Opracował: Konrad Kawecki <cogi@pjwstk.edu.pl>

Tematyka

Symulacje z wykorzystaniem protokołu TCP. Zdarzenia losowe.

Źródła danych: TCP

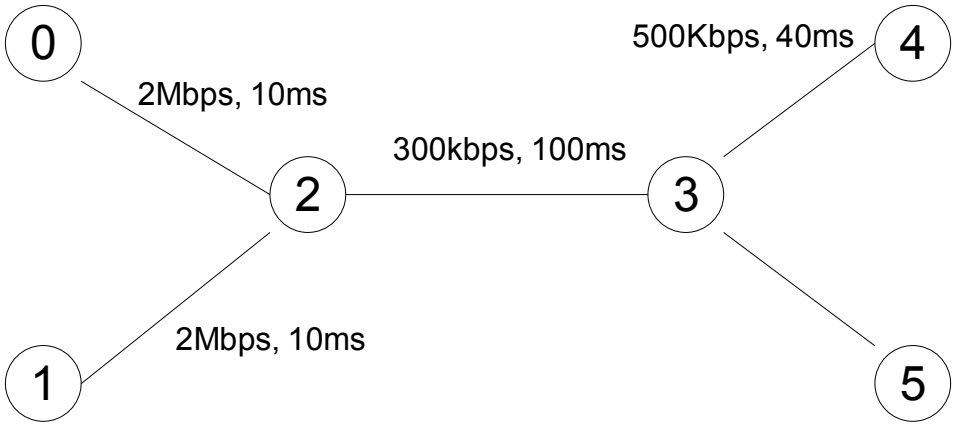
Do tej pory korzystaliśmy ze źródła danych CBR. Poniższy fragment skryptu pokazuje, jak możemy zdefiniować agenta TCP oraz wykorzystać aplikację FTP.

```
#definicja agenta TCP
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp


set tcp_sink [new Agent/TCPSink]
$ns attach-agent $n1 $tcp_sink

$ns connect $tcp $tcp_sink

#ustawienie FTP
set ftp [new Application/FTP]
$ftp attach-agent $tcp

$ns at 0.1 "$ftp start"
$ns at 1.0 "$ftp stop"
```

Ćwiczenie 1

Zbuduj topologię jak na ilustracji 1. Zdefiniuj źródło danych UDP oraz TCP. Zasymuluj awarię łącza. Sprawdź ile danych nie dotarło do odbiorcy (sprawdź dane zarówno dla UDP i TCP). Sformułuj wnioski. Czy wykorzystanie dynamicznego rutingu ma wpływ na straty danych w protokole TCP? Oblicz średnią przepustowość połączenia TCP.

W ćwiczeniu wykorzystaj poznane wcześniej metody na określenie położenia węzłów.



Ilustracja 1: Schemat sieci

Potwierdzenia (ACK), straty pakietów

W przypadku TCP potwierdzenia (ACK) informują nadawcę o numerze sekwencyjnym kolejnego, oczekiwanego przed odbiorcę pakietu. Pakiet uważa się za stracony jeżeli:

- nadawca odbierze kolejno trzy ACK dotyczące tego samego pakietu
- jeżeli przekroczony zostanie czas oczekiwania (przykładowe obliczenia przedstawione są na ćwiczeniach pierwszych)

Aby zmniejszyć liczbę ACK generowanych przez system, TCP używa tzw. „opóźnionych ACK" tzn. ACK jest wysyłane co *n* odebranych pakietów. Standardową wielkością *n* jest 2 (RFC 1122). Należy zwrócić uwagę, że w przypadku kiedy okno jest ustawione na 1 opóźnione ACK mogą powodować zakleszczenia. Jeżeli po odebraniu pierwszego pakietu (z *n* oczekiwanych) pozostałe nie dotrą w ustalonym czasie (standardowo 100ms), ACK jest wysyłane bez oczekiwania na pozostałe pakiety.

Protokół TCP: okno (sliding window)

Informacje dotyczące TCP oraz kontroli przeciążeń, były dokładnie omówione na wykładzie Sieci Komputerowe 1.

Protokół TCP kontroluje prędkość transmisji. Wykorzystuje do tego parametr nazywany oknem (*W*). Jeżeli ilość wysłanych pakietów osiągnie wartość *W, n*adawca musi wstrzymać transmisję i poczekać na potwierdzenie (nadawca wznowi transmisję, jeżeli czas oczekiwania przekroczy ustalony limit). Aby potwierdzenie i retransmisja zagubionych pakietów były możliwe, każdy pakiet jest oznaczony numerem sekwencyjnym. W zależności od warunków panujących w sieci, okno ma zmienną wielkość.

Poniżej przedstawiona jest procedura gromadząca informacje o wielkości okna.

```
proc tcpWindow {tcpSource file} {
global ns
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now + $time] "tcpWindow $tcpSource $file"
}

$ns at 0.1 "tcpWindow $tcp $winfile"
```

Gdzie:

- `$tcp` jest to agent tcp z którego chcemy gromadzić informacje
- `$winfile` jest to uchwyt do pliku w którym będziemy zapisywać wielkość okna

Ćwiczenie 2

Wykorzystaj topologię z ćwiczenia pierwszego. W symulacji zostaw tylko i wyłacznie jedno źródło danych TCP. Sprawdź działanie procedury `tcpWindow`. Zmiany okna przedstaw na wykresie.

Ćwiczenie 3

Przedstaw (na wykresie) przepustowość połączenia TCP względem czasu.

Zdarzenia losowe

Do naszych symulacji wprowadzimy zdarzenia losowe. Chcemy, aby nasze połączenie pomiędzy węzłami 2 i 3 losowo odrzucało pakiety. Zadanie to realizuje poniższy skrypt.

```
set loss_module [new ErrorModel]
$loss_module set rate_ 0.2
$loss_module ranvar [new RandomVariable/Uniform]
$loss_module drop-target [new Agent/Null]
$ns lossmodel $loss_module $n2 $n3
```

Powyższy fragment powoduje, że około 20% pakietów przesyłanych przez łącze n2-n3 będzie stracone.

Ćwiczenie 4

Sprawdź w działaniu powyższy skrypt. Do poprzednich ćwiczeń wprowadź losowe straty pakietów. Przedstaw przepustowość oraz wielkość okna TCP na wykresach. Sformułuj wnioski.