w których węzły symulują swój ruch i położenie wykorzystując do tego sekwencyjną metodę Monte Carlo (SMC) [2]. Szkic algorytmu wraz z założeniami przedstawiony jest poniżej:

Założenia:

Węzły rozmieszczone są losowo i nie znają swojego początkowego położenia,  
 Lokalizacja przedstawiana jest w dwóch wymiarach (dla uproszczenia przykładu),  
 Czas jest dyskretny,  
 Znana jest maksymalna prędkość poruszania się węzłów.

Algorytm:

1. Inicjalnie ( $t=0$ ) losowany jest  $N$  - elementowy podzbiór lokalizacji ze wszystkich możliwych na danym obszarze (zbiór  $L_0$ ),
2. W każdej chwili czasu  $t \neq 0$  dla każdego elementu ze zbioru  $L_t$ , wykonywane są kroki:
  1. **Prognoza** (ustalenie wszystkich możliwych lokalizacji węzła w czasie  $t$  w odniesieniu do lokalizacji w czasie  $t-1$ ).
  2. **Filtrowanie** (odrzućenie lokalizacji nieprawdopodobnych w chwili  $t$  na podstawie informacji o węzłach sąsiednich w chwili  $t-1$ ).
3. Ustalenie nowej lokalizacji jako średniej spośród wszystkich elementów zbioru  $L_t$

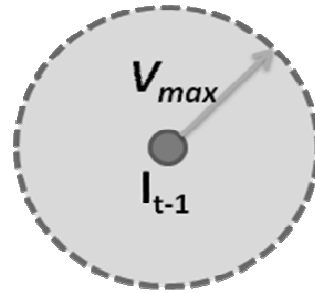
Istotą algorytmu jest wyznaczenie prawdopodobieństwa  $p(l_t|l_{t-1})$ , czyli znajdowania się węzła sieci w położeniu  $l_t$  pod warunkiem, że w poprzedniej rozpatrywanej chwili czasu znajdował się w położeniu  $l_{t-1}$ . Dzieje się to w dwóch krokach (fazach) nazywanych Prognozą i Filtrowaniem.

W ramach Prognozy wyznaczone są wszystkie możliwe lokalizacje węzła w chwili  $t$  w oparciu o wyznaczone lokalizacje w chwili  $t-1$ . Założenia dla tej fazy są następujące:

- Węzeł nie zna swojej prędkości ani kierunku ruchu a jedynie prędkość maksymalną  $V_{\max}$ ,
- Prędkość węzła z równym prawdopodobieństwem przyjmuje wartości z przedziału  $[0, V_{\max})$ ,
- Jako miarę odległości przyjmuje się miarę Euklidesową  $d(x,y)$ .

Uwzględniając powyższe można stwierdzić, że zbiór możliwych lokalizacji węzła zawiera się w kole o promieniu równym prędkości maksymalnej i środku w punkcie będącym wyznaczonym położeniem w chwili  $t-1$ . Prezentuje to poniższy Rys. 4. Ze zbioru prawdopodobnych lokalizacji losowana jest jedna dla każdego elementu ze zbioru  $L_t$ . Jak widać tak wyznaczone położenie jest mało dokładne i nie uwzględnia w żaden sposób charakterystyki ruchu węzłów. W celu zwiększenia dokładności estymacji położenia wykonywany jest krok algorytmu odpowiedzialny za filtrowanie znalezionych lokalizacji.

$$p(l_t | l_{t-1}) = \begin{cases} \frac{1}{\pi V_{\max}^2} & \text{dla } d(l_t, l_{t-1}) < V_{\max} \\ 0 & \text{dla } d(l_t, l_{t-1}) \geq V_{\max} \end{cases}$$

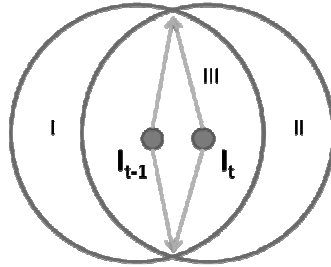


Rys.4. Możliwe lokalizacje sensora w chwili  $t$

Fig.4. Possible locations of a sensor at time  $t$

Faza filtrowania ma na celu odrzucenie ze zbioru  $L_t$  wszystkich lokalizacji nieprawdopodobnych, uwzględniając przy tym interakcję sensora z innymi węzłami sieci. Zakłada się że każdy węzeł w chwili  $t$  wysyła specjalny komunikat, w którym przekazuje swój identyfikator oraz historię swoich lokalizacji. Taka informacja dociera do wszystkich sąsiadów (węzłów będących w zasięgu radiowym) nadawcy. Dzięki temu w każdej chwili czasu  $t$ , sensor wie z jakimi węzłami sąsiaduje. Uwzględniając ruch można zdefiniować cztery rodzaje węzłów sąsiadujących:

- zewnętrzne – węzły niesłyszalne zarówno w chwili  $t$  jak i  $t-1$ ,
- wchodzące – węzły niesłyszalne w chwili  $t-1$  ale słyszalne w chwili  $t$ ,
- wychodzące – węzły słyszalne w chwili  $t-1$  ale niesłyszalne w chwili  $t$ ,
- wewnętrzne – węzły słyszalne zarówno w chwili  $t-1$  jak i  $t$ .



Rys.5. Obszary zawierające węzły wychodzące, wewnętrzne i wchodzące przy zmianie lokalizacji

Fig.5. Areas containing outgoing, internal and entering nodes, when changing location

Powyższy rysunek przedstawia jak ruch sensora wpływa na kształtowanie się obszarów w których znajdują się sąsiadujące węzły określonych rodzajów. Zakładając, że sensor zmienił położenie z  $l_{t-1}$  na  $l_t$  wszystkie węzły znajdujące się w obszarze I są wychodzące. Obszar III zawiera węzły wewnętrzne, natomiast w obszarze II znajdują się węzły wchodzące. Najwięcej informacji niosą ze sobą wchodzący i wychodzący. Dzięki nim sensor może estymować jaki dystans udało mu się przebyć. Każdy sensor przechowuje historię sąsiedztwa, stąd może wnioskować która lokalizacja jest prawdopodobna, a która nie. Dodatkowo może na tej podstawie estymować kierunek i zwrot swojego ruchu. Sposób wnioskowania zależy w dużej mierze od danych jakie pojedynczy sensor otrzymuje od swoich sąsiadów. Jeśli dopuszczalne jest większe zużycie energii przez sensory, mogą one przekazywać, poza obecnym i historycznym położeniem, także informację o swoich sąsiadach i ich lokalizacjach. W przypadku ćwiczeń poligonowych ważnym kryterium poprawności położenia sensora może być formacja w jakiej poruszają się żołnierze lub pojazdy, co może zwiększyć dokładność estymacji.

Kroki Prognozy i Filtrowania wykonywane są tak długo aż zebrane zostanie co najmniej  $N$  poprawnych (w tym znaczeniu możliwych) lokalizacji. Estymowane położenie węzła w chwili  $t$  wyznaczone jest jako średnia ze zbioru  $L_t$ .

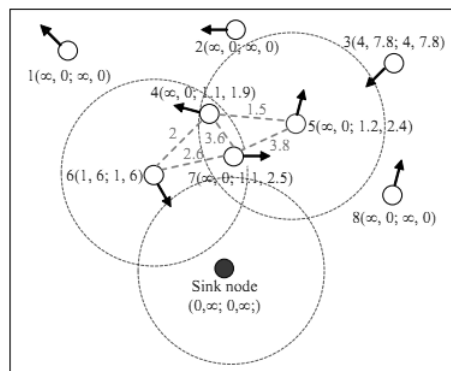
### 3 Algorytmy trasowania

Sieć sensorowa podobnie jak szeroko znane i rozpowszechnione sieci LAN dla poprawnego funkcjonowania potrzebują algorytmów trasowania (routingu). Odpowiadają one za dostarczenie danych od źródła komunikatu (nadawcy) do jego ujścia (odbiorcy). W sieciach sensorowych nie ma dedykowanych urządzeń, które odpowiadają za trasowanie a wiedza węzłów o topologii jest zazwyczaj mocno ograniczona, stąd algorytmy routingu różnią się zasadniczo od tych znanych z sieci LAN. Można je podzielić na trzy grupy (w oparciu o [5]):

- Algorytmy oparte na lokalizacji węzłów (Location-based) – trasa przesyłania komunikatu wyznaczana jest w oparciu o znanane położenie węzłów;
- Algorytmy zorientowane na dane (Data-centric) – węzły są różnicowane pod względem rodzaju danych jakiego oczekują i jaki dostarczają, trasa przesyłu danych nie jest istotna;

- Algorytmy hierarchiczne (Hierarchical) – w trakcie działania sieci wybierany jest węzeł (lub wiele węzłów), który gromadzi dane i przesyła je do ujścia. Węzły wyróżnione najczęściej zmieniają się w czasie.

W przypadku zastosowania sieci sensorowej dla ćwiczeń poligonowych (lub ogólnie symulacji rzeczywistej z udziałem ludzi) zidentyfikować można dwa różne scenariusze pozyskiwania i wysyłania danych z i do sieci. W jednym przypadku zapytanie o dane wysyłane jest do pojedynczego węzła lub niewielkiej ich grupy. Proponowanym rozwiązaniem w tej sytuacji jest zastosowanie algorytmu opartego o lokalizację węzłów. Zapewni on to, że wyznaczona zostanie trasa do zainteresowanego węzła (lub węzłów), bez niepotrzebnego angażowania pozostałych sensorów (poza koniecznością wykonania skoku). Istnieje szereg algorytmów wyznaczających drogę przesyłania komunikatu w oparciu o lokalizację węzłów. Szerszy ich opis można znaleźć w opracowaniach [4] oraz [5]. Wiele z nich nie uwzględnia jednak ruchu sensorów a tym bardziej sytuacji w której wyznaczona trasa z powodu przemieszczenia węzłów nie jest już dostępna, co występować będzie w przypadku ćwiczenia poligonowego. W takim przypadku można mówić o szczególnym rodzaju sieci sensorowej nazywanej Delay Tolerant Mobile Sensor Network. Przykładem algorytmu uwzględniającego ruch węzłów i zmianę tras komunikacyjnych jest algorytm Minimum Expected Delay-based Routing (MEDR) przedstawiony w opracowaniu [1]. W tym rozwiązaniu każdy z węzłów posiada dwa istotne parametry: minimalne spodziewane opóźnienie (MED- Minimum Expected Delay) oraz czas jego ważności. MED określa minimalny czas, po którym komunikat od węzła dotrze do ujścia (w przypadku skoków może to być inny węzeł). Ponieważ w wyniku ruchu zmienia się także topologia sieci algorytm wylicza okres czasu przez który określona ścieżka jest dostępna (czyli jak długo węzły znajdują się w zasięgu radiowym). Poniższy rysunek przedstawia schemat działania protokołu MEDR:



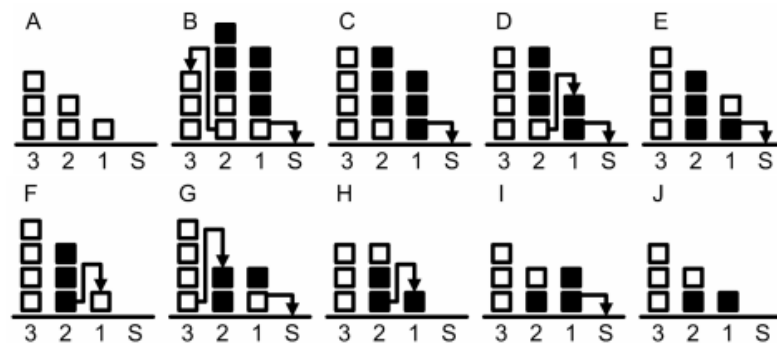
Rys.6. Przykład działania algorytmu MEDR

Fig.6. Example of routing process by MEDR

Okręgi reprezentują zasięg sygnału radiowego (naniesiony jedynie dla najistotniejszych węzłów) a skierowane strzałki wskazują kierunek i zwrot ruchu sensora. Przerywane linie pomiędzy węzłami przedstawiają możliwą trasę przesyłania danych. Każda z tras posiada wyznaczony okres czasu przez który połączenie istnieje. Dodatkowo w nawiasach przedstawione są parametry węzłów: spodziewany czas skontaktowania się z ujściem

(na rysunku Sink node), spodziewany czas utraty kontaktu z ujściem, MED oraz czas obowiązywania MED (szczegóły wyznaczania tych wartości przedstawione zostały w [1]). Spośród wszystkich węzłów najmniejszy oczekiwany czas po upływie którego nastąpi komunikacja z ujściem (pierwsza wartość w nawiasie) ma sensor 6. Począwszy od niego następuje budowanie możliwych tras. Zakładając, że źródłem komunikatu jest węzeł 5, możliwe trasy przesłania komunikatu to 5→4→6, 5→7→2→6 oraz 5→7→6. Uwzględniając liczbę skoków i czas wygaśnięcia łącza powinna zostać wybrana ostatnia trasa. MEDR oraz podobne algorytmy, choć skuteczne wymagają jednak dużo danych o topologii sieci i ruchu sensorów. Można je uzyskać stosując sposób autolokalizacji węzłów, którego koncepcja przedstawiona została w rozdziale 3.

Drugim scenariuszem pozyskiwania danych z sieci jest odpytywanie o stan wszystkich jej sensorów. Może to mieć miejsce kiedy potrzebna jest informacja o stanie całego ćwiczącego ugrupowania (np. zapytanie o pozostałą ilość amunicji). Najprostszym sposobem realizacji tego zadania jest zastosowanie metody zalewowej, co oznacza że każdy węzeł wysyła do wszystkich sąsiednich komunikat z informacją o swoim stanie i taka informacja jest propagowana aż dotrze do ujścia. Generuje to bardzo duży ruch sieciowy, w którym tylko pewna część wysłanych danych dociera do miejsca przeznaczenia. Rozwiązaniem w takiej sytuacji może być zastosowanie algorytmu routingu należącego do klasy określanej mianem „backpressure routing”. Opis tego rodzaju rozwiązań znaleźć można w opracowaniach [12] oraz [13]. Poniższy rysunek przedstawia ogólną koncepcję stojącą za backpressure routing (spotkać się można z przewrotnymi określeniami, że jest to trasowanie bez wyznaczania trasy):



Rys.7. Idea funkcjonowania backpressure routing

Fig.7. How a backpressure routing works

W przykładzie występują cztery węzły, z czego jeden stanowi ujście danych. Każdy z nich posiada kolejkę w której przechowuje komunikaty. W kroku A sieć znajduje się w stanie równowagi, czyli każdy węzeł bliżej ujścia ma o jeden komunikat w kolejce mniej niż poprzedni. W kroku B do węzła 1 i 2 dotarły po trzy nowe komunikaty. Ponieważ zachwiany został stan równowagi, więc węzły starają się do niego powrócić. Realizują to poprzez przesłanie nadmiarowego komunikatu do tego z sąsiednich węzłów, który ma krótszą kolejkę. W przedstawionej sytuacji węzeł 2 przekazuje jeden komunikat do węzła 3, natomiast węzeł 1, ponieważ ma bezpośredni kontakt z ujściem, przekazuje



do niego dane. Zmniejszanie długości kolejki przez pierwszy węzeł (krok C) powoduje, że węzeł 2 zaczyna przekazywać do niego komunikaty (krok D). Kurcząca się kolejka węzła 2 powoduje że węzeł 3 zaczyna przekazywać do niego elementy ze swojej kolejki (krok G) aż sieć znów uzyska stan równowagi (krok J). Zastosowanie koncepcji backpressure routing pozwala w wydajny sposób zebrać dane z całej sieci. W przeciwieństwie do metody flooding (broadcast) tylko niektóre komunikaty są nietrafione (jak pokazano w kroku B) i sieć szybko ustala tzw. gradient spływu komunikatów [13] do ujścia.

#### 4 Podsumowanie

W pracy opisana została koncepcja wykorzystania sieci sensorowych jako narzędzia pozyskiwania danych z symulacji rzeczywistej, która może mieć postać ćwiczeń poligonowych. W zaproponowanym rozwiązaniu każdy z żołnierzy i pojazdów byłby wyposażony w urządzenie komunikacyjne typu PDA z modułem WIFI, stanowiące węzeł sieci (sensor).

Charakterystyka ćwiczeń poligonowych powoduje, że sieć nie ma stałej topologii i zmienia się ona w sposób niekontrolowany na skutek przemieszczania sensorów (żołnierzy i pojazdów). Znajomość lokalizacji węzłów jest jednocześnie konieczna do wyznaczania tras przesyłania komunikatów. Dla rozwiązania tego problemu zaproponowane zostało wykorzystanie algorytmów autolokalizacji, które wykorzystują metodę Monte Carlo jako sposób symulowania położenia sensorów w czasie. Specyfika sieci wymaga algorytmów routingu, które uwzględniają ciągłą zmianę topologii i zapewniają dostarczenie komunikatu także w sytuacji ruchu sensorów. W tym zakresie przedstawiony został algorytm routingu MEDR, który może być punktem wyjścia dla opracowania adekwatnego sposobu trasowania. Na potrzeby zebrania danych z całej sieci zaproponowano wykorzystanie algorytmów routingu typu „backpressure routing”, które zapewniają dostarczenie komunikatów od wszystkich sensorów nieskim kosztem, gdyż nie ma potrzeby wyznaczania do tego celu położenia i tras.

#### Literatura

1. Yong Feng, Ming Liu, Xiaomin Wang and Haigang Gong: Minimum Expected Delay-Based Routing Protocol (MEDR) for Delay Tolerant Mobile Sensor Networks. Sensors 2010
2. Lingxuan Hu David Evans: Localization for Mobile Sensor Networks. Tenth Annual International Conference on Mobile Computing and Networking (MobiCom 2004)
3. W. Burgard, D. Fox, D. Hennig, and T. Schmidt, Estimating the absolute position of a mobile robot using position probability grids, Fourteenth National Conference on Artificial Intelligence (AAAI-96), pp. 896-901, 1996
4. Shio Kumar Singh, M P Singh, and D K Singh: Routing Protocols in Wireless Sensor Networks – A Survey, International Journal of Computer Science & Engineering Survey (IJCSSES) Vol.1, No.2, November 2010
5. Debnath Bhattacharyya, Tai-hoon Kim, Subhajit Pal: A Comparative Study of Wireless Sensor Networks and Their Routing Protocols, Sensors 2010, 10, 10506-10523
6. P. Bergamo, G. Mazzini: Localization In Sensor Networks With Fading And Mobility

7. Lance Doherty, Kristofer S. J. Pister, Laurent El Ghaoui: Convex Position Estimation in Wireless Sensor Networks, Dept. of Electrical Engineering and Computer Sciences – University of California, Berkeley
8. Jeffrey Antol, Philip Calhoun, John Flick, Gregory A. Hajos, Robert Kolacinski, David Minton, Rachel Owens and Jennifer Parker: Low Cost Mars Surface Exploration: The Mars Tumbleweed. NASA Langley Research Center. NASA/TM-2003-212411. August 2003
9. Frank Dellaert, Dieter Fox, Wolfram Burgard and Sebastian Thrun: Monte Carlo Localization for Mobile Robots. IEEE International Conference on Robotics and Automation (ICRA). May 1999
10. Arnaud Doucet, Nando de Freitas and Neil Gordon: An Introduction to Sequential Monte Carlo Methods. In Sequential Monte Carlo Methods in Practice, eds. Arnaud Doucet, Nando de Freitas and Neil Gordon. 2001
11. Srikanta Kumar, Ph.D, David Shepherd: SensIT: Sensor Information Technology For the Warfighter
12. Michael J. Neely: Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity, Conference on Information Sciences And Systems (CISS), March 2006
13. Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, Omprakash Gnawali: Routing Without Routes: The Backpressure Collection Protocol, University of Southern California, Los Angeles, CA, Stanford University, Palo Alto, CA

### Streszczenie

W zagadnieniu symulacji wielorozdzielczej dużym wyzwaniem jest akwizycja danych najwyższej rozdzielczości. Problem jest szczególnie wyraźny w przypadku współpracy systemów symulacji komputerowej oraz rzeczywistej. W rozważanej sytuacji zakłada się współpracę symulatorów wojskowych różnych poziomów z ćwiczeniami poligonowymi. Pomysłem na rozwiązanie tego problemu jest zastosowanie sieci sensorowych jako narzędzia komunikacji.

Dla działania sieci sensorowej konieczne jest opracowanie algorytmów trasowania i lokalizacji węzłów. Zadanie to jest szczególnie trudne jeśli węzły są w ruchu a ich dokładna lokalizacja nie jest znana, co ma miejsce w przypadku ćwiczenia poligonowego. Praca opisuje wymagania jakie powinny spełniać te algorytmy oraz dokonuje analizy obecnie funkcjonujących rozwiązań pod tym kątem.

## **Sensor networks for data acquisition in the problem of multiresolution simulation**

### Summary

In the problem of multiresolution simulation the highest-resolution data acquisition is a challenge. The problem is particularly important in the case of cooperation of computer simulation and real-world. In the considered situation is assumed to military co-operation simulation of different levels of real exercise. The idea to solve this problem is to use sensor networks as a communication tool.

*Sieci sensorowe dla potrzeb pozyskiwania danych w symulacji wielorozdzielczej*

For a sensor network operation is necessary to develop algorithms for routing and location of nodes. This task is especially difficult if the nodes are moving and their location is unknown, as is the case for exercise traverse. The paper describes the requirements to be met by these algorithms and analyzes the currently existing solutions.