Communication Services and Security
TCP Congestion Lab

Cèsar Fernández

Departament d'Informàtica
Universitat de Lleida

2019 - 2020

Objectives

- To understand the network simulator ns-2
- To work with the TCP congestion and flow control mechanisms

ns: concepts

- Discrete event simulator. 2 languages:
  - `C++`. Core
  - `Tcl`. User interface
- Large object library: Nodes, Agents (TCP, UDP, . . .), Traffic generators, . . .
- ns-2 no longer mantained
- Current project: ns-3

ns: installation

- allinone (2.35) easiest way. Or here (closer)
- Download, copy and untar to destination dir (`$HOME`, `/usr/local/ , ...`)
- Packages required: `libX11-devel tcl tk gcc-c++ libXt-devel`
- Run `install`
- If error: follow instructions
- Set up environment variables `LD_LIBRARY_PATH` i `TCL_LIBRARY`

```
IMPORTANT NOTICES:

(1) You MUST put /tmp/ns-allinone-2.35/otcl-1.14, /tmp/ns-allinone-2.35/lib,
    into your LD_LIBRARY_PATH environment variable.
    If it complains about X libraries, add path to your X libraries
    into LD_LIBRARY_PATH.
    If you are using csh, you can set it like:
        setenv LD_LIBRARY_PATH <paths>
    If you are using sh, you can set it like:
        export LD_LIBRARY_PATH=<paths>

(2) You MUST put /tmp/ns-allinone-2.35/tcl8.5.10/library into your
    TCL_LIBRARY environmental variable.
    Otherwise ns/nam will complain during startup.
```

ns: installation

- Version 2.35. Compilation errors. Fix them manually (Already fixed at local package)

```
nkstate/ls.h:137:20: note: declarations in dependent base ‘std::map<i
std::allocator<std::pair<const int, LsIdSeq> > >’ are not found by un
linkstate/ls.h:137:20: note: use ‘this->erase’ instead
make: *** [linkstate/ls.o] Error 1
Ns make failed!
```

- For Fedora $\geq$ 24 (gcc 6.X, 7.X), if error in `mdart`:
  - Change to dir `ns-2.35`
  - Include `-std=gnu++98` in file `Makefile` (CCOPT of the compiler)
  - Compile (`make`)
  - Go back to root dir
  - Edit `install` file, and comment files 536-538
  - Run again `install`
- Or easier:
  - Download gcc53 gcc53-c++-5.3.0-1.el6.x86_64.rpm
  - Run: `rpm -i ./gcc53-c++-5.3.0-1.el6.x86_64.rpm`
  - Export env. vars: `export CC=gcc53 CXX=g++53`
  - Run again `install`

ns: scripts

- TCL language
- Run

```
ns script.tcl parameters
```

ns: topology definition

- Slide examples (sim1.tcl ns code)

```
#Create  4 nodes
#
#  n0
#  \
#   \
#    n2--------n3
#   /
#  /
# n1

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Link the nodes with duplex comms links
$ns duplex-link $n0 $n2 5Mb 20ms DropTail
$ns duplex-link $n1 $n2 5Mb 20ms DropTail
$ns duplex-link $n2 $n3 1Mb 50ms DropTail
```

ns: topology definition

- Slide examples (sim1.tcl ns code)

```
# Node 0; UDP agent with  CBR traffic
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set rate_ 0.5Mbps
$cbr0 attach-agent $udp0
$udp0 set class_ 0   # flow identifier

# Node 1:  TCP agent using  Karn  algorithm
# Change  tcpTick  timer default value
# With  CBR traffic generator
set tcp0 [new Agent/TCP/RFC793edu]
$tcp0 set class_ 1   # flow identifier
$tcp0 set add793karnrtt_ $karn
$tcp0 set add793jacobsonrtt_ $jacobson
$tcp0 set add793expbackoff_ true
$tcp0 set add793slowstart_ true
$ns attach-agent $n1 $tcp0
$tcp0 set tcpTick_ 0.01
```

ns: topology definition

- Slide examples (sim1.tcl ns code)

```
set cbr1 [new Application/Traffic/CBR]
$cbr1 set rate_ 0.5Mbps
$cbr1 attach-agent $tcp0

# Node 3: 2 Sinks
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
set null1 [new Agent/TCPSink]
$ns attach-agent $n3 $null1

# Connect  agents
$ns connect $udp0 $null0
$ns connect $tcp0 $null1
```

ns: events planning

- Slide examples (sim1.tcl ns code)

```
$ns at 5.0 "$cbr0 start"
$ns at 10.0 "$cbr0 stop"

$ns at 0.0 "$cbr1 start"
$ns at 0.0 "record"
$ns at 15.0 "finish"

proc record { } {
    ....

    $ns at [expr $now+0.1] "record"
}
```

ns: tracing results

- Slide examples (sim1.tcl ns code)

```
set nf [open $arxiu.tr w]
$ns trace-all $nf
set nff [open $arxiu.rtt w]

proc record { } {
    ....
    set now [$ns now]
    puts $nff "$now $rtt $srtt $rto "
    ...
}
```

ns: internal variables

- Internal `ns` variables end with `_` (`cwnd_ rtt_ ...`)
- Some of them are:
  - `cwnd_`: `cwnd`
  - `ssthresh_`: `cwmax`
  - `window_`: `CWMAX`
- One can access through the TCP agent:

```
puts "Value of cwnd:  [$tcp0 set cwnd_]"
```

- Slide examples (sim1.tcl ns code)

```
set rtt [expr [$tcp0 set rtt_] * [$tcp0 set tcpTick_]]

set srtt  [expr ([$tcp0 set srtt_]
   >> [$tcp0 set T_SRTT_BITS])*[$tcp0 set tcpTick_]]

set rttvar  [expr ([$tcp0 set rttvar_]
   >> [$tcp0 set T_RTTVAR_BITS]) * [$tcp0 set tcpTick_]]

set rto [expr [$tcp0 set rto_] * [$tcp0 set tcpTick_]]
...
puts $nff "$now $rtt $srtt $rto "
```

- Usually:
  - `T_SRTT_BITS` = 3
  - `T_RTTVAR_BITS` = 2

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Event types:**
  - `+/-` put in and drop from queue
  - `r` received (at the end of the link)
  - `d` dropped

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Time** when the event occurs (seconds)

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Node** source and destination

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Segment type**

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Segment size:** in bytes

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Flags:** *Explicit Congestion Indicator*, *Priority*, ...

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Flow Identifier** (`class_`)

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Addresses** source and destination (`node:port`)

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Sequence number:** also for UDP

ns: trace format

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

d 5.541136 2 3 cbr 210  ------- 0 0.0 3.0 155 1368
```

- **Segment identifier.** Unique

ns: timing example

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

r 0.290752 2 1 ack 40   ------- 1 3.1 1.0 1   4
```

- Generate and transmit a segment from node 1
- Receiving at node 2

$$0.1623 - 0.1407 = 0.0216 = \frac{1000\,\text{bytes} \cdot 8\,\text{bits/byte}}{5 \cdot 10^6\,\text{Mbps}} + 20\,\text{ms}$$

- Generate ACK from node 3. Sequence number: 1
- ACK reception at node 1. $RTT = 0.2907552 - 0.140768 = 150\,\text{ms}$

ns: timing example

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

r 0.290752 2 1 ack 40   ------- 1 3.1 1.0 1   4
```

- Generate and transmit a segment from node 1
- Receiving at node 2

$$0.1623 - 0.1407 = 0.0216 = \frac{1000\,\text{bytes} \cdot 8\,\text{bits/byte}}{5 \cdot 10^6\,\text{Mbps}} + 20\,\text{ms}$$

- Generate ACK from node 3. Sequence number: 1
- ACK reception at node 1. $RTT = 0.2907552 - 0.140768 = 150\,\text{ms}$

ns: timing example

```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

r 0.290752 2 1 ack 40   ------- 1 3.1 1.0 1   4
```

- Generate and transmit a segment from node 1
- Receiving at node 2

$$0.1623 - 0.1407 = 0.0216 = \frac{1000\,\text{bytes} \cdot 8\,\text{bits/byte}}{5 \cdot 10^6\,\text{Mbps}} + 20\,\text{ms}$$

- Generate ACK from node 3. Sequence number: 1
- ACK reception at node 1. $RTT = 0.2907552 - 0.140768 = 150\,\text{ms}$

ns: timing example

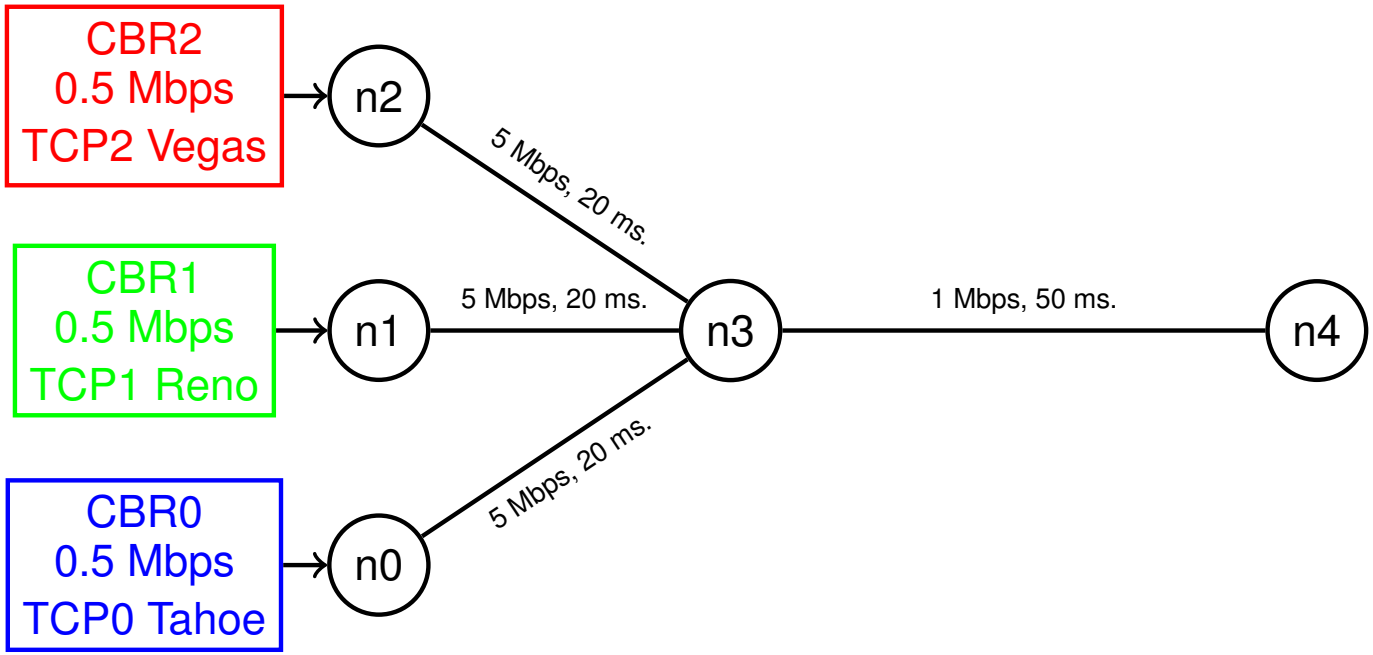
```
+ 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2
- 0.140768 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

r 0.162368 1 2 tcp 1000 ------- 1 1.0 3.1 1   2

+ 0.220368 3 2 ack 40   ------- 1 3.1 1.0 1   4

r 0.290752 2 1 ack 40   ------- 1 3.1 1.0 1   4
```

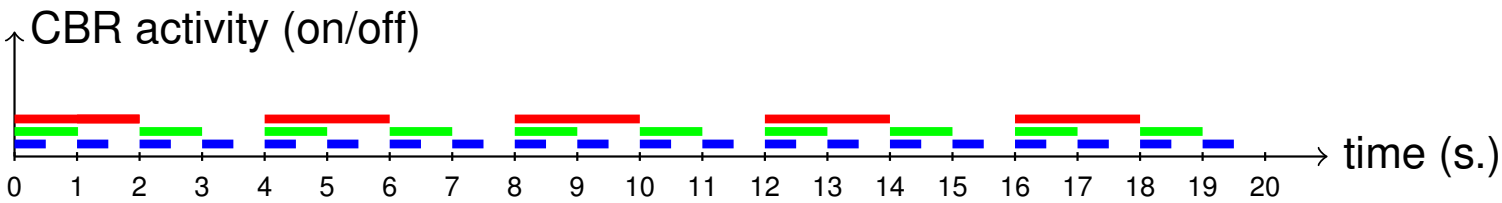
- Generate and transmit a segment from node 1
- Receiving at node 2

$$0.1623 - 0.1407 = 0.0216 = \frac{1000\,\text{bytes} \cdot 8\,\text{bits/byte}}{5 \cdot 10^6\,\text{Mbps}} + 20\,\text{ms}$$

- Generate ACK from node 3. Sequence number: 1
- ACK reception at node 1. $RTT = 0.2907552 - 0.140768 = 150\,\text{ms}$

Simulation scenario





Lab work requirements

To write a ns-2 script for the previous simulation scenario, and for each TCP agent:

1. Compute the number of packet losses and bytes transferred
2. Compute the congestion window

Set `tcpTick` to 0.01 s. and `cwmax` to 40. Define as queue limit size for $n3$ (i.e. 20)

To deliver

A `tgz` file containing:

- Scripts code (simulation, processing, plot, . . .)
- A PDF file explaining scripts execution, and showing results:
  - Packet losses and transferred bytes table
  - Congestion windows plots

  as well as some conclusions

Bibliography

- The network simulator ns-2. ns-2 website
- ns manual
- ns for beginners