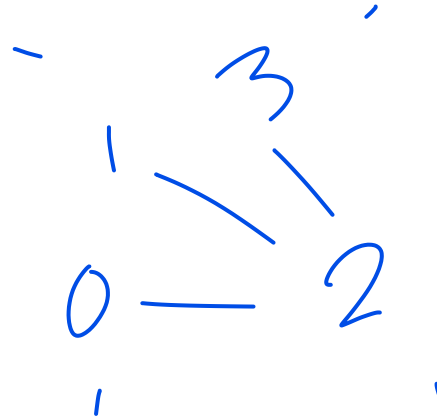


```

1
2 # ALL PART -1
3
4 #gedit ex1.tcl
5 #ns ex1.tcl
6
7 #gedit script.awk
8 #awk -f script.awk ex1.tr
9
10 #grep -c "r" ex2.tr
11 #grep "^r" ex2.tr      #or
12
13
14 #1.....
15
16 #duplex-link n0-n2, n1-n3 (n1-n2, n2-n3)
17 #TCP agent b/w n0-n3
18 #UDP agent b/w n1-n3
19 #applications over TCP and UDP agents
20 #queue size to 5
21 #vary the bandwidth to find the no. of packets dropped and recieved by TCP/UDP
22 #using awk script and grep command
23
24
25 #5
26 set ns [new Simulator]
27 set tf [open ex1.tr w]
28 set nf [open ex1.nam w]
29 $ns trace-all $tf
30 $ns namtrace-all $nf
31
32 set n0 [$ns node]
33 set n1 [$ns node]
34 set n2 [$ns node]
35 set n3 [$ns node]
36
37 $n0 label "TCP Source"
38 $n3 label "TCP Sink"
39 $n1 label "UDP Source"
40 $n3 label "UDP Null"
41
42         #bandwidth,delay,queuing disciplin
43 $ns duplex-link $n0 $n2 2Mb 2ms DropTail
44 $ns duplex-link $n1 $n2 2Mb 2ms DropTail
45 $ns duplex-link $n2 $n3 0.4Mb 10ms DropTail
46 $ns queue-limit $n0 $n2 5
47
48 set tcp [new Agent/TCP]
49 set sink [new Agent/TCPSink]
50 set ftp [new Application/FTP]
51 $ns attach-agent $n0 $tcp

```



```

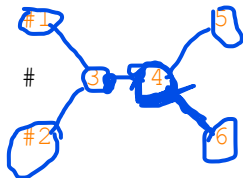
52 $ns attach-agent $n3 $sink
53 $ns connect $tcp $sink
54 $ftp attach-agent $tcp
55 #7
56 set udp [new Agent/UDP]
57 set null [new Agent/Null]
58 set cbr [new Application/Traffic/CBR]    #traffic only in cbr
59 $ns attach-agent $n1 $udp
60 $ns attach-agent $n3 $null
61 $ns connect $udp $null
62 $cbr attach-agent $udp
63
64 $ns at 0.1 "$ftp start"
65 $ns at 1.1 "$cbr start"
66 $ns at 10.0 "finish"
67
68 proc finish {} { #7
69     global ns tf nf
70     $ns flush-trace
71     close $tf
72     close $nf
73     puts "running nam..."
74     exec nam ex1.nam &
75     exit 0
76 }
77 $ns run
78
79 #awk -f ex1.awk ex1.tr
80
81 BEGIN {
82     tcp_d = 0;
83     tcp_r = 0;
84     udp_d = 0;
85     udp_r = 0;
86 }
87 {
88     if($1 == "d" && $5 == "tcp")
89         tcp_d++;
90     if($1 == "r" && $5 == "tcp")
91         tcp_r++;
92     if($1 == "d" && $5 == "cbr")
93         udp_d++;
94     if($1 == "r" && $5 == "cbr")
95         udp_r++;
96 }
97 END{
98     printf("TCP: No. of packets: recieved = %d, dropped = %d", tcp_r, tcp_d);
99     printf("UDP: No. of packets: recieved = %d, dropped = %d", udp_r, udp_d);
100 }
101
102

```

```

103 #2.....
104
105 #FTP b/w the nodes n1-n6      (0&5)
106 #Telnet b/w nodes n2-n5.      (1&4)
107 #congestion window
108 #throughput
109
110 set ns [new Simulator]
111 set tf [open ex2.tr w]
112 set nf [open ex2.nam w]
113 set cwind [open win2.tr w] #***dhyan
114 $ns trace-all $tf
115 $ns namtrace-all $nf
116
117 $ns color 1 Blue
118 $ns color 2 Red
119
120 set n1 [$ns node]
121 set n2 [$ns node]
122 set n3 [$ns node]
123 set n4 [$ns node]
124 set n5 [$ns node]
125 set n6 [$ns node]
126
127 $n1 label "FTP Source"
128 $n6 label "FTP Sink"
129 $n2 label "Telnet Source"
130 $n5 label "Telnet Sink"
131
132 $ns duplex-link $n1 $n3 2Mb 2ms DropTail
133 $ns duplex-link $n2 $n3 2Mb 2ms DropTail
134 $ns duplex-link $n3 $n4 0.4Mb 5ms DropTail
135 $ns duplex-link $n4 $n5 2Mb 2ms DropTail
136 $ns duplex-link $n4 $n6 2Mb 2ms DropTail
137 #set queue-limit $n3 $n4 10
138
139 $ns duplex-link-op $n1 $n3 orient down-right
140 $ns duplex-link-op $n2 $n3 orient up-right
141 $ns duplex-link-op $n3 $n4 orient right
142 $ns duplex-link-op $n4 $n5 orient up-right
143 $ns duplex-link-op $n4 $n6 orient down-right
144
145 set tcp1 [new Agent/TCP]
146 set sink1 [new Agent/TCPSink]
147 set ftp1 [new Application/FTP]
148 $ns attach-agent $n1 $tcp1
149 $ns attach-agent $n6 $sink1
150 $ns connect $tcp1 $sink1
151 $ftp1 attach-agent $tcp1
152
153 set tcp2 [new Agent/TCP]

```



```

154 set sink2 [new Agent/TCPSink]
155 set telnet1 [new Application/Telnet]
156 $ns attach-agent $n2 $tcp2
157 $ns attach-agent $n5 $sink2
158 $ns connect $tcp2 $sink2
159 $telnet1 attach-agent $tcp2
160
161 $tcp1 set fid_ 1
162 $tcp2 set fid_ 2
163
164 $ns at 1.2 "$ftp1 start"
165 $ns at 5.0 "$ftp1 stop"
166 $ns at 5.1 "$telnet1 start"
167 $ns at 10.0 "finish"
168
169 proc plotWindow {tcpSource file} { #6
170     global ns
171     set time 0.01                #interval
172     set now [$ns now]            #returns the current simulation time
173     set cwnd [$tcpSource set cwnd_] #conjection window size
174     puts $file "$now $cwnd"      #printing time and cwnd value #to plot the graph
175     $ns at [expr $now + $time] "plotWindow $tcpSource $file" #time at which to plot #now+0.01
176 }
177
178 $ns at 2.0 "plotWindow $tcp1 $cwind"
179 $ns at 5.5 "plotWindow $tcp2 $cwind"
180
181 proc finish {} { #7+1=8
182     global ns tf nf cwind
183     $ns flush-trace
184     close $tf
185     close $nf
186     #close $cwind
187     puts "running nam..."
188     exec nam ex2.nam &
189     exec xgragh win2.tr &
190     exit 0
191 }
192 $ns run
193
194 #awk -f ex2.awk ex2.tr
195
196 BEGIN{
197     last = 0;
198     tcp_sz = 0;
199     cbr_sz = 0;
200     total_sz = 0;
201 }
202 {
203     action=$1;
204     time=$2;

```

```

205     from=$3;
206     to=$4;
207     type=$5;
208     pktsize=$6;
209
210     if($1 == "r" && $5 == "tcp" && to == "3")
211         tcp_sz+=pktsize;
212     if($1 == "r" && $5 == "cbr" && to == "3")
213         cbr_sz+=pktsize;
214     total_sz+=pktsize;
215 }
216
217 END {
218     printf("Time = %f\n", time);
219     last = total_sz;           # Assign total_sz to last for reference
220     printf("Throughput = %f Mbps\n", (total_sz * 8 / 1000000)); #throughput in Mbps
221     printf("TCP Throughput = %f Mbps\n", (tcp_sz * 8 / 1000000)); #throughput of TCP packets in Mbps
222 }
223
224
225 #grep -c "r" ex2.tr
226
227 #grep "^r" ex2.tr    (or)

```

```

230 #3.....
231
232 #Distance vector routing protocol.
233 #link b/w node 1 and 4 breaks at 1.0 ms, comes up at 3.0 ms.
234 #source node 0 transmits packets to node 4.
235 #congestion window when TCP sends packets via other nodes.
236 #own parameters for bandwidth and delay.
237
238 set ns [new Simulator]
239 set tf [open ex3.tr w]
240 set nf [open ex3.nam w]
241 set cwind [open win3.tr w]      #***dhyan
242 $ns trace-all $tf
243 $ns namtrace-all $nf
244
245 $ns rtproto DV                 #protocol, built in
246 #set the protocol as Dist Vector #dynamic routing protocol #updates source every 2 ms
247
248 set n0 [$ns node]
249 set n1 [$ns node]
250 set n2 [$ns node]
251 set n3 [$ns node]
252 set n4 [$ns node]
253 set n5 [$ns node]
254
255 $n0 label "FTP source"

```

```

256 $n4 label "FTP sink"
257
258 $ns color 1 Orange
259
260 $ns duplex-link $n0 $n1 1Mb 10ms DropTail
261 $ns duplex-link $n1 $n4 1Mb 10ms DropTail
262 $ns duplex-link $n4 $n5 1Mb 10ms DropTail
263 $ns duplex-link $n0 $n2 1Mb 10ms DropTail
264 $ns duplex-link $n2 $n3 1Mb 10ms DropTail
265 $ns duolex-link $n3 $n5 1Mb 10ms DropTail
266 #not neccesary
267 $ns queue-limit $n2 $n3 10
268 $ns queue-limit $n1 $n4 10
269
270 $ns duplex-link-op $n0 $n1 orient up-right
271 $ns duplex-link-op $n1 $n4 orient right
272 $ns duplex-link-op $n4 $n5 orient down-right
273 $ns duplex-link-op $n0 $n2 orient down-right
274 $ns duplex-link-op $n2 $n3 orient right
275 $ns duplex-link-op $n3 $n5 orient up-right
276
277 set tcp [new Agent/TCP]
278 set sink [new Agent/TCPSink]
279 set ftp [new Application/FTP]
280 $ns attach-agent $n0 $tcp
281 $ns attach-agent $n4 $sink
282 $ns connect $tcp $sink
283 $ftp attach-agent $tcp
284 $tcp set fid_ 1
285
286 $ns rtmodel-at 1.0 down $n1 $n4 } #break link b/w these 2 nodes,the path-blocked=>packets-alternate path to reach destination.
287 $ns rtmodel-at 3.0 up $n $n4 } #connect the link back-packets-transmitted-original path
288
289 $ns at 0.1 "$ftp start"
290 $ns at 10.0 "finish"
291
292 proc plotWindow {tcpSource file} {
293     global ns
294     set time 0.01
295     set now [$ns now]
296     set cwnd [$tcpSource set cwnd_]
297     puts $file "$now $cwnd"
298     $ns at [expr $now+$time] "plotWindow $tcpSource $file"
299 }
300 $ns at 1.0 "plotWindow $tcp $cwnd"
301
302 proc finish {} { #7+2=9
303     global ns tf nf cwind
304     $ns flush-trace
305     close $tf
306     close $nf

```

```

307     close $cwind
308     puts "running nam..."
309     exec nam ex3.nam &
310     exec xgraph win3.tr &
311     exit 0
312 }
313
314 $ns run

```

```

317 #4.....
318

```

```

319 #server is running a FTP application over TCP.
320 #client sends a request to download a file of size 10Mb form the server.
321 #node n0 - server , node n1 - client.
322 #TCP packet size is 1500 Bytes
323

```

```

324 set ns [new Simulator]
325 set tf [open ex4.tr w]
326 set nf [open ex4.nam w]
327 $ns trace-all $tf
328 $ns namtrace-all $nf
329

```

```

330 set s [$ns node]
331 set c [$ns node]
332 $s label "Server"
333 $c label "Client"
334 $ns color 1 Blue
335

```

```

336 $ns duplex-link $s $c 10Mb 22ms DropTail
337 $ns duplex-link-op $s $c orient right
338

```

```

339 set tcp [new Agent/TCP]
340 set sink [new Agent/TCPSink]
341 set ftp [new Application/FTP]
342 $ns attach-agent $s $tcp
343 $ns attach-agent $c $sink
344 $ns connect $tcp $sink
345 $ftp attach-agent $tcp
346

```

```

347 $tcp set packetsize_ 1500
348 $tcp set fid_ 1
349

```

```

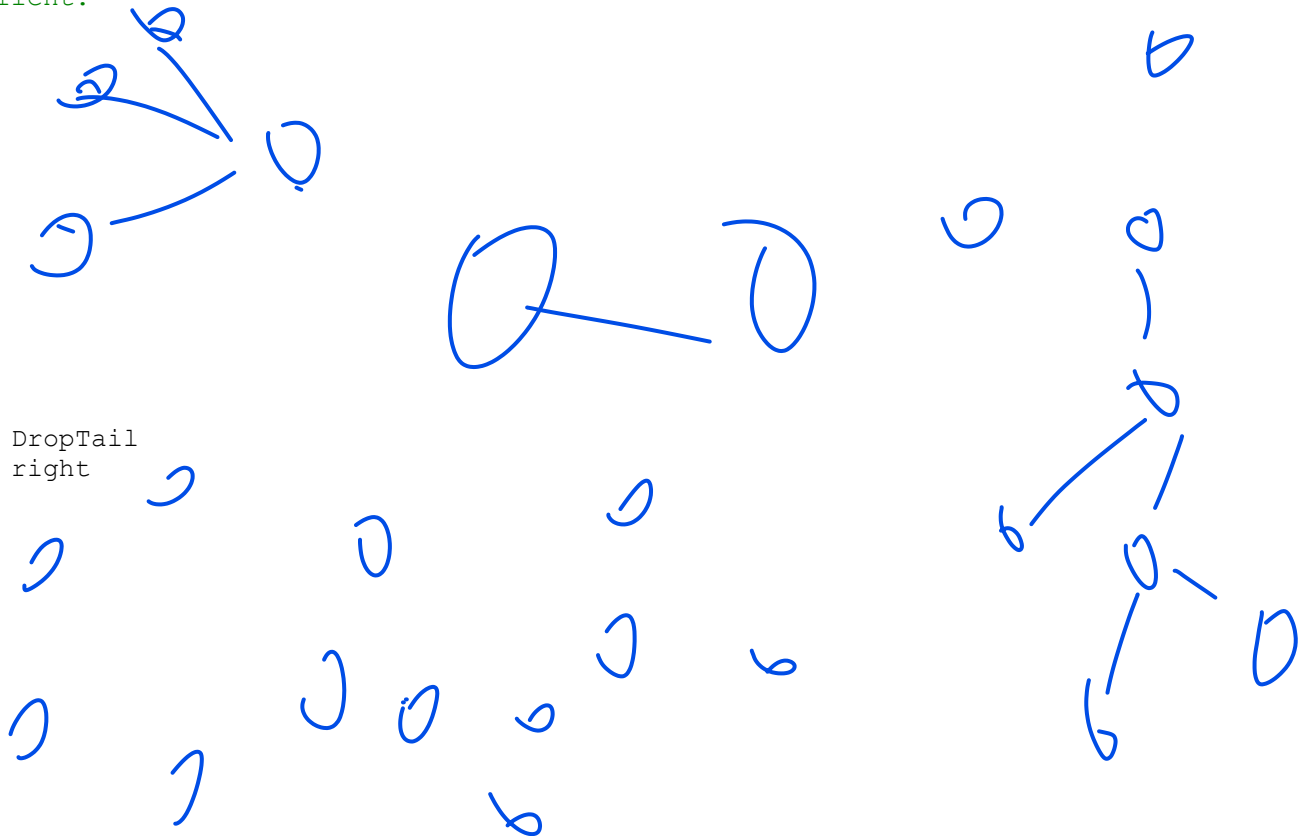
350 $ns at 0.01 "$ftp start"
351 $ns at 15.0 "$ftp stop"
352 $ns at 15.1 "finish"
353

```

```

354 proc finish {} {
355     global ns tf nf
356     $ns flush-trace
357     close $tf

```



```

358     close $nf
359     exec nam ex4.nam &
360     exec awk -f ex4transfer.awk ex4.tr &
361     exec awk -f ex4convert.awk ex4.tr > convert.tr &    #executes ex4convert.awk and redirects the o/p to convert.tr
362     exec xgraph convert.tr -geometry 800*400 -t "bytes_received_at_client" -x "time_in_secs" -y "bytes-in-bps" &
363 }
364 $ns run
365
366 #transfer.awk
367
368 BEGIN {
369     count = 0;
370     time = 0;          # $2
371     total_bytes_sent = 0;
372     total_bytes_received = 0;
373 }
374 {
375     if($1=="r" && $5=="tcp" && $4==1)    #action, type, to
376         total_bytes_received += $6;    #pktsize
377     if($1=="+" && $5=="tcp" && $3==0)    #action, type, from
378         total_bytes_sent += $6;
379 }
380
381 END{
382     system("clear");
383     printf("Transmission time required to transfer the file = %f", $2);
384     printf("Actual data sent from server = %f Mbps", (total_bytes_sent)/1000000);
385     printf("Data recieved by client = %f Mbps", (total_bytes_received)/1000000);
386 }
387
388 #ex4convert.awk
389
390 #to copy values from awk script- ex4convert.awk to a trace file- convert.tr from which the graph can be plotted
391 BEGIN {
392     count = 0;
393     time = 0;
394 }
395 {
396     if ($1=="r" && $5=="tcp" && $4==1) {
397         count += $6;    #pktsize
398         time = $2;
399         printf("%f %f", time, (count)/1000000);
400     }
401 }
402 END{
403 }

```

.....

```

407
408 #multicast routing protocol

```



```
409 #own parameters for bandwidth delay
410
411 set ns [new Simulator -multicast on]
412 set tf [open mcast.tr w]
413 set nf [open mcast.nam w]
414 $ns trace-all $tf
415 $ns namtrace-all $nf
416
417 set n0 [$ns node]
418 set n1 [$ns node]
419 set n2 [$ns node]
420 set n3 [$ns node]
421 set n4 [$ns node]
422 set n5 [$ns node]
423 set n6 [$ns node]
424 set n7 [$ns node]
425
426 $n0 label "Source 1"
427 $n1 label "Source 2"
428 $n5 label "Reciever 1"
429 $n6 label "Reciever 2"
430 $n7 label "Reciever 3"
431
432 $n0 color Blue
433 $n1 color Blue
434 $n5 color Purple
435 $n6 color Purple
436 $n7 color Purple
437
438 $ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
439 $ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
440 $ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
441 $ns duplex-link $n3 $n7 1.5Mb 10ms DropTail
442 $ns duplex-link $n3 $n4 1.5mb 10ms DropTial
443 $ns duplex-link $n4 $n5 1.5Mb 10ms DropTail
444 $ns duplex-link $n4 $n6 1.5Mb 10ms DropTail
445
446 set mrproto DM
447 set mrthandle [$ns mrtproto $mrproto {}]
448
449 set group1 [Node allocaddr]
450 set group2 [Node allocaddr]
451
452 #
453 set udp0 [new Agent/UDP]
454 set cbr1 [new Application/Traffic/CBR]
455 $ns attach-agent $n0 $udp0
456 $cbr1 attach-agent $udp0
457
458 $udp0 set dst_addr_ $group1
459 $udp0 set dst_port_ 0
```

```

460 #
461 set udp1 [new Agent/UDP]
462 set cbr2 [new Application/Traffic/CBR]
463 $ns attach-agent $n1 $udp1
464 $cbr2 attach-agent $udp1
465
466 $udp1 set dst_addr_ $group2
467 $udp1 set dst_port_ 0
468 #
469
470 #
471 set rcvr1 [new Agent/Null]
472 $ns attach-agent $n5 $rcvr1
473
474 set rcvr2 [new Agent/Null]
475 $ns attach-agent $n6 $rcvr2
476
477 set rcvr3 [new Agent/Null]
478 $ns attach-agent $n7 $rcvr3
479 #
480 set rcvr4 [new Agent/Null]
481 $ns attach-agent $n5 $rcvr4
482
483 set rcvr5 [new Agent/Null]
484 $ns attach-agent $n6 $rcvr5
485
486 set rcvr6 [new Agent/Null]
487 $ns attach-agent $n7 $rcvr6
488 #
489
490 $ns at 1.0 "$n5 join-group $rcvr1 $group1"
491 $ns at 1.5 "$n6 join-group $rcvr2 $group1"
492 $ns at 2.0 "$n7 join-group $rcvr3 $group1"
493 $ns at 2.5 "$n5 join-group $rcvr4 $group2"
494 $ns at 3.0 "$n6 join-group $rcvr5 $group2"
495 $ns at 3.5 "$n7 join-group $rcvr6 $group2"
496
497 $ns at 4.0 "$n5 leave-group $rcvr1 $group1"
498 $ns at 4.5 "$n6 leave-group $rcvr2 $group1"
499 $ns at 5.0 "$n7 leave-group $rcvr3 $group1"
500 $ns at 5.5 "$n5 leave-group $rcvr4 $group2"
501 $ns at 6.0 "$n6 leave-group $rcvr5 $group2"
502 $ns at 6.5 "$n7 leave-group $rcvr6 $group2"
503
504 $ns at 0.5 "$cbr1 start"
505 $ns at 9.5 "$cbr1 stop"
506 $ns at 0.5 "$cbr2 start"
507 $ns at 9.5 "$cbr2 stop"
508 $ns at 10.0 "finish"
509
510 proc finish {} {

```

```
511     global ns tf nf
512     $ns flush-trace
513     close $tf
514     close $nf
515     exec nam mcast.nam &
516     exit 0
517 }
518
519 $ns run
520
521
522
```