

Przykład działania algorytmu adaptacyjnego diagnozowania sieci on-line

- system składa się z 4 węzłów: $nodenum=4$;
- każdy węzeł może komunikować się z każdym: struktura komunikacyjna jest reprezentowana przez graf pełny składający się z 4 wierzchołków lub macierz sąsiedztwa 4×4 złożoną z samych jedynek z zerową przekątną;
- do systemu włączane są kolejno węzły o numerach od 0 do 3

1. Uruchomiony węzeł 0 wykonuje procedurę inicjacji węzła i przechodzi w stan testowania okresowego sąsiadów. Struktury danych węzła 0 po wykonaniu procedury inicjującej wyglądają następująco:

Nodeid	0			
Nodenum	4			
Event	-1	-1	-1	-1
Diagnosis	0	1	1	1
tests	2	1	1	1

Wartości w tablicach diagnosis i tests oznaczają:
0 – sprawny (*FaultFree*)
1 – uszkodzony (*Faulty*)
2 – nie testuj (*NoTest*)

Uruchomiony węzeł 0 co określony czas t_1 testuje trzy pozostałe węzły (tj takie których pozycjom w tablicy tests odpowiada wartość inna niż 2 – *NoTest*);
Ponieważ żaden inny węzeł nie jest włączony, węzeł 0 odbiera to jako uszkodzenie uciszające wszystkich pozostałych węzłów, i ponieważ jest to zgodne z jego zapisem w tablicy tests nie inicjuje on stanu aktywnego algorytmu.

2. Uruchamiany węzeł 1 wykonuje procedurę inicjalizacji węzła. Jego wewnętrzne struktury danych mają po tej operacji postać:

Nodeid	1			
Nodenum	4			
Event	-1	-1	-1	-1
Diagnosis	1	0	1	1
Tests	1	2	1	1

Wartości w tablicach diagnosis i tests oznaczają:
0 – sprawny (*FaultFree*)
1 – uszkodzony (*Faulty*)
2 – nie testuj (*NoTest*)

Załóżmy, że czas t_1 odstępu pomiędzy kolejnymi testami okresowymi jest dużo większy od czasu przesyłania i interpretacji pakietów oraz że test przeprowadzony przez węzeł 0 na zainicjowanym węźle 1 został wykonany wcześniej niż test węzła 1 na węźle 0. Wynik testu przeprowadzonego przez węzeł 0 na węźle 1 dał wynik *FaultFree* co jest niezgodne z zapisem w tablicy tests węzła 0 i w związku z tym zwiększa on znacznik czasowy w swojej tablicy event, aktualizuje tablicę tests i inicjuje fazę aktywną algorytmu wysyłając pakiet adapt2 o następującej treści:

Pakiet1

Sendid	0			
Recvid	1			
Rootevent	0			
Event	0	-1	-1	-1
From	0	-1	-1	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Algorytm przeszukiwania drzewa „najpierw w głąb” (DFS) wyznacza na odbiorcę pakietu jedyne go sprawnego sąsiada – węzeł 1. Struktury danych węzłów po zinterpretowaniu pakietu przez węzeł 1 wyglądają następująco:

Nodeid	0			
Nodenum	4			
Event	0	-1	-1	-1
Diagnosis	0	1	1	1
Tests	2	0	1	1

Nodeid	1			
Nodenum	4			
Event	0	-1	-1	-1
Diagnosis	1	0	1	1
tests	1	2	1	1

Algorytm DFS wykonywany w węźle 1 nie znajduje żadnego innego sprawnego sąsiada więc odsyła go do rodzica – węzła 0. Odesłany pakiet ma postać:

Pakiet2

Sendid	1			
Recvid	0			
Rootevent	0			
Event	0	-1	-1	-1
From	0	0	-1	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Widać że węzeł 1 zaznaczył jedynie w tablicy from pakietu skąd otrzymał pakiet.

Węzeł 0 stwierdza, że wszystkie sprawne węzły były już odwiedzane, że on sam jest węzłem korzeniem, że pakiet po raz pierwszy powrócił do korzenia, że nowy przydział testów nie jest jeszcze ustawiony w węźle. Po stwierdzeniu tych faktów kopiuje tablicę from do topologię pakietu, kasuje tablicę from, przypisuje sobie prawo do testowania wykrytego sąsiada oraz dotąd uszkodzonych węzłów i wysyła pakiet do sąsiada. Jego dane po tej operacji wyglądają następująco:

Nodeid	0			
Nodenum	4			
Event	0	-1	-1	-1
Diagnosis	0	0	1	1
Tests	2	0	1	1

Postać wysłanego pakietu jest następująca:

Pakiet3

Sendid	0			
Recvid	1			
Rootevent	0			
Event	0	-1	-1	-1
From	0	-1	-1	-1
Topology	0	0	-1	-1
Istested	0	0	1	1

Zawartość tablicy topology informuje o proponowanym przydziale testów, tablica istested informuje że węzły uznane za uszkodzone (tj 2 i 3) są przez „kogoś” testowane.

Węzeł 1 po zinterpretowaniu pakietu ustawi swoje tablice tests i diagnosis. Jego dane będą wyglądały następująco:

Nodeid	1			
Nodenum	4			
Event	0	-1	-1	-1
Diagnosis	0	0	1	1
Tests	0	2	2	2

Z danych wynika, że węzeł jest „świadomy” sprawności dwu węzłów w systemie (dwa zera na zerowej i pierwszej pozycji tablicy diagnosis) oraz tego, że węzeł ma za zadanie testować okresowo węzeł 0 (jedynek na zerowej pozycji tablicy tests).

Faza aktywna kończy się odesłaniem pakietu następującej postaci do korzenia – węzła 0.

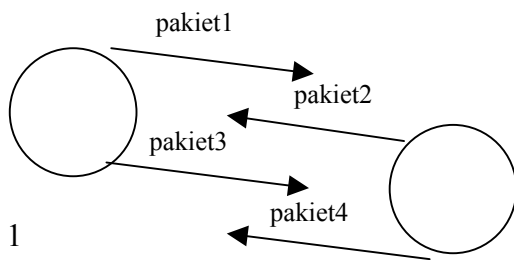
Pakiet ma postać:

Pakiet4

Sendid	1			
Recvid	0			
Rootevent	0			
Event	0	-1	-1	-1
From	0	0	-1	-1
Topology	0	0	-1	-1
Istested	0	0	1	1

Korzeń pamięta że jest to już trzecia wizyta pakietu z tym samym znacznikiem czasu, więc kończy fazę aktywną algorytmu (nie wysyła już dalej pakietu).

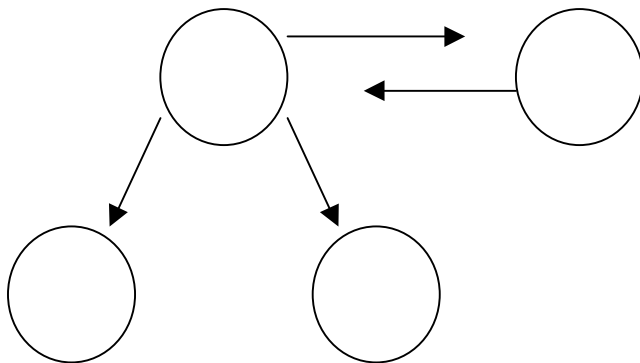
Przesłania realizowane podczas opisywanych wyżej operacji przedstawia rysunek 1:



Rys 1

System wraca do fazy pasywnej, w której węzeł 0 testuje okresowo węzły: 1, 2 i 3 (tests[2011]), a węzeł 1 testuje okresowo węzeł 0 (tests[0222]).

Przydział testów (struktura diagnostyczna systemu) po zakończeniu fazy aktywnej przedstawia rysunek2:



Rys 2

Ciąg zdarzeń będzie taki jak przedstawiony powyżej jeżeli węzeł 1 nie rozpocznie testowania węzła 0 przed zinterpretowaniem pakietu3.

3. Do systemu jest dołączany węzeł 2. Po wykonaniu procedury inicjującej dane węzła wyglądają następująco:

Nodeid	2			
Nodenum	4			
Event	-1	-1	-1	-1
Diagnosis	1	1	0	1
Tests	1	1	2	1

Wartości w tablicach diagnosis i tests oznaczają:

0 – zdalny (FaultFree)

1 – **niezdalny (Faulty)**

2 – nie testuj (NoTest)

Węzeł 0 odpowiedzialny za jego testowanie zauważy jego obecność przy najbliższym testowaniu okresowym. Fakt wykrycia sprawnego sąsiada węzeł 0 zaznaczy w swojej tablicy tests, zwiększy swój znacznik czasu w tablicy event i zainicjuje fazę aktywną algorytmu wysyłając do węzła 1 pakiet testowy.

Dane węzła 0 po wysłaniu pakietu i zawartość pakietu są następujące:

Nodeid	0			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	1	1
Tests	2	0	0	1

Pakiet5

Sendid	0			
Recvid	1			
Rootevent	1			
Event	1	-1	-1	-1
From	0	-1	-1	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Dane węzła 1 po odebraniu pakietu i zawartość pakietu odesłanego do węzła 0 będą miały postać:

Nodeid	1			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	1	1
Tests	0	2	2	2

Węzeł 0 po odebraniu tego pakietu nie zmienia swoich danych transmituje jedynie pakiet do nowo wykrytego węzła 2. Pakiet ma postać:

Pakiet7

Sendid	0			
Recvid	2			
Rootevent	1			
Event	1	-1	-1	-1
From	0	0	-1	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Węzeł 2 modyfikuje swoje dane do postaci:

Nodeid	2			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	1	1	0	1
Tests	1	1	2	1

Po czym odsyła do węzła 0 pakiet postaci:

Pakiet8

Sendid	2			
Recvid	0			
Rootevent	1			
Event	1	-1	-1	-1
From	0	0	0	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Węzeł 0 stwierdza, że wszystkie sprawne węzły były już odwiedzane, że on sam jest węzłem korzeniem, że nowy przydział testów nie jest jeszcze ustawiony w węzle. Po stwierdzeniu tych faktów kopiuje tablicę from do topologii pakietu, kasuje tablicę from, przypisuje sobie prawo do testowania wykrytych sąsiadów oraz dotąd uszkodzonego węzła 3 i wysyła pakiet do sąsiada 1. Jego dane po tej operacji wyglądają następująco:

Nodeid	0			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	0	1
Tests	2	0	0	1

Wysłany pakiet ma postać:

Pakiet9

Sendid	0			
Recvid	1			
Rootevent	1			
Event	1	-1	-1	-1
From	0	-1	-1	-1
Topology	0	0	0	-1
Istested	0	0	0	1

Dane węzła 1 po zinterpretowaniu tego pakietu i pakiet przesłany dalej są postaci:

Nodeid	1			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	0	1
Tests	0	2	2	2

Pakiet10

Sendid	1			
Recvid	0			
Rootevent	1			
Event	1	-1	-1	-1
From	0	0	-1	-1
Topology	0	0	0	-1
Istested	0	0	0	1

Węzeł 0 nie zmienia swoich danych, wysyła natomiast pakiet do węzła 2 postaci:

Pakiet11

Sendid	0			
Recvid	2			
Rootevent	1			
Event	1	-1	-1	-1
From	0	0	-1	-1
Topology	0	0	0	-1
Istested	0	0	0	1

Węzeł 2 po odebraniu tego pakietu zmienia swoje dane do postaci:

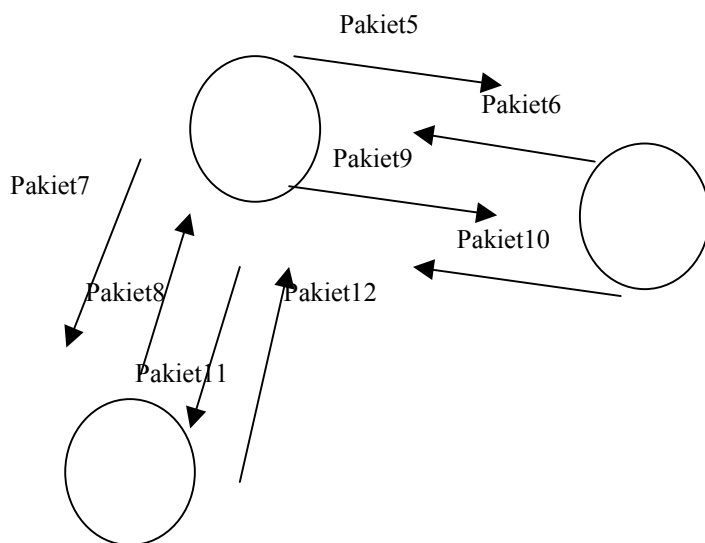
Nodeid	2			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	0	1
Tests	0	2	2	2

Następnie odsyła do węzła 0 pakiet postaci:

Pakiet12

Sendid	2			
Recvid	0			
Rootevent	1			
Event	1	-1	-1	-1
From	0	0	-1	-1
Topology	0	0	0	-1
Istested	0	0	0	1

Węzeł 0 – korzeń nie wysyła już dalej pakietu, kończąc w ten sposób fazę aktywną algorytmu. Trasę przebytą przez pakiet w ostatniej fazie aktywnej przedstawia rysunek 3:



Rys 3

Wartości danych wewnętrznych poszczególnych węzłów przed i podczas procesu rekonfiguracji oraz treści przesyłanych pakietów przedstawiono poniżej.

Dane węzłów po zainicjowaniu węzła 3, i przed zauważeniem tego faktu przez węzeł 0:

Nodeid	0			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	0	1
Tests	2	0	0	1

Nodeid	1			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	0	1
Tests	0	2	2	2

Nodeid	2			
Nodenum	4			
Event	1	-1	-1	-1
Diagnosis	0	0	0	1
Tests	0	2	2	2

Nodeid	3			
Nodenum	4			
Event	-1	-1	-1	-1
Diagnosis	1	1	1	0
Tests	1	1	1	2

Po stwierdzeniu obecności węzła 3 węzeł 0 inicjuje fazę aktywną.

Nodeid	0			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	1
Tests	2	0	0	0

Topology	-1	-1	-1	-1
Istested	-	-	-	-

Pakiet13

Sendid	0			
Recvid	1			
Rootevent	2			
Event	2	-1	-1	-1
From	0	-1	-1	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Pakiet15

Sendid	0			
Recvid	2			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	-1	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Nodeid	1			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	1
Tests	0	2	2	2

Nodeid	2			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	1
Tests	0	2	2	2

Pakiet14

Sendid	1			
Recvid	0			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	-1	-1

Pakiet16

Sendid	2			
Recvid	0			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	0	-1

Topology	-1	-1	-1	-1
Istested	-	-	-	-
Pakiet17				
Sendid	0			
Recvid	3			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	0	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Rootevent	2			
Event	2	-1	-1	-1
From	0	0	-1	-1
Topology	0	0	0	0
Istested	0	0	0	0

Nodeid	3			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	1	1	1	0
Tests	1	1	1	2

Pakiet21

Sendid	0			
Recvid	2			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	-1	-1
Topology	0	0	0	0
Istested	0	0	0	0

Pakiet18				
Sendid	3			
Recvid	0			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	0	0
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Nodeid	2			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

Nodeid	0			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	2	0	0	0

Pakiet22

Sendid	2			
Recvid	0			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	0	-1
Topology	0	0	0	0
Istested	0	0	0	0

Pakiet19				
Sendid	0			
Recvid	1			
Rootevent	2			
Event	2	-1	-1	-1
From	0	-1	-1	-1
Topology	0	0	0	0
Istested	0	0	0	0

Pakiet23

Sendid	0			
Recvid	3			
Rootevent	2			
Event	2	-1	-1	-1
From	0	0	0	-1
Topology	0	0	0	0
Istested	0	0	0	0

Nodeid	1			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

Pakiet20				
Sendid	1			
Recvid	0			

Nodeid	3			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

Pakiet24

Sendid	3			
Recvid	0			
Rootevent	2			
Event	2	-1	-1	-1

From	0	0	0	0
Topology	0	0	0	0
Istested	0	0	0	0

4. Załóżmy, że uszkodzeniu ulega węzeł 0 i że jako pierwszy dostrzega to węzeł 2 oraz że udaje się mu przeprowadzić fazę aktywną algorytmu zanim uszkodzenie zauważyły węzły 1 i 3.

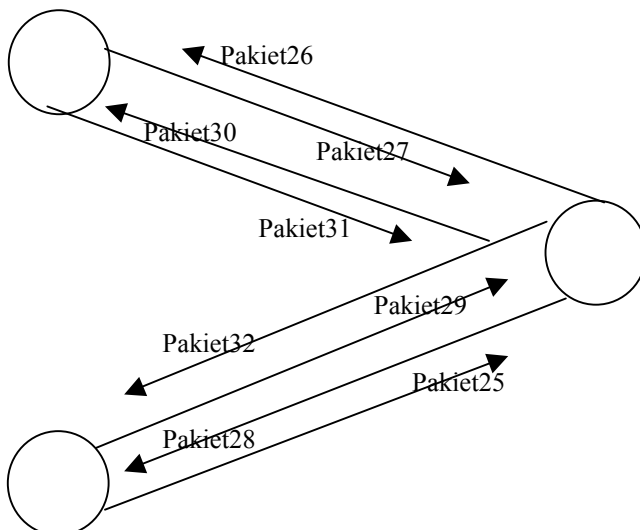
Wartości dany trzech sprawnych węzłów w momencie uszkodzenia węzła 0 są następujące:

Nodeid	1			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

Nodeid	2			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

Nodeid	3			
Nodenum	4			
Event	2	-1	-1	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

W takiej sytuacji pakiety przesyłane będą w kolejności przedstawionej na rysunku:



Rys 7

Zmiany w danych węzłów i pakietów przedstawiono poniżej:

Nodeid	2			
Nodenum	4			
Event	2	-1	0	-1
Diagnosis	0	0	0	0
Tests	1	0	2	2

Pakiet25

Sendid	2			
Recvid	1			
Rootevent	0			
Event	2	-1	0	-1
From	-1	-1	2	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Nodeid	1			
Nodenum	4			
Event	2	-1	0	-1
Diagnosis	0	0	0	0
Tests	0	2	2	0

Istested	-	-	-	-
----------	---	---	---	---

Pakiet26

Sendid	1			
Recvid	3			
Rootevent	0			
Event	2	-1	0	-1
From	-1	2	2	-1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Nodeid	2			
Nodenum	4			
Event	2	-1	0	-1
Diagnosis	1	0	0	0
Tests	1	0	2	2

Nodeid	3			
Nodenum	4			
Event	2	-1	0	-1
Diagnosis	0	0	0	0
Tests	0	2	2	2

Pakiet29

Sendid	2			
Recvid	1			
Rootevent	0			
Event	2	-1	0	-1
From	-1	-1	2	-1
Topology	-1	2	2	1
Istested	1	0	0	0

Pakiet27

Sendid	3			
Recvid	1			
Rootevent	0			
Event	2	-1	0	-1
From	-1	2	2	1
Topology	-1	-1	-1	-1
Istested	-	-	-	-

Nodeid	1			
Nodenum	4			
Event	2	-1	0	-1
Diagnosis	1	0	0	0
Tests	2	2	0	0

Pakiet28

Sendid	1			
Recvid	2			
Rootevent	0			
Event	2	-1	0	-1
From	-1	2	2	1
Topology	-1	-1	-1	-1

Pakiet30

Sendid	1			
Recvid	3			
Rootevent	0			
Event	2	-1	0	-1

From	-1	2	2	-1
Topology	-1	2	2	1
Istested	1	0	0	0

Nodeid	3			
Nodenum	4			
Event	2	-1	0	-1
Diagnosis	1	0	0	0
Tests	2	0	2	2

Pakiet31

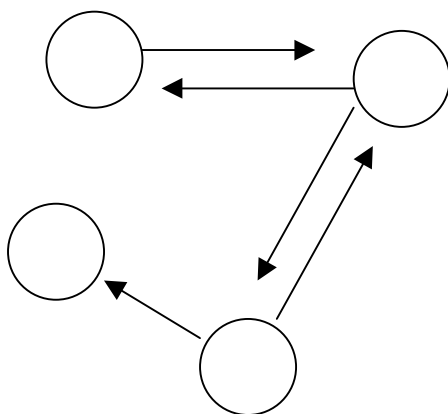
Sendid	3
Recvid	1
Rootevent	0

Event	2	-1	0	-1
From	-1	2	2	1
Topology	-1	2	2	1
Istested	1	0	0	0

Pakiet32

Sendid	1			
Recvid	2			
Rootevent	0			
Event	2	-1	0	-1
From	-1	2	2	1
Topology	-1	2	2	1
Istested	1	0	0	0

Po zakończeniu fazy aktywnej przydział testów jest taki jak przedstawiony na rysunku 8.



Rys 8

Gdyby węzły 1 lub 3 zauważyły uszkodzenie węzła 0 przed rozesłaniem do nich kompletnego pakietu z nowymi przydziałami rozpoczęłyby również fazę aktywną z założeniami że odpowiednio każdy z nich jest korzeniem drzewa. Najbardziej prawdopodobny ostateczny przydział testów byłby przydziałem wyznaczonym przez węzeł korzeń o największym numerze nodeid. Wynika to z zastosowanego mechanizmu rozstrzygania tego typu konfliktów, tj mechanizmu „dogrywki” (tie break). Nowy przydział konstruowany w tej samej fazie aktywnej jest nadpisywany w systemie na już istniejący, jeżeli węzeł-korzeń nowego przydziału ma numer większy od węzła-korzenia poprzedniego przydziału.