

ĐO ĐẠC CÁC THAM SỐ MẠNG TRONG NETWORK SIMULATOR 2

I. MỤC ĐÍCH THÍ NGHIỆM

Trong bài thí nghiệm số 8, sinh viên đã được làm quen với công cụ mô phỏng mạng NS-2 với việc tạo đồ hình mạng đơn giản. Bài thí nghiệm này sẽ giúp sinh viên học được cách sử dụng NS-2 để đo đạc các tham số của mạng mô phỏng.

II. THẢO LUẬN

2.1. Trace-all

Trong bài 8, khi viết chương trình mô phỏng bằng file .tcl, sinh viên đã được làm quen với file trace và lệnh trace-all để theo dõi hoạt động toàn bộ các gói tin. Bây giờ chúng ta sẽ tìm hiểu kỹ hơn về file trace .tr này.

Đối với đồ hình mạng có dây mà ta tiến hành mô phỏng, file trace thu được sẽ là các string gồm 12 thông tin:

1	2	3	4	5	6	7	8	9	10	11	12
event	time	From node	To node	Packet type	Packet size	flags	Flow ID	Source address	Destination address	Sequence number	Packet ID

1. Event: sự kiện

+ : gói tin vào hàng đợi

- : gói tin ra khỏi hàng đợi

r : gói tin đến đích

d : gói bị dropped

c : gói bị xung đột ở lớp MAC

2. Time: thời điểm

3. From node: nút truyền
4. To node: nút nhận
5. Packet Type: Loại gói (tcp/ cbr/ ...)
6. Packet Size: Kích thước gói (bytes)
7. Flags: cờ
8. Flow ID: Mã số luồng
9. Source Address: Nút gốc
10. Destination Address: Nút đích
11. Sequence Number: Số thứ tự gói
12. Packet ID: Mã số gói

Dựa vào các thông số này của file trace, ta có thể tiến hành viết các thuật toán để tính toán các thông số như lưu lượng, trễ lan truyền, tỉ lệ mất gói, ... của mạng mô phỏng.

2.2. File .awk

File .awk được dùng để viết các thuật toán để giúp ta tính toán các thông số của mạng khi mô phỏng bằng NS-2.

Ta sẽ sử dụng file .awk để in các thông số cần thiết ra thành 1 file .tr mới gồm các string có thông số thời gian và thông số cần thiết.

2.3. Xgraph

Xgraph là 1 ứng dụng vẽ đồ thị được tích hợp sẵn khi cài đặt NS-2.

Sau khi sử dụng các file .awk để có được các file .tr mới. Ta sẽ dùng xgraph để vẽ các đồ thị theo thời gian của các thông số cần thiết.

III. YÊU CẦU VỀ THIẾT BỊ

- Máy tính chạy hệ điều hành Ubuntu 14.04
- Cài đặt phần mềm Network Simulator 2 (NS-2)

IV. NỘI DUNG THÍ NGHIỆM

- Tìm hiểu về công cụ cho phép phân tích dữ liệu trong đo đạc, mô phỏng
- Tạo một đồ hình mạng bất kỳ.
- Tạo các nguồn phát lưu lượng.
- Đo lưu lượng, trễ, tỷ lệ mất gói:
 - + Đo lưu lượng theo luồng, đo lưu lượng trên từng liên kết
 - + Đo trễ trên từng chặng (1 luồng), trễ từ đầu cuối tới đầu cuối
 - + Đo tỷ lệ mất gói theo luồng, tỷ lệ mất gói trên từng liên kết

V. TRÌNH TỰ THÍ NGHIỆM

5.1. Code

5.1.1. Tạo folder lưu trữ

Trong folder **[MSSV]** đã tạo ngoài desktop ở bài 8, tạo folder mới, đặt tên là **bai9**.

5.1.2 Tạo file .tcl và thiết lập đồ hình mạng

Tạo file **bai9.tcl** trong folder **bai9** vừa tạo và tạo mạng như sau:

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

```
$ns color 1 Red
```

```
$ns color 2 Green
```

```
set pksize 512
```

```
-      Khai báo các file trace sẽ được tạo ra do .awk
```

```
+      Bandwidth
```

```
set tcpbw [open tcpbw.tr w]
```

```
set udpbw [open udpbw.tr w]
```

```
+      Delay
```

```
set tcpdl [open tcpdl.tr w]
```

```
set udpdl [open udpdl.tr w]
```

```
+      Lost rate
```

```
set tcplr [open tcplr.tr w]
```

```
set udplr [open udplr.tr w]
```

```
set tf [open bai9.tr w]
```

```
$ns trace-all $tf
```

```
set nf [open bai9.nam w]
```

```
$ns namtrace-all $nf
```

```
proc finish {} {
```

```
global ns tf nf tcpbw udpbw tcpdl udpdl tcplr udplr
```

```
$ns flush-trace
```

```
close $udpbw
```

```
close $tcpbw
```

close \$tcpdl

close \$udpdl

close \$tcpplr

close \$udpplr

close \$nf

close \$tf

exec nam bai9.nam &

- Sử dụng các file .awk để xử lý số liệu của file bai9.tr và tiến hành vẽ đồ thị xgraph:

exec awk -f [tên file .awk] [tên file .tr cần xử lý]

exec xgraph [tên file .tr cần vẽ] -bg [màu nền] -fg [màu chi tiết] -geometry [kích thước đồ thị] -t [tên đồ thị] -x [tên trục hoành] -y [tên trục tung] &

exec awk -f bandwidth.awk bai9.tr

exec xgraph tcpbw.tr udpbw.tr -bg "black" -fg "white" -geometry 1000x500 -t "Bandwidth" -x "s" -y "Mbps" &

exec awk -f lostrate.awk bai9.tr

exec xgraph tcplr.tr udplr.tr -bg "black" -fg "white" -geometry 1000x500 -t "Lost Rate" -x "s" -y "%" &

exec awk -f delay.awk bai9.tr

exec xgraph tcpdl.tr udpdl.tr -bg "black" -fg "white" -geometry 1000x500 -t "Delay" -x "s" -y "ms" &

exit 0

}

set s0 [\$ns node]

\$s0 label "TCPSource"

set s1 [\$ns node]

\$s1 label "UDPSource"

set d2 [\$ns node]

\$d2 label "UDPNull"

set d3 [\$ns node]

\$d3 label "TCPSink"

\$s0 shape "hexagon"

\$s0 color "red"

\$s1 shape "hexagon"

\$s1 color "green"

\$d2 shape "square"

\$d2 color "green"

\$d3 shape "square"

\$d3 color "red"

\$ns duplex-link \$s0 \$s1 4Mb 10ms DropTail

\$ns duplex-link \$s1 \$d2 6Mb 10ms DropTail

\$ns duplex-link \$d2 \$d3 4Mb 10ms DropTail

\$ns duplex-link-op \$s0 \$s1 orient right

\$ns duplex-link-op \$s1 \$d2 orient right

\$ns duplex-link-op \$d2 \$d3 orient right

\$ns duplex-link-op \$s0 \$s1 queuePos 0.5

\$ns duplex-link-op \$s1 \$d2 queuePos 0.5

\$ns duplex-link-op \$d2 \$d3 queuePos 0.5

\$ns queue-limit \$s0 \$s1 20

\$ns queue-limit \$s1 \$d2 50

\$ns queue-limit \$d2 \$d3 20

- Tạo nguồn lưu lượng

+ Tạo nguồn TCP

• Tạo nguồn

set tcp [new Agent/TCP/Reno]

• Đặt kích thước gói tin

\$tcp set packetSize_ \$pksize

• Đặt kích thước cửa sổ trượt

\$tcp set window_ 64

\$tcp set maxcwnd_ 64

• Tạo đích

set sink [new Agent/TCPSink]

- Gắn nguồn vào nút s0

\$ns attach-agent \$s0 \$tcp

- Gắn đích vào nút d3

\$ns attach-agent \$d3 \$sink

- Kết nối nguồn với đích

\$ns connect \$tcp \$sink

- Đặt Flow ID

\$tcp set fid_ 1

- Thiết lập ứng dụng phát FTP cho nguồn TCP

set ftp [new Application/FTP]

\$ftp attach-agent \$tcp

\$ftp set type_ FTP

- + Tạo nguồn UDP

set udp [new Agent/UDP]

set null [new Agent/LossMonitor]

\$ns attach-agent \$s1 \$udp

\$ns attach-agent \$d2 \$null

\$ns connect \$udp \$null

\$udp set fid_ 2

- Thiết lập ứng dụng phát CBR cho nguồn UDP

set cbr [new Application/Traffic/CBR]

\$cbr set packetSize_ \$pksize

\$cbr set rate_ 4Mb

\$cbr set random_ false

\$cbr attach-agent \$udp

\$cbr set type_ CBR

\$ns at 0.1 "\$cbr start"

\$ns at 0.1 "\$ftp start"

\$ns at 50.1 "\$ftp stop"

\$ns at 50.1 "\$cbr stop"

\$ns at 50.2 "finish"

\$ns run

5.1.3. Viết file .awk

5.1.3.1. Tính lưu lượng các luồng

Bước 1: Trong folder *bai9* tạo 1 file mới đặt tên là *bandwidth.awk*

Bước 2: Mở file *bandwidth.awk* vừa tạo và bắt đầu viết thuật toán

- Mở đầu: Khai báo các biến, struct cần dùng

```
BEGIN {
```

```
i1=0; i2=0;
```

```
t1=0.1; t2=0.1;
```

```
k1=0; k2=0;
```

```
bw1[2000]; bw2[2000];
```

```
}
```

- Nội dung thuật toán:

```
{
```

- Đặt lại tên cho các phần của string trong file trace-all để tiện sử dụng

```
action = $1;
```

```
time = $2;
```

```
from = $3;
```

```
to = $4;
```

```
type = $5;
```

```
pktsize = $6;
```

```
flow_id = $8;
```

```
src = $9;
```

```
dst = $10;
```

```
seq_no = $11;
```

```
packet_id = $12;
```

```
if (flow_id == "1")
```

```
{
```

```
if ((action == "r") && (to == "3") && (time >= t1))
```

```

    { k1 += pktsize; }
    if (time >= (t1+0.5))
    {
        bw1[i1] = k1*8/0.5/1048576;
        print time " " bw1[i1] >> "tcpbw.tr"
        k1 = 0; t1 += 0.5; i1++;
    }
}
else
{
    if ((action == "r") && (to == "2") && (time >= t2))
    { k2 += pktsize; }
    if (time >= (t2+0.5))
    {
        bw2[i2] = k2*8/0.5/1048576;
        print time " " bw2[i2] >> "udpbw.tr"
        k2 = 0; t2 += 0.5; i2++;
    }
}
}
}
END{}

```

5.1.3.2. Tính trễ đầu cuối đến đầu cuối các luồng

Bước 1: Trong folder *bai9* tạo 1 file mới đặt tên là *delay.awk*

Bước 2: Mở file *delay.awk* vừa tạo và bắt đầu viết thuật toán

```

BEGIN {
    ts[2000];
    tr[2000];
    k1[101]; k2[101];
    d1[2000]; d2[2000];
    i1=0; i2=0; j1=0; j2=0;

```



```

}

{
  action = $1;
  time = $2;
  from = $3;
  to = $4;
  type = $5;
  pktsize = $6;
  flow_id = $8;
  src = $9;
  dst = $10;
  seq_no = $11;
  packet_id = $12;
  if (flow_id=="1")
  {
    if ((action == "+") && (from == "0"))
    { ts[packet_id] = time; }
    if ((action == "r") && (to == "3"))
    {
      tr[packet_id] = time;
      k1[i1] = (tr[packet_id] - ts[packet_id])*1000;
      d1[j1] += k1[i1];
      i1++;
    }
    if (i1>99)
    {
      d1[j1] /= 100;
      print time " " d1[j1] >> "tcpdl.tr";
      i1=0; t1=0; j1++;
    }
  }
}

```

```

}
else
{
    if ((action == "+") && (from == "1"))
    { ts[packet_id] = time; }
    if ((action == "r") && (to == "2"))
    {
        tr[packet_id] = time;
        k2[i2] = (tr[packet_id] - ts[packet_id])*1000;
        d2[j2] += k2[i2];
        i2++;
    }
    if (i2>99)
    {
        d2[j2] /= 100;
        print time " " d2[j2] >> "udpdL.tr";
        i2=0; t2=0; j2++;
    }
}
}
}
END {}

```

5.1.3.3. Tính xác suất mất gói trên các luồng

Bước 1: Trong folder *bai9* tạo 1 file mới đặt tên là *lostrate.awk*

Bước 2: Mở file *lostrate.awk* vừa tạo và bắt đầu viết thuật toán

```

BEGIN {
s1=0; s2=0;
l1=0; l2=0;
lr1[2000]; lr2[2000];
t1=0.1; t2=0.1;
i1=0; i2=0;

```

```

}

{
  action = $1;
  time = $2;
  from = $3;
  to = $4;
  type = $5;
  pktsize = $6;
  flow_id = $8;
  src = $9;
  dst = $10;
  seq_no = $11;
  packet_id = $12;
  if (flow_id == "1")
  {
    if ((action == "+") && (from == "0"))
    { ++s1; }
    if (action == "d")
    { ++l1;      }
    if (time >= (t1+0.5))
    {
      lr1[i1] = 100*l1/s1;
      print time " " lr1[i1] >> "tcplr.tr"
      t1 += 0.5; i1++;
    }
  }
}
else
{
  if ((action == "+") && (from == "1"))
  { ++s2; }

```

```

if (action == "d")
{ ++l2; }
if (time >= (t2+0.5))
{
    lr2[i2] = 100*l2/s2;
    print time " " lr2[i2] >> "udplr.tr"
    t2 += 0.5; i2++;
}
}
}
END {}

```

5.2. Mô phỏng

Mở Terminal và tiến hành mô phỏng:

```

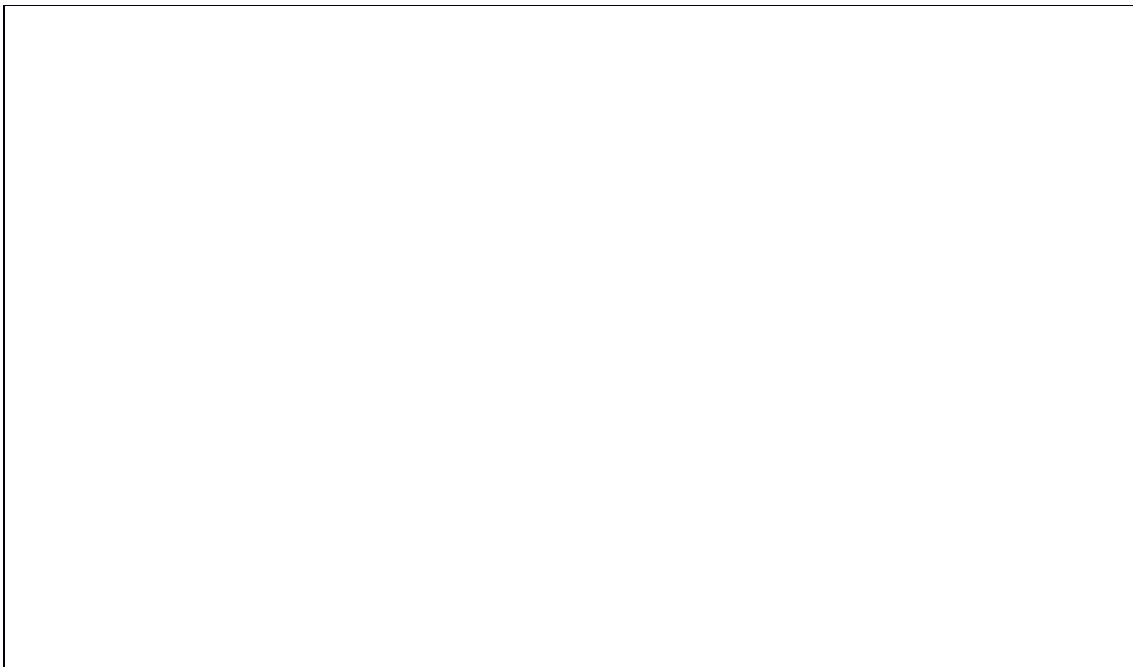
cd ~/Desktop/[MSSV]/bai9
ns bai9.tcl

```

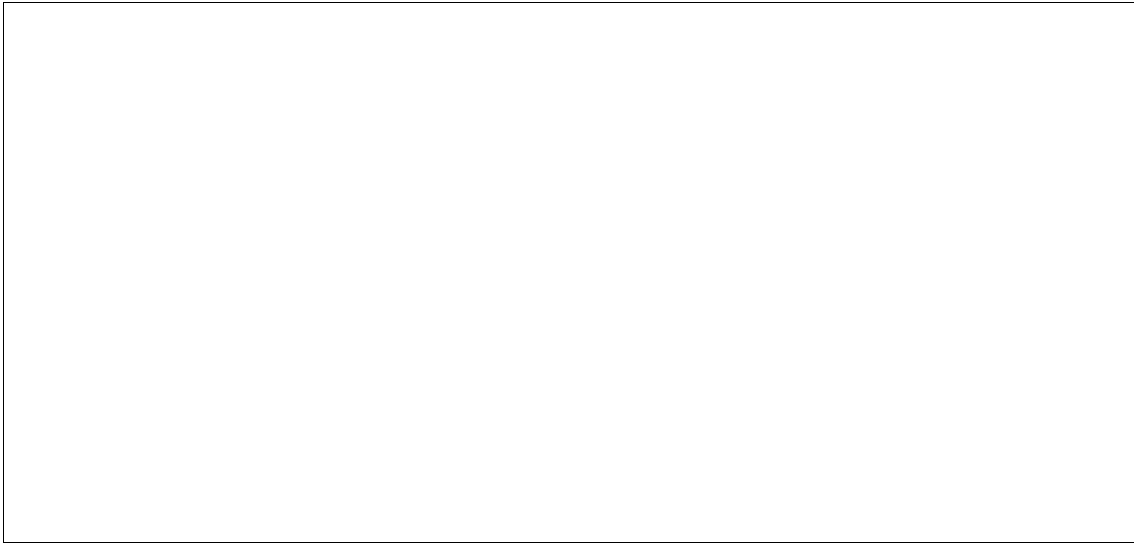
Ngoài NAM và bảng GUI, lần này sẽ có thêm 3 đồ thị xuất hiện, ứng với lưu lượng, trễ và tỉ lệ mất gói.

5.3. Kết quả

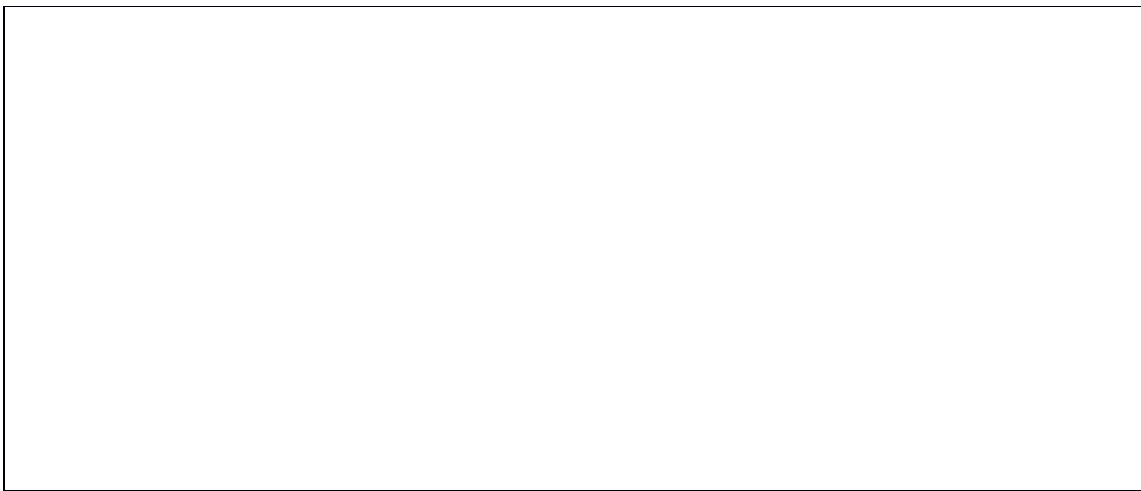
5.3.1. Đồ thị



Đồ thị 5.1. Bảng thông các luồng.



Đồ thị 5.2. Trễ đầu cuối đến đầu cuối các luồng.



Đồ thị 5.3. Tỷ lệ mất gói các luồng.

5.3.2. Giá trị trung bình

Xử lý số liệu thu được ở các file trace .tr mới được tạo ra do sử dụng file .awk rồi điền vào bảng sau.

Lưu lượng (Mbps)	TCP	
	UDP	
Trễ (ms)	TCP	
	UDP	
Tỷ lệ mất gói (%)	TCP	
	UDP	

Chú ý: Cách lưu lại đồ thị Xgraph để sử dụng cho Windows:

Bước 1: Sau khi đã chạy mô phỏng và xuất hiện cửa sổ đồ thị Xgraph, ấn vào nút Hardcopy (hdcpy)

Output device: Postscript

Disposition: To File

File or Device Nam: [name].ps

Ok

Sau bước 1, thu được file .ps ở trong folder chứa file .tcl.

Bước 2: Mở file .ps, vào phần Print

Trong tab General:

Printer: Print to File

Output format: PDF

Print

Sau bước 2, thu được file .pdf của đồ thị Xgraph và có thể mang sang sử dụng bên Windows

VI. KẾT LUẬN

Qua bài thí nghiệm này, sinh viên đã được:

- Học cách sử dụng NS-2 để tính toán các thông số của mạng.
- Sử dụng Xgraph để vẽ đồ thị biểu diễn các thông số của mạng theo thời gian.

VII. CÂU HỎI KIỂM TRA

1. Giải thích công thức tính lưu lượng trong file ***bandwidth.awk***?

$bw1[i1] = k1 * 8 / 0.5 / 1048576$

.....

.....

.....

.....

.....

.....

2. Viết file .awk để tính delay giữa nút s1 và nút d2 của luồng tcp?

[illegible]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Đồ thị Xgraph thu được khi áp dụng thuật toán trên?

