

Tematyka

Na ćwiczeniach przeprowadzimy symulację z wykorzystaniem routingu dynamicznego. Dokonamy analizy plików z danymi symulacji.

Większe topologie, routing dynamiczny

Zacniemy od zdefiniowania topologii. Tym razem zrobimy to inaczej niż na poprzednich ćwiczeniach. Węzły zapiszemy w tablicy, uprości nam to tworzenie bardziej złożonych scenariuszy. Poniższy fragment skryptu tworzy siedem węzłów i zapisuje je w tablicy `n()`.

```
for {set i 0} {$i < 7} {incr i} {  
  set n($i) [$ns node]  
}
```

Następnie łączymy węzły w pierścień.

```
for {set i 0} {$i < 7} {incr i} {  
  $ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail  
}
```

Ćwiczenie 1

Uruchom symulację. Sprawdź topologię.

Poniższy skrypt przesyła dane z węzła `n(0)` do węzła `n(3)`.

```
#Tworzymy agenta UDP i dołączamy go do węzła n0  
set udp0 [new Agent/UDP]  
$ns attach-agent $n(0) $udp0  
  
#Tworzymy źródło danych typu CBR i dołączamy je do udp0  
set cbr0 [new Application/Traffic/CBR]  
$cbr0 set packetSize_ 500  
$cbr0 set interval_ 0.005  
$cbr0 attach-agent $udp0  
  
set null0 [new Agent/Null]  
$ns attach-agent $n(3) $null0  
  
$ns connect $udp0 $null0  
  
$ns at 0.5 "$cbr0 start"  
$ns at 4.5 "$cbr0 stop"
```

Ćwiczenie 2

Uruchom symulację. Zaobserwuj przepływ danych.

Struktura plików z danymi symulacji (trace files)

Aby dokonać dokładnej analizy, potrzebujemy pliku z danymi symulacji (obecnie korzystaliśmy tylko z plików z danymi do wizualizacji programem nam).

```
#otwieramy trace file
set tf [open out.tr w]
$ns trace-all $tf
```

Musimy również zmodyfikować procedurę finish.

```
proc finish {} {
    global ns nf tf
    $ns flush-trace
    close $nf
    close $tf
    exec nam out.nam &
    exit 0
}
```

Poniżej znajduje się przykład pliku z danymi symulacji.

```
+ 0.5 0 1 cbr 500 ----- 0 0.0 3.0 0 0
- 0.5 0 1 cbr 500 ----- 0 0.0 3.0 0 0
+ 0.505 0 1 cbr 500 ----- 0 0.0 3.0 1 1
- 0.505 0 1 cbr 500 ----- 0 0.0 3.0 1 1
+ 0.51 0 1 cbr 500 ----- 0 0.0 3.0 2 2
- 0.51 0 1 cbr 500 ----- 0 0.0 3.0 2 2
r 0.514 0 1 cbr 500 ----- 0 0.0 3.0 0 0
+ 0.514 1 2 cbr 500 ----- 0 0.0 3.0 0 0
- 0.514 1 2 cbr 500 ----- 0 0.0 3.0 0 0
```

Plik składa się z dwunastu kolumn:

1. pole opisujące typ zdarzenia. r, +, -, d oznaczają odpowiednio: odbiór, przyjęcie do kolejki, opuszczenie kolejki, odrzucenie pakietu
2. czas zdarzenia
3. węzeł źródłowy
4. węzeł docelowy
5. rodzaj pakietu
6. wielkość pakietu
7. flagi
8. identyfikator przepływu
9. adres źródłowy (w postaci „węzeł.port”)
10. adres docelowy
11. numer sekwencyjny pakietu

12. unikalny numer pakietu

Mając plik z danymi i znając jego format możemy w łatwy sposób policzyć np. ilość informacji wysłanych/odebranych przez dany węzeł. Poniższy skrypt oblicza ilość danych wysłanych przez węzeł n0.

```
#!/usr/bin/perl -w

$bytes = 0;

while($tline = <>)
{
    chomp $tline;

    @tr = split(" ", $tline);

    #interesują nas tylko zdarzenia opuszczenia kolejki
    if($tr[0] eq '-') {

        #interesuje nas tylko węzeł 0
        if($tr[2] eq '0') {

            #kolumna 5 zawiera wielkość pakietu
            $bytes += $tr[5];

        }

    }

}

print "$bytes\n"
```

Ćwiczenie 3

Sprawdź działanie powyższego skryptu.

Awarie łącza

Zgodnie z oczekiwaniami dane w naszych symulacjach były przesłane najkrótszą ścieżką (przez węzły 1 i 2). Wprowadźmy awarię łącza. Połączenie pomiędzy węzłami 1 i 2 (które jest aktualnie używane do przesyłania danych) wyłączymy na jedną sekundę.

```
$ns rtmodel-at 1.0 down $n(1) $n(2)
$ns rtmodel-at 2.0 up $n(1) $n(2)
```

Zrozumienie powyższych dwóch linii programu nie powinno sprawiać trudności. Po uruchomieniu symulacji można zauważyć, że dane przesłane w pierwszej i drugiej sekundzie są stracone.

Ćwiczenie 4

Uruchom symulację. Zaobserwuj przepływ danych. Sprawdź ile danych wysłał węzeł n0 a ile z nich dotarło do węzła n3.

Sieci Komputerowe 2 / Ćwiczenia 3

Następnie uruchomimy protokół routingu dynamicznego (w tym przypadku będzie to protokół typu wektor odległości). Poniższą linię dodajemy po utworzeniu obiektu symulatora.

```
$ns rtproto DV
```

Ćwiczenie 5

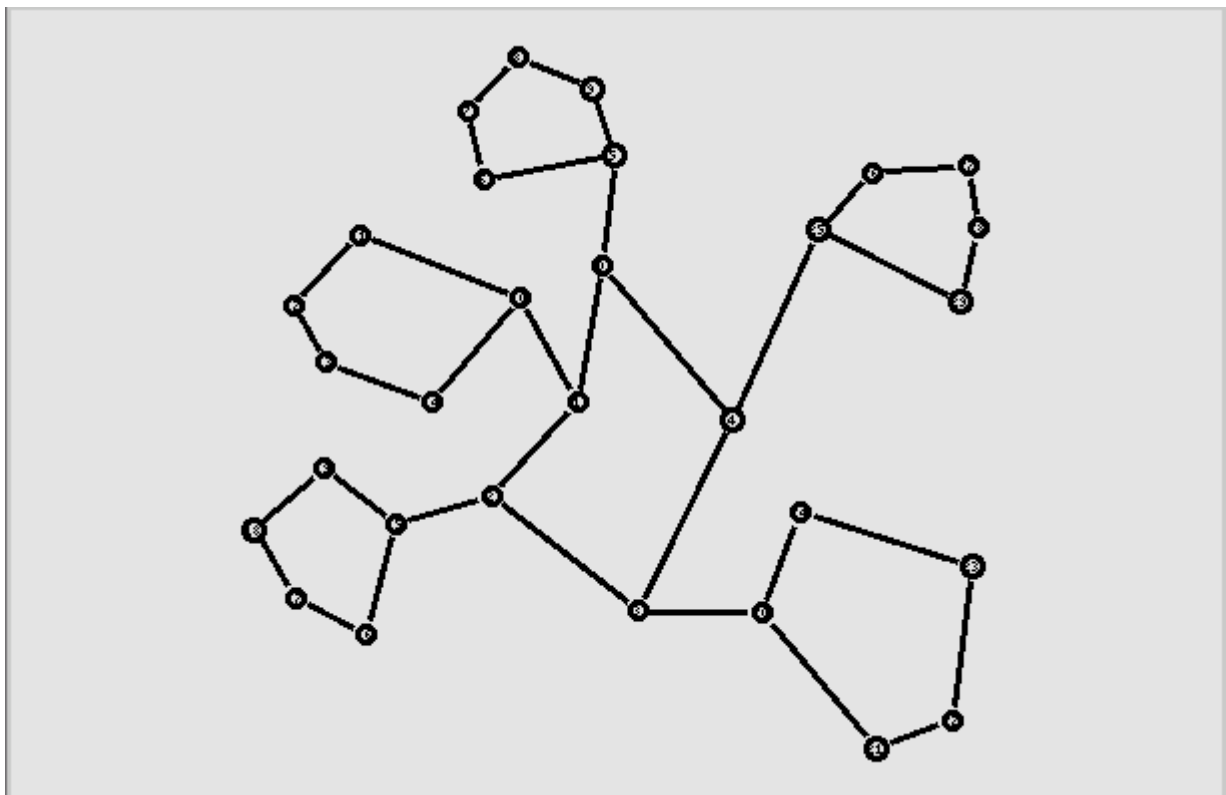
Uruchom symulację. Zaobserwuj przepływy danych. Sprawdź ile danych zostało straconych.

Po uruchomieniu symulacji możemy zauważyć wiele małych pakietów wysyłanych pomiędzy węzłami. Jeżeli zwolnimy symulację na tyle, aby udało się kliknąć na pakiet zobaczymy, że jest to pakiet typu 'rtProtoDV' odpowiedzialny za wymianę informacji między węzłami. Kiedy w pierwszej sekundzie łącze przestaje działać, informacje o dostępnych ścieżkach zostaną zaktualizowane i ostatecznie ruch zostanie przekierowany przez węzły 6, 5, 4.

Ćwiczenie 6

Zbuduj topologię jak na ilustracji 1. Dodaj trzy strumienie danych (dodaj kolory, aby strumienie danych były łatwe do rozróżnienia). Wprowadź awarie łącz. Zaobserwuj zmiany tras. Czas symulacji ustaw na 2 godziny. Sprawdź ile danych wygenerował protokół routingu typu wektor odległości. Sprawdź ile danych zostało straconych podczas transmisji. Zapisz wyniki.

Wykorzystaj protokół routingu typu stan łącza (\$ns rtproto LS). Przeprowadź te same badania. Sformułuj wnioski.



Ilustracja 1: Topologia sieci (ekran programu nam)