```
3
4
     #duplex-link n0-n2, n1-n3
5
     #TCP agent b/w n0-n3
6
     #UDP agent b/w n1-n3
7
     #applications over TCP and UDP agents
8
     #queue size to 5
9
     #vary the bandwidth to find the no. of packets dropped and recieved by TCP/UDP
10
     #using awk script and grep command
11
12
     set ns [new Simulator]
13
     set tf [open ex1.tr w]
14
     set nf [open ex1.nam w]
15
     $ns trace-all $tf
16
     $ns namtrace-all $nf
17
18
     set n0 [$ns node]
19
     set n1 [$ns node]
20
     set n2 [$ns node]
21
     set n3 [$ns node]
22
23
     $n0 label "TCP Source"
     $n3 label "TCP Sink"
24
25
     $n1 label "UDP Source"
     $n3 label "UDP Null"
26
27
28
     $ns duplex-link $n0 $n2 2Mb 2ms DropTail
29
     $ns duplex-link $n1 $n2 2Mb 2ms DropTail
30
     $ns duplex-link $n2 $n3 0.4Mb 10ms DropTail
31
     $ns queue-limit $n0 $n2 5
32
33
     set tcp [new Agent/TCP]
34
     set sink [new Agent/TCPSink]
35
     set ftp [new Application/FTP]
36
     $ns attach-agent $n0 $tcp
37
     $ns attach-agent $n3 $sink
38
     $ns connect $tcp $sink
39
     $ftp attach-agent $tcp
40
41
     set udp [new Agent/UDP]
42
     set null [new Agent/Null]
43
     set cbr [new Application/Traffic/CBR]
44
     $ns attach-agent $n1 $udp
45
     $ns attach-agent $n3 $null
46
     $ns connect $udp $null
47
     $cbr attach-agent $udp
48
     $ns at 0.1 "$ftp start"
49
     $ns at 1.1 "$cbr start"
50
    $ns at 10.0 "finish"
51
52
53
   proc finish {} {
54
         global ns tf nf
55
         $ns flush-trace
56
         close $tf
57
         close $nf
58
         puts "running nam..."
59
         exec nam ex1.nam &
60
         exit 0
61
    }
62
63
    $ns run
64
65
     #awk -f ex1.awk ex1.tr
66
67
    BEGIN {
68
        tcp d = 0;
69
         tcp_r = 0;
70
         udp_d = 0;
71
         udp_r = 0;
72
     }
73
     {
```

```
74
          if($1 == "d" && $5 == "tcp")
 75
              tcp d++;
 76
          if($1 == "r" && $5 == "tcp")
 77
              tcp r++;
          if($1 == "d" && $5 == "cbr")
 78
 79
              udp d++;
 80
          if($1 == "r" && $5 == "cbr")
 81
              udp r++;
 82
      1
 83
     END{
 84
          printf("TCP: No. of packets: recieved = %d, dropped = %d", tcp_r, tcp_d);
 85
          printf("UDP: No. of packets: recieved = %d, dropped = %d", udp r, udp d);
 86
 87
 88
 89
 90
 91
      \#FTP b/w the nodes n1-n6
 92
      #Telnet b/w nodes n2-n5.
 93
      #congestion window
 94
      #throughput
 95
 96
      set ns [new Simulator]
 97
      set tf [open ex2.tr w]
      set nf [open ex2.nam w]
 98
 99
      set cwind [open win2.tr w]
100
      $ns trace-all $tf
101
      $ns namtrace-all $nf
102
103
      $ns color 1 Blue
104
     $ns color 2 Red
105
106
     set n1 [$ns node]
     set n2 [$ns node]
107
108
     set n3 [$ns node]
109
     set n4 [$ns node]
110
     set n5 [$ns node]
111
     set n6 [$ns node]
112
113
      $n1 label "FTP Source"
114
      $n6 label "FTP Sink"
115
      $n2 label "Telnet Source"
116
      $n5 label "Telnet Sink"
117
118
      $ns duplex-link $n1 $n3 2Mb 2ms DropTail
119
      $ns duplex-link $n2 $n3 2Mb 2ms DropTail
120
      $ns duplex-link $n3 $n4 0.4Mb 5ms DropTail
      $ns duplex-link $n4 $n5 2Mb 2ms DropTail
121
122
      $ns duplex-link $n4 $n6 2Mb 2ms DropTail
123
124
      $ns duplex-link-op $n1 $n3 orient down-right
125
      $ns duplex-link-op $n2 $n3 orient up-right
126
      $ns duplex-link-op $n3 $n4 orient right
127
      $ns duplex-link-op $n4 $n5 orient up-right
128
      $ns duplex-link-op $n4 $n6 orient down-right
129
130
      set tcp1 [new Agent/TCP]
131
      set sink1 [new Agent/TCPSink]
132
      set ftp1 [new Application/FTP]
133
      $ns attach-agent $n1 $tcp1
134
      $ns attach-agent $n6 $sink1
135
      $ns connect $tcp1 $sink1
136
      $ftp1 attach-agent $tcp1
137
138
      set tcp2 [new Agent/TCP]
139
      set sink2 [new Agent/TCPSink]
140
      set telnet1 [new Application/Telnet]
141
      $ns attach-agent $n2 $tcp2
142
      $ns attach-agent $n6 $sink2
143
      $ns connect $tcp2 $sink2
144
      $telnet1 attach-agent $tcp2
145
146
      $tcp1 set fid_ 1
```

```
$tcp2 set fid 2
147
148
149
      $ns at 1.2 "$ftp1 start"
      $ns at 5.0 "$ftp1 stop"
150
151
      $ns at 5.1 "telnet1 start"
152
153
      proc plotWindow {tcpSource file} {
154
          global ns
          set time 0.01
155
156
          set now[$ns now]
157
          set cwnd [$tcpSource set cwnd ]
          puts $file "$now $cwnd"
158
          $ns at [expr $now + $time] "plotWindow $tcpSource $file"
159
160
      }
161
162
      $ns at 2.0 "plotWindow $tcp1 $cwind"
163
      $ns at 5.5 "plotWindow $tcp2 $cwind"
164
165
      proc finish {} {
166
          global ns tf nf cwind
167
          $ns flush-trace
168
          close $tf
169
          close $nf
170
          puts "running nam..."
171
          exec nam ex2.nam &
172
          exec xgragh win2.tr &
173
          exit 0
174
      }
175
176
      $ns run
177
178
      #awk -f ex2.awk ex2.tr
179
180
      BEGIN {
181
         last = 0;
182
         tcp_sz = 0;
183
          cbr sz = 0;
184
          total sz = 0;
185
      }
186
      {
187
          action=$1;
188
          time=$2;
189
          form=$3;
190
          to=$4;
191
          type=$5;
192
          pktsize=$6;
193
194
          if($1 == "r" && $5 == "tcp" && to == "4")
195
              tcp sz+=pktsize;
          if($1 == "r" && $5 == "cbr" && to == "4")
196
197
              cbr sz+=pktsize;
198
          total sz+=pktsize;
199
      }
     END{
200
201
          printf("Time = %f", time);
202
          last=total sz;
203
          printf("Throughput %f", (total sz*8/1000000)); #megabytes #1 byte=8bits
204
          print time(tcp_sz*8/1000000);
205
206
207
208
      /*
      END {
209
210
         printf("Time = %f\n", time);
                                                # Print the value of time
211
                                                # Assign total_sz to last for reference
          last = total sz;
212
213
          printf("Throughput = %f Mbps\n", (total sz * 8 / 1000000)); #throughput in Mbps
214
          printf("TCP Throughput = %f Mbps\n", (tcp sz * 8 / 1000000)); #throughput of
          TCP packets in Mbps
215
      }
216
217
218
      #grep -c "r" ex2.tr
```

```
220
221
222
223
      #Distance vector routing protocol.
224
      #link b/w node 1 and 4 breaks at 1.0 ms, comes up at 3.0 ms.
      #source node 0 transmits packets to node 4.
225
226
      #congestion window when TCP sends packets via other nodes.
227
      #own parameters for bandwidth and delay.
228
229
      set ns [new Simulator]
      set tf [open ex3.tr w]
230
      set nf [open ex3.nam w]
231
232
      set cwind [open win3.tr w]
233
      $ns trace-all $tf
234
      $ns namtrace-all $nf
235
      $ns rtproto DV
236
      #set the protocol as Dist Vector #dynamic routing protocol #updates source every 2 ms
237
238
      set n0 [$ns node]
239
      set n1 [$ns node]
240
      set n2 [$ns node]
241
      set n3 [$ns node]
242
      set n4 [$ns node]
243
      set n5 [$ns node]
244
245
      $ns color 1 Orange
246
247
      $ns duplex-link $n0 $n1 1Mb 10ms DropTail
248
      $ns duplex-link $n1 $n4 1Mb 10ms DropTail
249
      $ns duplex-link $n4 $n5 1Mb 10ms DropTail
250
      $ns duplex-link $n0 $n2 1Mb 10ms DropTail
251
      $ns duplex-link $n2 $n3 1Mb 10ms DropTail
252
      $ns duolex-link $n3 $n5 1Mb 10ms DropTail
253
254
      $ns duplex-link-op $n0 $n1 orient up-right
255
      $ns duplex-link-op $n1 $n4 orient right
256
      $ns duplex-link-op $n4 $n5 orient down-right
257
      $ns duplex-link-op $n0 $n2 orient down-right
258
      $ns duplex-link-op $n2 $n3 orient right
259
      $ns duplex-link-op $n3 $n5 orient up-right
260
      #not neccesary
      $ns queue-limit $n2 $n3 10
261
262
      $ns queue-limit $n1 $n4 10
263
264
      set tcp0 [new Agent/TCP]
265
      set sink0 [new Agent/TCPSink]
      set ftp0 [new Application/FTP]
266
267
      $ns attach-agent $n0 $tcp0
268
      $ns attach-agent $n4 $sink0
269
      $ftp0 attach-agent $tcp0
270
      $tcp0 set fid 1
271
272
      $ns rtmodel-at 1.0 down $n1 $n4
273
      $ns rtmodel-at 3.0 up $n $n4
274
275
      $ns at 0.1 "$ftp start"
276
      $ns at 10.0 "finish"
277
278
     proc plotWindow {tcpSource file} {
279
          global ns
280
          set time 0.01
281
          set now [$ns now]
282
          set cwnd [$tcpSource set cwnd ]
283
          puts $file "$now $cwnd"
284
          $ns at [expr $now+$time] "plotWindow $tcpSource $file"
285
      }
286
      $ns at 1.0 "plotWindow $tcp0 $cwind"
287
288
      proc finish {} {
289
          global ns tf nf cwind
290
          $ns flush-trace
291
          close $tf
```

219

```
292
          close $nf
293
          close $cwind
294
          puts "running nam..."
295
          exec nam ex3.nam &
296
          exec xgraph win3.tr &
297
          exit 0
298
      }
299
300
      $ns run
301
302
303
304
305
      #server is running a FTP application over TCP.
306
      #client sends a request to download a file of size 10Mb form the server.
307
      #node n0 - server , node n1 - client.
      #TCP packet size is 1500 Bytes
308
309
310
      set ns [new Simulator]
311
      set tf [open ex4.tr w]
312
      set nf [open ex4.nam w]
313
      $ns trace-all $tf
314
      $ns namtrace-all $nf
315
316
     set s [$ns node]
317
      set c [ns node]
      $s label "Server"
318
319
      $c label "Client"
320
      $ns color 1 Blue
321
322
      $ns duplex-link $s $c 10Mb 22ms DropTail
323
      $ns duplex-link-op $s $c orient right
324
325
      set tcp0 [new Agent/TCP]
326
      set sink0 [new Agent/TCPSink]
327
      set ftp0 [new Application/FTP]
328
      $ns attach-agent $s $tcp0
329
      $ns attach-agent $c $sink0
330
      $ns connect $tcp0 $sink0
331
      $ftp0 attach-agent $tcp0
332
333
      $tcp0 set packetsize 1500
334
      $tcp0 set fid_ 1
335
336
      proc finish {} {
337
          global ns tf nf
338
          $ns flush-trace
339
          close tf
340
          close nf
341
          exec nam ex4.nam &
342
          exec awk -f transfer.awk ex4.tr &
          exec awk -f convert.awk ex4.tr &
343
344
          exec xgragh convert.tr -geometry 800*400 -t "bytes_received_at_client" -x
          "time_in_secs" -y "bytes-in-bps" &
345
      }
346
347
      $ns at 0.01 "$ftp0 start"
348
      $ns at 15.0 "$ftp0 stop"
349
      $ns at 15.1 "finish"
350
351
      $ns run
352
353
      #transfer.awk
354
355
      BEGIN {
356
         count = 0;
357
          time = 0;
358
          total bytes sent = 0;
359
          total_bytes_received = 0;
360
      }
361
      {
362
          if($1=="r" && $5=="tcp" && $4==1)
363
              total_bytes_received += $6;
```

```
364
          if($1=="+" && $5=="tcp" && $3==0)
365
              total bytes sent += $6;
366
      1
367
      END{
368
369
          system("clear");
370
          printf("Tranmission time required to transfer the file = %f", $2);
371
          printf("Actual data sent form the server is %f Mbps", (total bytes sent)/1000000);
372
          printf("Data recieved by the client is %f Mbps", (total bytes received)/1000000);
373
      }
374
375
      #convert.awk
376
377
      BEGIN {
378
          count = 0;
379
          time = 0;
380
      }
381
382
          if ($1=="r" && $5=="tcp" && $4==1) {
383
              count += $6;
384
              time = $2;
385
              printf("%f %f", time,(count)/1000000);
386
387
      }
388
      END{
389
      }
390
391
392
393
394
      #multicast routing protocol
395
      #own parameters for bandwidth delay
396
397
      set ns [new Simulator -multicast on]
398
      set tf [open mcast.tr w]
399
      set nf [open mcast.nam w]
400
      $ns trace-all $tf
401
      $ns namtrace-all $nf
402
403
     set n0 [$ns node]
404
      set n1 [$ns node]
      set n2 [$ns node]
405
406
      set n3 [$ns node]
407
      set n4 [$ns node]
408
      set n5 [$ns node]
409
      set n6 [$ns node]
410
      set n7 [$ns node]
411
412
      $n0 color Blue
413
      $n1 color Blue
414
      $n5 color Purple
415
      $n6 color Purple
      $n7 color Purple
416
417
418
      $n0 label "Source 1"
      $n1 label "Source 2"
419
      $n5 label "Reciever 1"
420
421
      $n6 label "Reciever 2"
422
      $n7 label "Reciever 3"
423
424
      $ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
425
      $ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
426
      $ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
427
      $ns duplex-link $n3 $n7 1.5Mb 10ms DropTail
428
      $ns duplex-link $n3 $n4 1.5mb 10ms DropTial
429
      $ns duplex-link $n4 $n5 1.5Mb 10ms DropTail
430
      $ns duplex-link $n4 $n6 1.5Mb 10ms DropTail
431
432
      set mrproto DM
433
      set mrthandle [$ns mrtproto $mrproto {}]
434
435
      set group1 [Node allocaddr]
436
      set group2 [Node allocaddr]
```

```
437
438
      set udp0 [new Agent/UDP]
439
      set cbr1 [new Appliaction/Traffic/CBR]
440
      $ns attach-agent $n0 $udp0
441
      $udp0 set dst addr $group1
442
      $udp0 dst port 0
443
      $cbr1 attach-agent $udp0
444
      set udp1 [new Agent/UDP]
445
446
      set cbr2 [new Application/Traffic/CBR]
447
      $ns attach-agent $n1 $udp1
448
      $udp1 set dst_addr_ $group2
      $udp1 set dst_port_ 0
$cbr2 attach-agent $udp1
449
450
451
452
453
      set rcvr1 [new Agent/Null]
454
      $ns attach-agent $n5 $rcvr1
455
      $ns at 1.0 "$n5 join-group $rcvr1 $group1"
456
457
      set rcvr2 [new Agent/Null]
458
      $ns attach-agent $n6 $rcvr2
      $ns at 1.5 "$n6 join-group $rcvr2 $group1"
459
460
461
      set rcvr3 [new Agent/Null]
462
      $ns attach-agent $n7 $rcvr3
463
      $ns at 2.0 "$n7 join-group $rcvr3 $group1"
464
465
      set rcvr4 [new Agent/Null]
466
      $ns attach-agent $n5 $rcvr4
467
      $ns at 2.5 "$n5 join-group $rcvr4 $group2"
468
469
     set rcvr5 [new Agent/Null]
470
      $ns attach-agent $n6 $rcvr5
471
      $ns at 3.0 "$n6 join-group $rcvr5 $group2"
472
473
      set rcvr6 [new Agent/Null]
474
      $ns atatch-agent $n7 $rcvr6
475
      $ns at 3.5 "$n7 join-group $rcvr6 $group2"
476
477
      $ns at 4.0 "$n5 leave-group $rcvr1 $group1"
      $ns at 4.5 "$n6 leave-group $rcvr2 $group1"
478
479
      $ns at 5.0 "$n7 leave-group $rcvr3 $group1"
480
      $ns at 5.5 "$n5 leave-group $rcvr4 $group2"
481
      $ns at 6.0 "$n6 leave-group $rcvr5 $group2"
      $ns at 6.5 "$n7 leave-group $rcvr6 $group2"
482
483
      $ns at 0.5 "$cbr1 start"
484
      $ns at 9.5 "$cbr1 stop"
485
      $ns at 0.5 "$cbr2 start"
486
      $ns at 9.5 "$cbr2 stop"
487
488
      $ns at 10.0 "finish"
489
490
      proc finish {} {
491
          global ns tf nf
492
          $ns flush-trace
493
          close $tf
494
          close $nf
495
          exec nam mcat.nam &
496
          exit 0
497
      }
498
499
      $ns run
500
501
502
503
504
```

505