

# EEC 687/787 Mobile Computing

## Ns-2 Laboratory #9

Prof. Chansu Yu

<http://academic.csuohio.edu/yuc/>  
c.yu91@csuohio.edu

## How to make changes in ns-2?

- Source file changes
- Variable binding
  - Variables can be set to different values in tcl script
  - Variables can be traced or monitored
- Command method

# Variable Binding

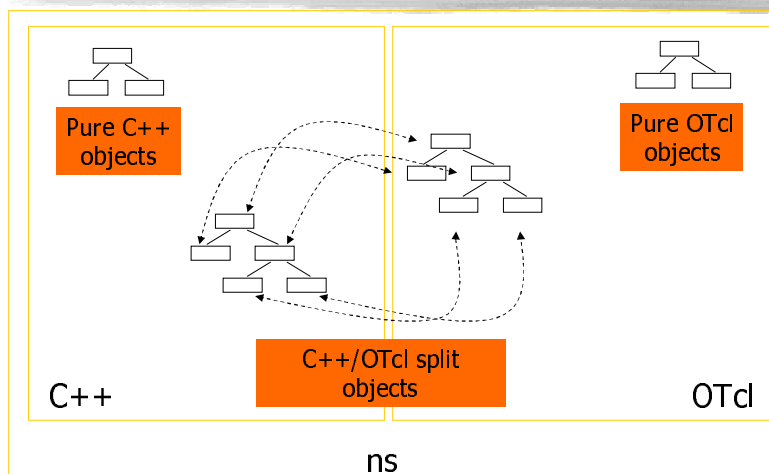
- ❑ Set/change a variable of C++ component from TCL run script
  - Advantage: No need to modify the source file (no need to compile & link)
  
- ❑ Special function “bind”
  - Interface: bind, bind\_bw, bind\_time, bind\_bool

```
WirelessPhy::WirelessPhy() : Phy(), sleep_timer_(this), status_(IDLE) {  
    bind("CPTthresh_", &CPTthresh_);  
    bind("CSTthresh_", &CSTthresh_);  
    bind("RXThresh_", &RXThresh_);  
    bind("Pt_", &Pt_);  
}
```
- Default value is defined in \$ns/tcl/lib/ns-default.tcl
  - Mac/802\_11 set dataRate\_ 1.0e6
  - Mac/802\_11 set PLCPDataRate\_ 1.0e6
  - Phy/WirelessPhy set CPTthresh\_ 10.0
  - Phy/WirelessPhy set RXThresh\_ 3.652e-10
  - Phy/WirelessPhy set bandwidth\_ 2e6
  - Phy/WirelessPhy set Pt\_ 0.28183815

3

c.yu91@csuohio.edu

# OTcl and C++



4

c.yu91@csuohio.edu

## NS2 Design: C++ and Otcl Separation

- ❑ C++ for data
  - Protocol implementation, such as TCP.cc, UDP.cc
  - per packet action
- ❑ Otcl for control
  - Configuration and control in simulation
  - periodic or triggered action
- + Compromise between composibility and speed
- Learning & debugging

5

*c.yu91@csuohio.edu*

## Tracing using Bound Variables

- ❑ Periodic probing in OTCL
  - Periodically record the value of traced variable
  - Self-calling of OTCL procedure
  - Variable should be visible in OTCL (bound variable)
- ❑ Variable tracing support
  - Automatically record whenever the value of traced variable changes
  - Variable must be visible in OTCL
  - Variable must belong to trace class

6

*c.yu91@csuohio.edu*

## In-Class Lab #1

- Download tcp\_simple.tcl, fil2tcp.awk, fil3tcp.awk
  - Number of nodes (fixed)
  - Topology and link capacity
  - Traffic
  
- What is expected result? Try!
  
- Which variables do you want to trace? What are the variable names for them? Check with ~/tcp/tcp.{cc, h}
  - Packet sequence number ??
  - Congestion window size ??

7

*c.yu91@csuohio.edu*

## In-Class Lab #2

- Periodic probing of cwnd\_ & chart
- Variable tracing of cwnd\_ & chart
- How to compare them?
  
- Try another variable !
  - Refer Ch. 35 of ns-2 manual for candidates

8

*c.yu91@csuohio.edu*

## Tracing using Bound Variables

- ❑ Example: “cwnd” (congestion window size) in TCP protocol
  - Let us try the **periodic probing**
  - Variable should be bound: “~tcp/tcp.cc”

```
proc probe {} {  
    global ns tcp1  
    set now [$ns now]  
    set cwnd [$tcp1 set cwnd_  
    puts "$now $cwnd"  
  
    $ns at [expr $now+1] "probe"  
}  
  
$ns at 1.001 "probe"
```

*Bound variable  
in tcp.cc*

*Self-calling*

*Initial start-up*

- ❑ How to draw a chart for “cwnd”? 9

```
ns tcp_simple.tcl > cwnd1  
xgraph cwnd1
```

*c.yu91@csuohio.edu*

## Tracing using Bound Variables

- ❑ More detailed information on “cwnd\_” changes
  - Let us try the **variable tracing**
    - Variable should be bound: “~tcp/tcp.cc”
    - Variable must be belong to trace class: “~tcp/tcp.h”

```
set tracer_ [new Trace/Var]  
$tracer_ attach [open cwndtrace.tr w]  
$tcp1 trace cwnd_ $tracer_
```

10

*c.yu91@csuohio.edu*

# Tracing using Bound Variables

% more cwndtrace.tr

```
f t0 a_o50 ncwnd_ v1
f t100.004248 a_o50 ncwnd_ v2
f t100.027196 a_o50 ncwnd_ v2.5
f t100.039368 a_o50 ncwnd_ v2.9

f t100.041344 a_o50 ncwnd_ v3.24483
f t100.073669 a_o50 ncwnd_ v3.55301
f t100.085981 a_o50 ncwnd_ v3.83446
f t100.098013 a_o50 ncwnd_ v4.09525
f t100.099949 a_o50 ncwnd_ v4.33944
f t100.102025 a_o50 ncwnd_ v4.56988
f t100.115863 a_o50 ncwnd_ v4.78871
f t100.129382 a_o50 ncwnd_ v4.99753
```

f: trace type  
t: time  
a: name of trace owner  
n: name of traced variable  
v: value

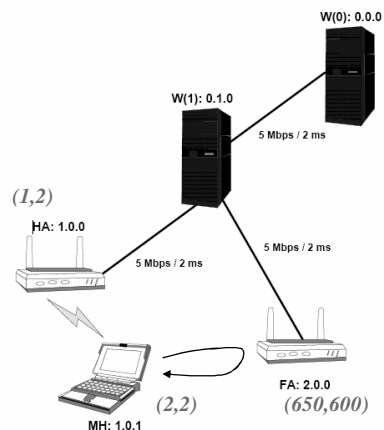
```
cat cwndtrace.tr | awk '{print
substr($2, 2, length($2)-1),
substr($5, 2, length($5)-1)}'
> cwnd2
```

*graph cwnd2*

c.yu91@csuohio.edu

# Lab Report: Mobile IP (revisited)

- 5 nodes
  - 2 wired nodes, W(0) and W(1)
  - 2 mobile agents, HA and FA
  - A mobile host, MH
- Traffic: W(0) => MH
  - W(0) → W(1) → HA → MH
- Mobility
  - MH moves toward FA
  - Then, moves back to HA
  - Routing path changes to
    - W(0) → W(1) → FA → MH and then
    - W(0) → W(1) → HA → MH again



12

c.yu91@csuohio.edu

# Lab Report

---

- Download infra.tcl, fil-tcp.awk
  
- Periodic probing of cwnd\_
- Variable tracing of cwnd\_
- Try another variable !
  
- Discussion:
  - What is the expected dynamics of congestion window?