

Symulator sieci komputerowych NS-2

Główna strona projektu: <http://www.isi.edu/nsnam/ns>
http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/SIK_G_zadania
(^ materiały zebrane przez MH)

NS-2 to symulator (nie emulator) sieci komputerowych...
pracuje się z nim "wsadowo", a wygląda to tak:

1. przygotowuje się skrypt (np. ns01.tcl), który:
 - + definiuje **topologie sieci** (węzły, połączenia, kolejki komunikatów)
 - + definiuje **ruch pakietów w sieci** poprzez określenie agentów, aplikacji oraz początkowych zdarzeń;
agent definiuje protokół, np. udp, tcp
aplikacja to np. ftp, cbr (constant bit rate)
2. uruchamiamy symulację programem ns: `./ns ns01.tcl`
 - + program ns przeprowadza symulację i tworzy log/trace ze zdarzeniami w pliku ns01.tr oraz ns01.nam (animacje)
3. oglądamy log ns01.tr,
przetwarzamy go różnymi narzędziami np. programem awk, gnuplot,
oglądamy animację działania sieci przy pomocy `./nam ns01.nam`

Sposób pracy z NS-2:

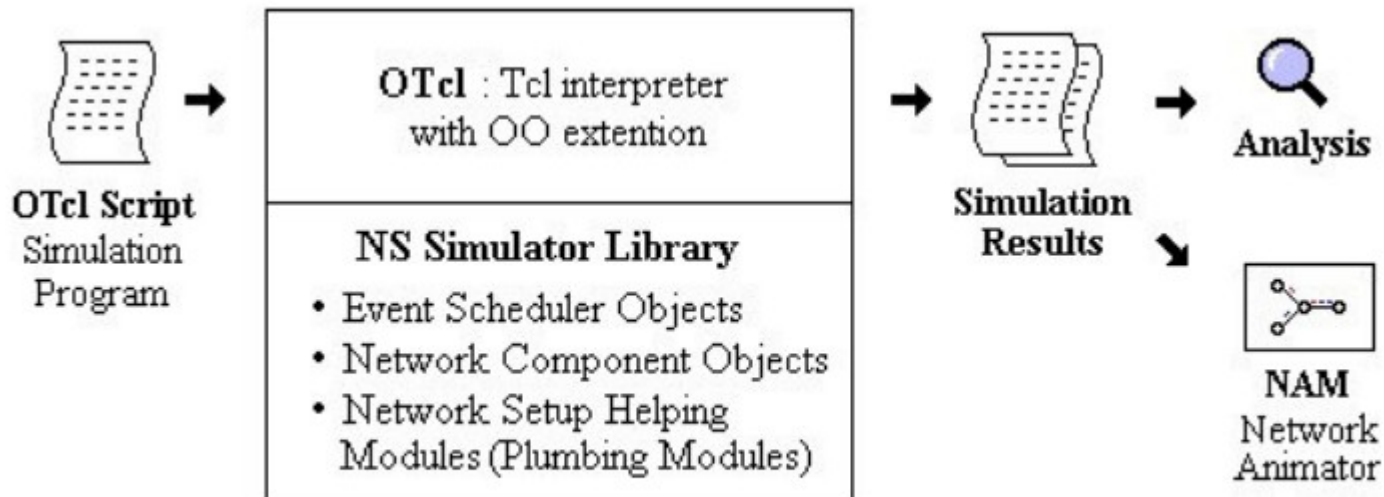


Figure 3. Simplified User's View of NS-2 [9]

Główna książka o NS-2: <http://www.isi.edu/nsnam/ns/doc/index.html>

(^ głównie potrzebna do rozszerzania symulatora...)

<http://www.isi.edu/nsnam/ns/tutorial/index.html> - tutorial 1

<http://nile.wpi.edu/NS/> - tutorial 2

E. Altman, T. Jimenez "NS Simulator for beginners" (dobre!!)

T. Issariyakul, E. Hossain, "Introduction to Network Simulator NS2" (obszerna!!)

Instalacja NS-2: pod lin32, skopiować .zip z folderu:

<https://mhanckow.students.wmi.amu.edu.pl/sik/ns-2/>

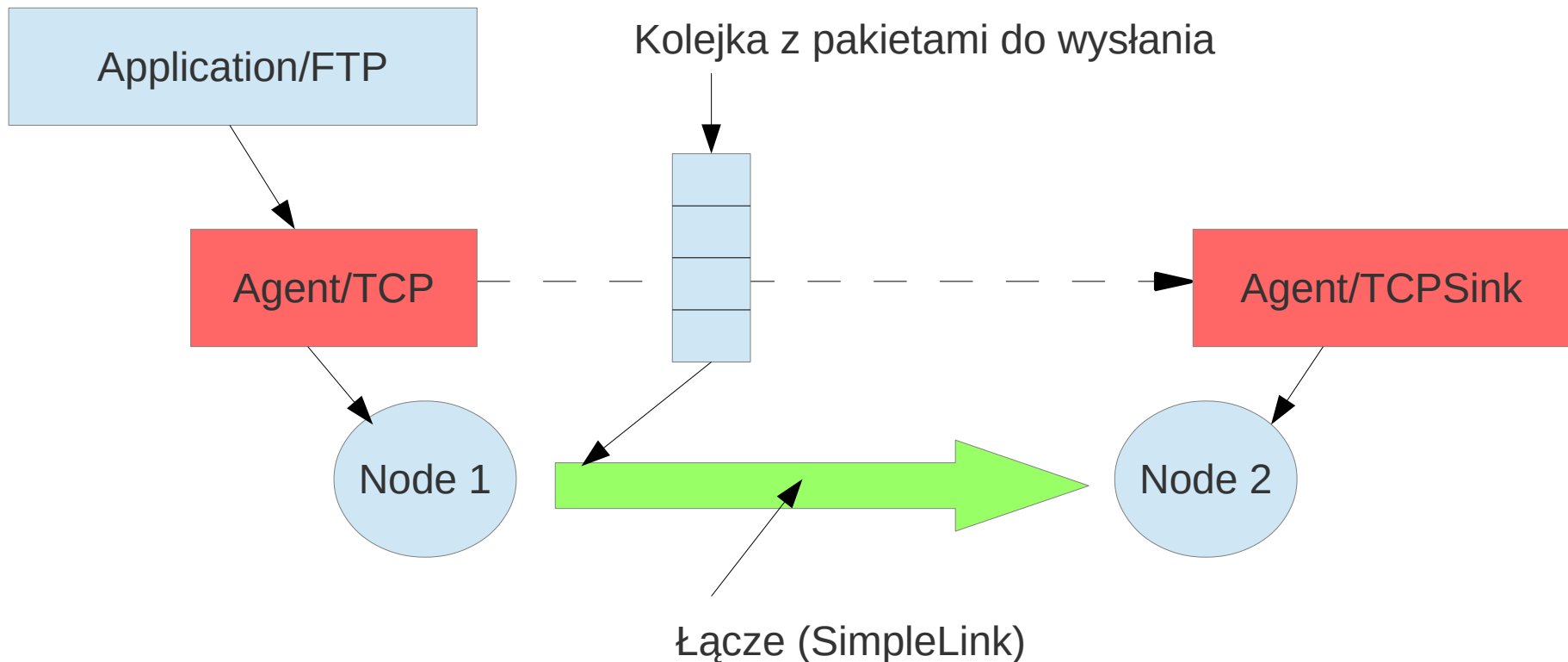
Pokażemy jak się definiuje sieć oraz ruch sieciowy przy pomocy obiektów:

Simulator, node, SimpleLink, Agent/UDP, Agent/TCP*, Application/*, ...

Najprostszy kompletny przykład (ale z UDP i CBR):

http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/SIK_G_przyk01

Zasada łączenia aplikacji, agenta i węzła (node:)



Tworzenie połączenia (2x SimpleLink):

Podaje się: 1) przepustowość, 2) opóźnienie, 3) typ kolejki (qdisc !!)

```
set ns [new Simulator]
```

```
$ns color 1 red
```

```
$ns color 2 blue
```

```
$ns namtrace-all [open ns07.nam w]
```

```
$ns trace-all [open ns07.tr w]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
set n3 [$ns node]
```

```
$ns duplex-link $n1 $n2 30Mb 50ms DropTail
```

```
$ns duplex-link $n2 $n3 20Mb 50ms DropTail
```

```
# kolejka DropTail (najprostsza)
```

```
# met duplex-link tworzy 2 ob. SimpleLink, w obie strony...
```

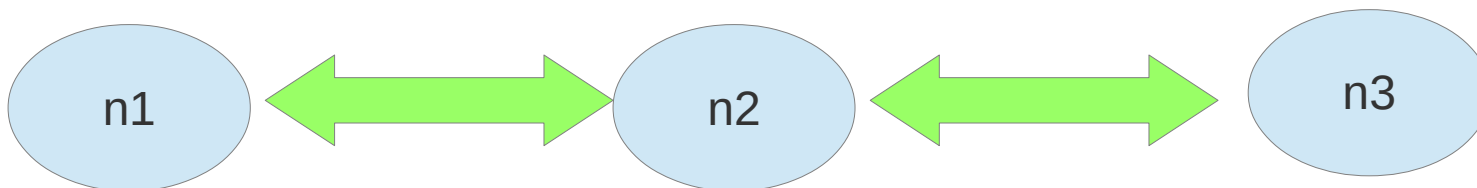
Podczepianie Agentów (prot 4 warstwy) oraz Aplikacji (FTP, CBR)

```
set tcp [new Agent/TCP]  
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $n1 $tcp  
$ns attach-agent $n3 $sink  
# podczepiamy agentów pod węzły
```

```
$ns connect $tcp $sink  
# tworzymy połączenie
```

```
set ftp [new Application/FTP]  
$ftp attach-agent $tcp  
# instalujemy aplikację FTP nad TCP
```



Definiowanie początkowych zdarzeń:

```
$ns at 0.1 {$ftp start}
```

```
$ns at 7.9 {$ftp stop}
```

```
$ns at 8.0 {$ns halt}
```

```
# początkowe zdarzenia; następne wynikają z symulacji...
```

```
$ns run
```

```
$ns flush-trace
```

```
puts "...koniec"
```

```
# uruchomienie symulacji + napisy końcowe
```

Okazuje się, że NS-2 jest „sterowany zdarzeniami” a nie zegarem...

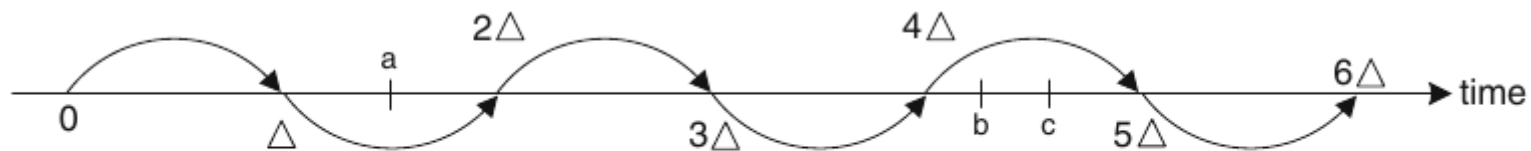


Fig. 1.3 Clock advancement in a time-driven simulation

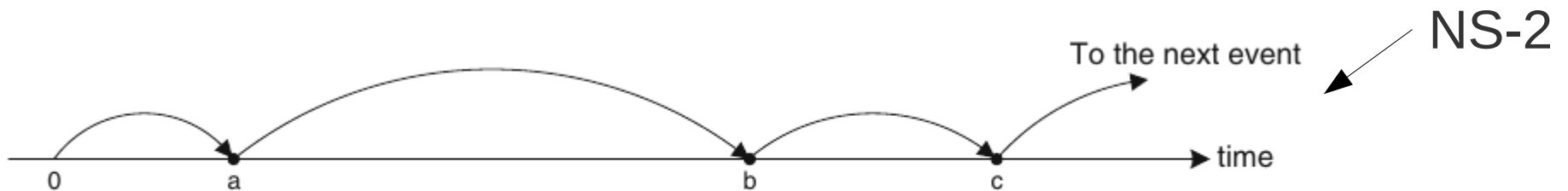


Fig. 1.4 Clock advancement in an event-driven simulation

Ogólne uwagi o NS-2:

1. skrypty definiujące sieć i ruch w sieci programuje się w j. OTcl (rozszerzenie OOP Tcl);
programy ns, nstk, nse to po prostu interpretery Tcl z dodanymi klasami NS-2
2. NS-2 jest rozszerzalny - można dodawać nowe protokoły, robi się to w j. OTcl i C++;
techn. "*split-objects*", każdy obiekt ma 2 połówki: w OTcl i w C++;
dlaczego używa się 2 języków? odp: ???
3. NS-2 zawiera bardzo wiele protokołów różnych warstw (np. kilka(naście) odmian TCP)
4. j. OTcl (tak jak Tcl) posiada introspekcję, dlatego czasami warto uruchomić konsolę: `./nstk konsola2c.tcl`
możemy wtedy interaktywnie "zajrzeć" do obiektów, wyświetlić ich metody, zmienne...
możemy wyświetlić klasy, np. "info comm Agent/Tcp/*" wyświetli wszystkie wersje prot. TCP w NS-2 !
jednak symulacje wygodniej prowadzić wsadowo, czyli bez konsoli...

Format pliku log/trace:

Każde zdarzenie w osobnej linii w poniższym formacie,

Typy zdarzeń:

1. (+) wstawia się pakiet do kolejki połączenia
2. (-) wyciąga się pakiet z tej kolejki, i jest on przesyłany fizycznie przez połączenie
3. (r) pakiet wychodzi drugim końcem połączenia
4. (d) jest też możliwe że pakiet został porzucony, np. z powodu przepełnienia kolejki

Event	Time	From node	To nodetype	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id
-------	------	-----------	-------------	----------	-------	-----	----------	----------	---------	--------

Figure 2.5: Fields appearing in a trace

przykład pliku trace (.tr)

```
+ 0.1 0 1 tcp 40 ----- 0 0.0 2.0 0 0
- 0.1 0 1 tcp 40 ----- 0 0.0 2.0 0 0
r 0.150107 0 1 tcp 40 ----- 0 0.0 2.0 0 0
+ 0.150107 1 2 tcp 40 ----- 0 0.0 2.0 0 0
- 0.150107 1 2 tcp 40 ----- 0 0.0 2.0 0 0
r 0.200213 1 2 tcp 40 ----- 0 0.0 2.0 0 0
+ 0.200213 2 1 ack 40 ----- 0 2.0 0.0 0 1
```