EEC 687 Mobile Computing (Spring, 2007)

Ns-2 Laboratory

Chansu Yu

Cleveland State University

Contents

- Lab exercise 1: TCP in fixed network
- Lab exercise 2: TCP in Mobile IP network
- Homework #4
- Trace and monitoring support to obtain other outputs than standard trace output provides
  - Tracing using Bound Variables
  - Tracing via Command Method

c.yu91@csuohio.edu

Lab Exercise #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 ??

3

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 be belong to trace class

4

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”?

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]
$tcp trace cwnd_ $tracer_
```


c.yu91@csuohio.edu

Tracing using Bound Variables

% more cwndtrace.tr

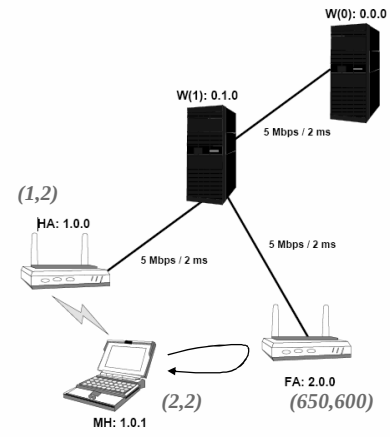
```
f t0 a_o50 ncwnd_ v1
f t100.004248 a_o50 ncwnd_ v2
f t100.027196 a_o50 ncwnd_ v2.5
f t100.039368 a_o50 ncwnd_ v2.9
f t100.041344 a_o50 ncwnd_ v3.24483
f t100.073669 a_o50 ncwnd_ v3.55301
f t100.085981 a_o50 ncwnd_ v3.83446
f t100.098013 a_o50 ncwnd_ v4.09525
f t100.099949 a_o50 ncwnd_ v4.33944
f 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

c.yu91@csuohio.edu

Lab Exercise #2: 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



c.yu91@csuohio.edu

Lab Exercise #2

- Download infra.tcl, fil-tcp.awk
- What is the expected dynamics of congestion window?
- Try by yourself.

9

c.yu91@csuohio.edu

Homework #4: TCP layer (due Apr. 23 M)

- Goal: TCP performance is critically limited in wireless networks simply because TCP has been designed for wired networks.

  The most important fact is that TCP regards packet loss as congestion and takes some necessary actions mostly targeted to generate less traffic. However, in wireless networks, packet loss is often due to node mobility and the network is able to function normally as soon as a new route is established because there was no congestion from the beginning.

  One recent paper showed that TCP performance can be greatly improved in wireless networks by reducing the congestion window. The goal of this homework is to investigate the TCP performance with different congestion windows size via ns-2 simulation.

10

c.yu91@csuohio.edu

Homework #4: TCP layer (due Apr. 23 M)

- Output: Simulation report including excel charts with explanation and discussion. Show throughput and congestion window size over time (average for all or for just one node??).
- Bonus1: It will be interesting to see the TCP performance versus hop count because TCP performance depends on the hop count between two end nodes.
- Bonus2: It is also interesting to reveal the instantaneous TCP performance since it is known that TCP performance fluctuates over time in a significant way.

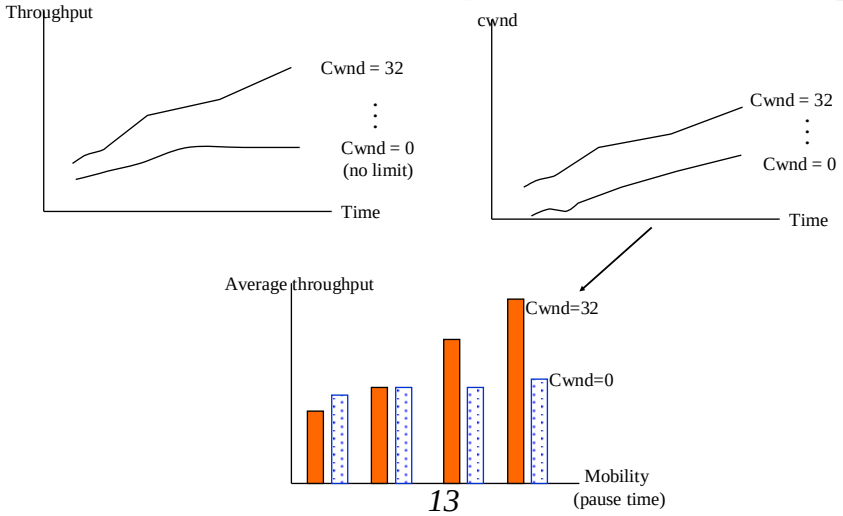
c.yu91@csuohio.edu

Scenario (fifty.tcl)

- Standard scenario
  - 50 nodes, 1500x300m network area, simulation time of 900 seconds
  - TwoRayGround, 802.11, AODV
  - 20 CBR sources, ten 256-byte packets/second
  - Maximum node speed 5m/s, pause time of 0~900 seconds
- Mobility generator: setdest
- Traffic generator: cbrgen (it is used to generate TCP traffic as well)

c.yu91@csuohio.edu

Homework #4: TCP layer



c.yu91@csuohio.edu