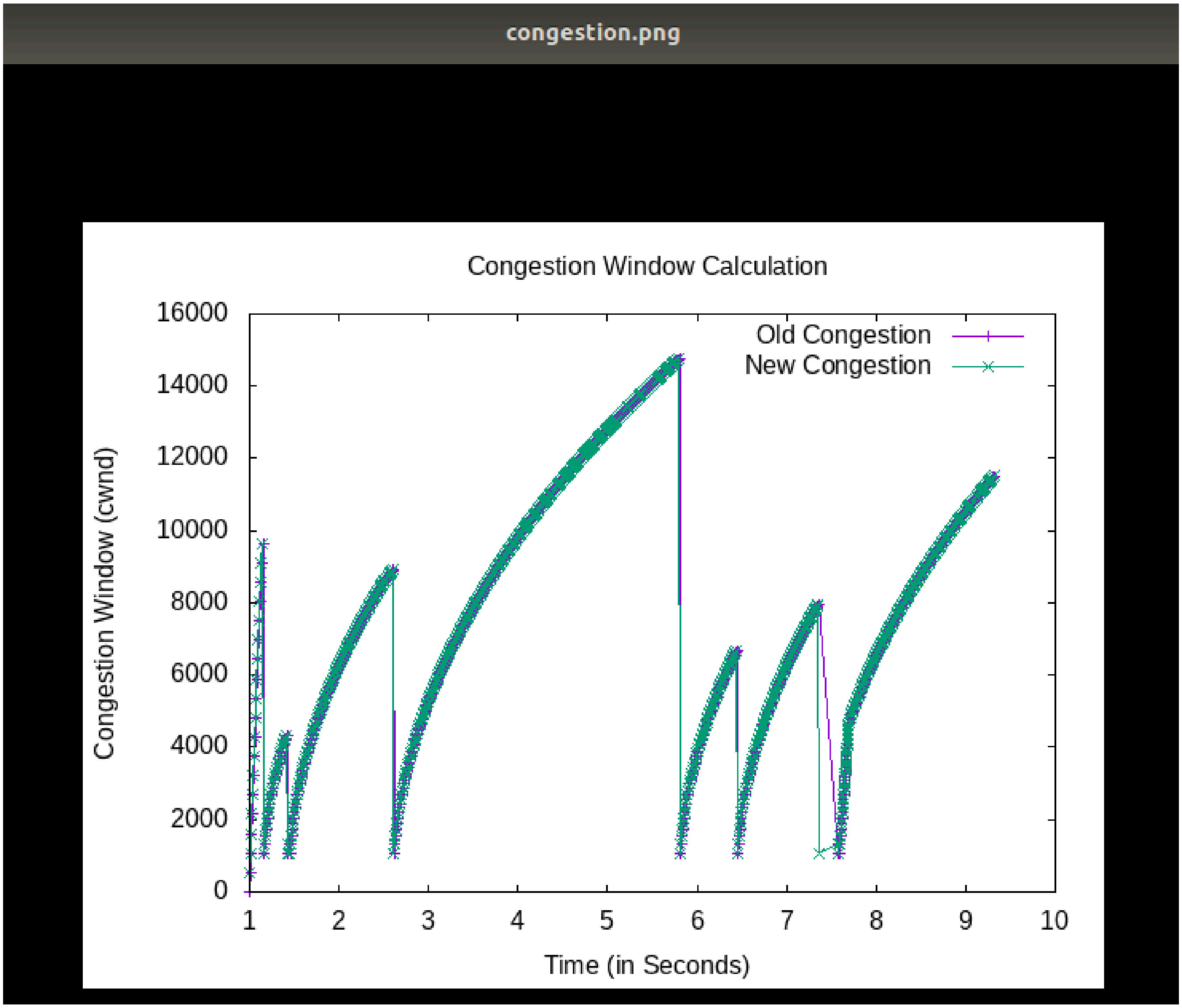
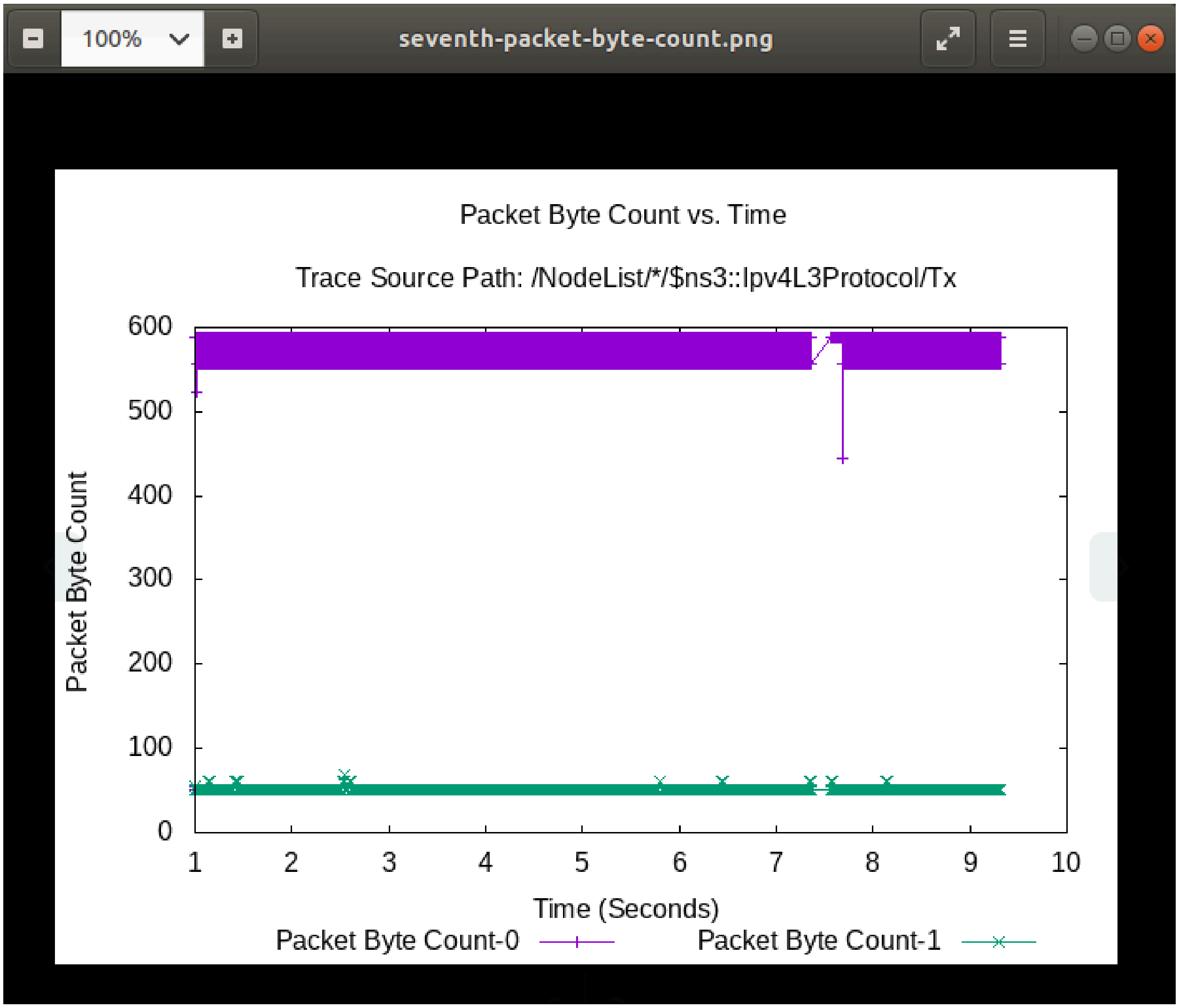
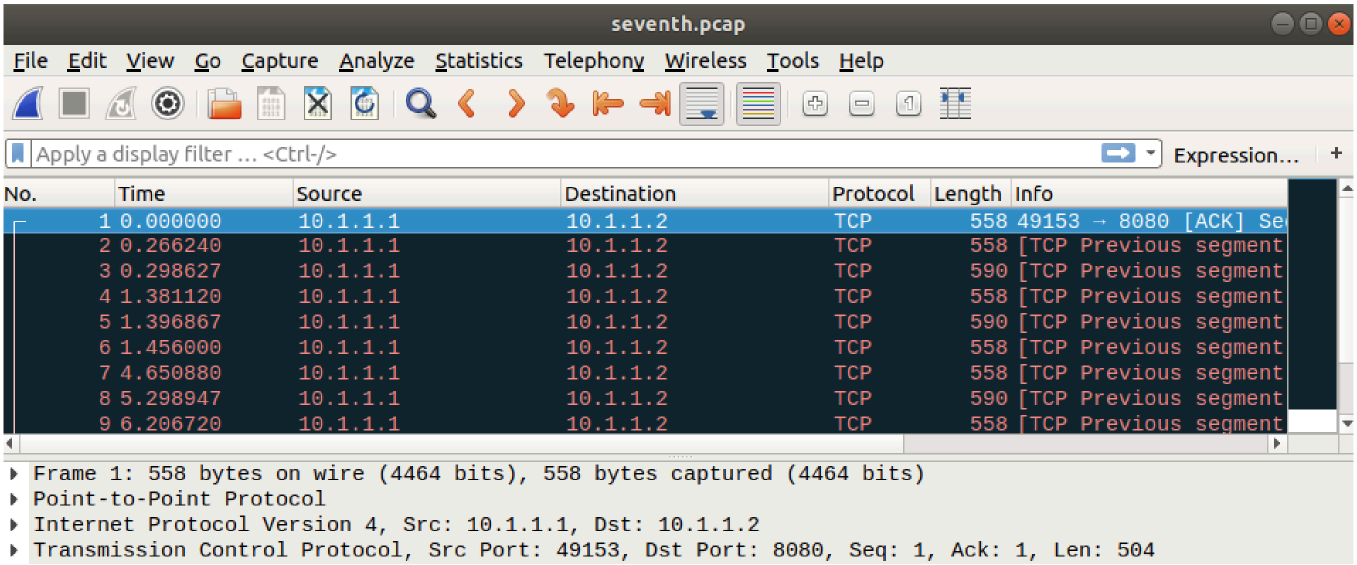
TCP Congestion Control:

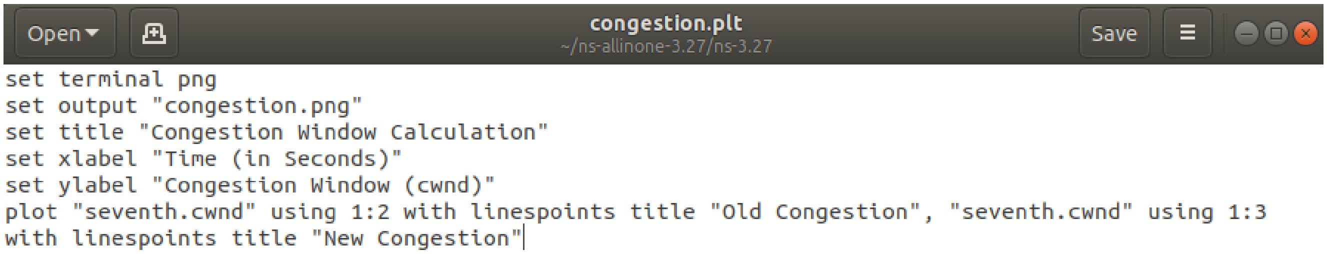





seventh.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

```
No.    Time       Source      Destination  Protocol  Length Info
     1 0.000000   10.1.1.1    10.1.1.2     TCP          558 49153 → 8080 [ACK] Se
     2 0.266240   10.1.1.1    10.1.1.2     TCP          558 [TCP Previous segment
     3 0.298627   10.1.1.1    10.1.1.2     TCP          590 [TCP Previous segment
     4 1.381120   10.1.1.1    10.1.1.2     TCP          558 [TCP Previous segment
     5 1.396867   10.1.1.1    10.1.1.2     TCP          590 [TCP Previous segment
     6 1.456000   10.1.1.1    10.1.1.2     TCP          558 [TCP Previous segment
     7 4.650880   10.1.1.1    10.1.1.2     TCP          558 [TCP Previous segment
     8 5.298947   10.1.1.1    10.1.1.2     TCP          590 [TCP Previous segment
     9 6.206720   10.1.1.1    10.1.1.2     TCP          558 [TCP Previous segment
```

```
▸ Frame 1: 558 bytes on wire (4464 bits), 558 bytes captured (4464 bits)
▸ Point-to-Point Protocol
▸ Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.1.2
▸ Transmission Control Protocol, Src Port: 49153, Dst Port: 8080, Seq: 1, Ack: 1, Len: 504
```



```
set terminal png
set output "congestion.png"
set title "Congestion Window Calculation"
set xlabel "Time (in Seconds)"
set ylabel "Congestion Window (cwnd)"
plot "seventh.cwnd" using 1:2 with linespoints title "Old Congestion", "seventh.cwnd" using 1:3
with linespoints title "New Congestion"
```

```
set terminal png
set output "congestion.png"
set title "Congestion Window Calculation"
set xlabel "Time (in Seconds)"
set ylabel "Congestion Window (cwnd)"
plot "seventh.cwnd" using 1:2 with linespoints title "Old Congestion", "seventh.cwnd" using 1:3
with linespoints title "New Congestion"
```

Seventh.cwnd

```
1.00419 0 536
1.0093 536 1072
1.01528 1072 1608
1.02167 1608 2144
1.02999 2144 2680
1.03831 2680 3216
1.04663 3216 3752
1.05495 3752 4288
1.06327 4288 4824
1.07159 4824 5360
1.07991 5360 5896
1.08823 5896 6432
1.09655 6432 6968
1.10487 6968 7504
1.11319 7504 8040
1.12151 8040 8576
1.12983 8576 9112
1.13815 9112 9648
1.1548 9648 1072
1.16481 1072 1340
1.17232 1340 1554
1.18064 1554 1738
1.18896 1738 1903
1.19728 1903 2053
1.2056 2053 2192
1.21392 2192 2323
1.22224 2323 2446
1.23056 2446 2563
```

```
1.23888 2563 2675
1.2472 2675 2782
1.25552 2782 2885
1.26384 2885 2984
1.27216 2984 3080
1.28048 3080 3173
1.2888 3173 3263
1.29712 3263 3351
1.30544 3351 3436
1.31376 3436 3519
1.32208 3519 3600
1.3304 3600 3679
1.33872 3679 3757
1.34704 3757 3833
1.35536 3833 3907
1.36368 3907 3980
1.372 3980 4052
1.38032 4052 4122
1.38864 4122 4191
1.39696 4191 4259
1.41272 4259 4326
1.42194 4326 1072
1.43105 1072 1340
1.451 1340 1072
1.461 1072 1340
1.46693 1340 1554
1.47285 1554 1738
1.48016 1738 1903
1.48848 1903 2053
1.4968 2053 2192
1.50512 2192 2323
1.51344 2323 2446
1.52176 2446 2563
1.53008 2563 2675
1.5384 2675 2782
1.54672 2782 2885
1.55504 2885 2984
1.56336 2984 3080
1.57168 3080 3173
1.58 3173 3263
1.58832 3263 3351
1.59664 3351 3436
1.60496 3436 3519
1.61328 3519 3600
```

```
1.6216 3600 3679
1.62992 3679 3757
1.63824 3757 3833
1.64656 3833 3907
1.65488 3907 3980
1.6632 3980 4052
1.67152 4052 4122
1.67984 4122 4191
1.68816 4191 4259
1.69648 4259 4326
1.7048 4326 4392
1.71312 4392 4457
1.72144 4457 4521
1.72976 4521 4584
1.73808 4584 4646
1.7464 4646 4707
1.75472 4707 4768
1.76304 4768 4828
1.77136 4828 4887
1.77968 4887 4945
1.788 4945 5003
1.79632 5003 5060
1.80464 5060 5116
1.81296 5116 5172
1.82128 5172 5227
1.8296 5227 5281
1.83792 5281 5335
1.84624 5335 5388
1.85456 5388 5441
1.86288 5441 5493
1.8712 5493 5545
1.87952 5545 5596
1.88784 5596 5647
1.89616 5647 5697
1.90448 5697 5747
1.9128 5747 5796
1.92112 5796 5845
1.92944 5845 5894
1.93776 5894 5942
1.94608 5942 5990
1.9544 5990 6037
1.96272 6037 6084
1.97104 6084 6131
1.97936 6131 6177
```

```
1.98768 6177 6223
1.996 6223 6269
2.00432 6269 6314
2.01264 6314 6359
2.02096 6359 6404
2.02928 6404 6448
2.0376 6448 6492
2.04592 6492 6536
2.05424 6536 6579
2.06256 6579 6622
2.07088 6622 6665
2.0792 6665 6708
2.08752 6708 6750
2.09584 6750 6792
2.10416 6792 6834
2.11248 6834 6876
2.1208 6876 6917
2.12912 6917 6958
2.13744 6958 6999
2.14576 6999 7040
2.15408 7040 7080
2.1624 7080 7120
2.17072 7120 7160
2.17904 7160 7200
2.18736 7200 7239
2.19568 7239 7278
2.204 7278 7317
2.21232 7317 7356
2.22064 7356 7395
2.22896 7395 7433
2.23728 7433 7471
2.2456 7471 7509
2.25392 7509 7547
2.26224 7547 7585
2.27056 7585 7622
2.27888 7622 7659
2.2872 7659 7696
2.29552 7696 7733
2.30384 7733 7770
2.31216 7770 7806
2.32048 7806 7842
2.3288 7842 7878
2.33712 7878 7914
2.34544 7914 7950
```

```
2.35376 7950 7986
2.36208 7986 8021
2.3704 8021 8056
2.37872 8056 8091
2.38704 8091 8126
2.39536 8126 8161
2.40368 8161 8196
2.412 8196 8231
2.42032 8231 8265
2.42864 8265 8299
2.43696 8299 8333
2.44528 8333 8367
2.4536 8367 8401
2.46192 8401 8435
2.47024 8435 8469
2.47856 8469 8502
2.48688 8502 8535
2.4952 8535 8568
2.50352 8568 8601
2.51184 8601 8634
2.5276 8634 8667
2.54187 8667 8700
2.55017 8700 8733
2.55344 8733 8765
2.56176 8765 8797
2.57008 8797 8829
2.5784 8829 8861
2.58672 8861 8893
2.60248 8893 8925
2.6117 8925 1072
2.62081 1072 1340
2.62832 1340 1554
2.63664 1554 1738
2.64496 1738 1903
2.65328 1903 2053
2.6616 2053 2192
2.66992 2192 2323
2.67824 2323 2446
2.68656 2446 2563
2.69488 2563 2675
2.7032 2675 2782
2.71152 2782 2885
2.71984 2885 2984
2.72816 2984 3080
```

```
2.73648 3080 3173
2.7448 3173 3263
2.75312 3263 3351
2.76144 3351 3436
2.76976 3436 3519
2.77808 3519 3600
2.7864 3600 3679
2.79472 3679 3757
2.80304 3757 3833
2.81136 3833 3907
2.81968 3907 3980
2.828 3980 4052
2.83632 4052 4122
2.84464 4122 4191
2.85296 4191 4259
2.86128 4259 4326
2.8696 4326 4392
2.87792 4392 4457
2.88624 4457 4521
2.89456 4521 4584
2.90288 4584 4646
2.9112 4646 4707
2.91952 4707 4768
2.92784 4768 4828
2.93616 4828 4887
2.94448 4887 4945
2.9528 4945 5003
2.96112 5003 5060
2.96944 5060 5116
2.97776 5116 5172
2.98608 5172 5227
2.9944 5227 5281
3.00272 5281 5335
3.01104 5335 5388
3.01936 5388 5441
3.02768 5441 5493
3.036 5493 5545
3.04432 5545 5596
3.05264 5596 5647
3.06096 5647 5697
3.06928 5697 5747
3.0776 5747 5796
3.08592 5796 5845
3.09424 5845 5894
```

```
3.10256 5894 5942
3.11088 5942 5990
3.1192 5990 6037
3.12752 6037 6084
3.13584 6084 6131
3.14416 6131 6177
3.15248 6177 6223
3.1608 6223 6269
3.16912 6269 6314
3.17744 6314 6359
3.18576 6359 6404
3.19408 6404 6448
3.2024 6448 6492
3.21072 6492 6536
3.21904 6536 6579
3.22736 6579 6622
3.23568 6622 6665
3.244 6665 6708
3.25232 6708 6750
3.26064 6750 6792
3.26896 6792 6834
3.27728 6834 6876
3.2856 6876 6917
3.29392 6917 6958
3.30224 6958 6999
3.31056 6999 7040
3.31888 7040 7080
3.3272 7080 7120
3.33552 7120 7160
3.34384 7160 7200
3.35216 7200 7239
3.36048 7239 7278
3.3688 7278 7317
3.37712 7317 7356
3.38544 7356 7395
3.39376 7395 7433
3.40208 7433 7471
3.4104 7471 7509
3.41872 7509 7547
3.42704 7547 7585
3.43536 7585 7622
3.44368 7622 7659
3.452 7659 7696
3.46032 7696 7733
```

```
3.46864 7733 7770
3.47696 7770 7806
3.48528 7806 7842
3.4936 7842 7878
3.50192 7878 7914
3.51024 7914 7950
3.51856 7950 7986
3.52688 7986 8021
3.5352 8021 8056
3.54352 8056 8091
3.55184 8091 8126
3.56016 8126 8161
3.56848 8161 8196
3.5768 8196 8231
3.58512 8231 8265
3.59344 8265 8299
3.60176 8299 8333
3.61008 8333 8367
3.6184 8367 8401
3.62672 8401 8435
3.63504 8435 8469
3.64336 8469 8502
3.65168 8502 8535
3.66 8535 8568
3.66832 8568 8601
3.67664 8601 8634
3.68496 8634 8667
3.69328 8667 8700
3.7016 8700 8733
3.70992 8733 8765
3.71824 8765 8797
3.72656 8797 8829
3.73488 8829 8861
3.7432 8861 8893
3.75152 8893 8925
3.75984 8925 8957
3.76816 8957 8989
3.77648 8989 9020
3.7848 9020 9051
3.79312 9051 9082
3.80144 9082 9113
3.80976 9113 9144
3.81808 9144 9175
3.8264 9175 9206
```

```
3.83472 9206 9237
3.84304 9237 9268
3.85136 9268 9298
3.85968 9298 9328
3.868 9328 9358
3.87632 9358 9388
3.88464 9388 9418
3.89296 9418 9448
3.90128 9448 9478
3.9096 9478 9508
3.91792 9508 9538
3.92624 9538 9568
3.93456 9568 9598
3.94288 9598 9627
3.9512 9627 9656
3.95952 9656 9685
3.96784 9685 9714
3.97616 9714 9743
3.98448 9743 9772
3.9928 9772 9801
4.00112 9801 9830
4.00944 9830 9859
4.01776 9859 9888
4.02608 9888 9917
4.0344 9917 9945
4.04272 9945 9973
4.05104 9973 10001
4.05936 10001 10029
4.06768 10029 10057
4.076 10057 10085
4.08432 10085 10113
4.09264 10113 10141
4.10096 10141 10169
4.10928 10169 10197
4.1176 10197 10225
4.12592 10225 10253
4.13424 10253 10281
4.14256 10281 10308
4.15088 10308 10335
4.1592 10335 10362
4.16752 10362 10389
4.17584 10389 10416
4.18416 10416 10443
4.19248 10443 10470
```

```
4.2008 10470 10497
4.20912 10497 10524
4.21744 10524 10551
4.22576 10551 10578
4.23408 10578 10605
4.2424 10605 10632
4.25072 10632 10659
4.25904 10659 10685
4.26736 10685 10711
4.27568 10711 10737
4.284 10737 10763
4.29232 10763 10789
4.30064 10789 10815
4.30896 10815 10841
4.31728 10841 10867
4.3256 10867 10893
4.33392 10893 10919
4.34224 10919 10945
4.35056 10945 10971
4.35888 10971 10997
4.3672 10997 11023
4.37552 11023 11049
4.38384 11049 11075
4.39216 11075 11100
4.40048 11100 11125
4.4088 11125 11150
4.41712 11150 11175
4.42544 11175 11200
4.43376 11200 11225
4.44208 11225 11250
4.4504 11250 11275
4.45872 11275 11300
4.46704 11300 11325
4.47536 11325 11350
4.48368 11350 11375
4.492 11375 11400
4.50032 11400 11425
4.50864 11425 11450
4.51696 11450 11475
4.52528 11475 11500
4.5336 11500 11524
4.54192 11524 11548
4.55024 11548 11572
4.55856 11572 11596
```

```
4.56688 11596 11620
4.5752 11620 11644
4.58352 11644 11668
4.59184 11668 11692
4.60016 11692 11716
4.60848 11716 11740
4.6168 11740 11764
4.62512 11764 11788
4.63344 11788 11812
4.64176 11812 11836
4.65008 11836 11860
4.6584 11860 11884
4.66672 11884 11908
4.67504 11908 11932
4.68336 11932 11956
4.69168 11956 11980
4.7 11980 12003
4.70832 12003 12026
4.71664 12026 12049
4.72496 12049 12072
4.73328 12072 12095
4.7416 12095 12118
4.74992 12118 12141
4.75824 12141 12164
4.76656 12164 12187
4.77488 12187 12210
4.7832 12210 12233
4.79152 12233 12256
4.79984 12256 12279
4.80816 12279 12302
4.81648 12302 12325
4.8248 12325 12348
4.83312 12348 12371
4.84144 12371 12394
4.84976 12394 12417
4.85808 12417 12440
4.8664 12440 12463
4.87472 12463 12486
4.88304 12486 12509
4.89136 12509 12531
4.89968 12531 12553
4.908 12553 12575
4.91632 12575 12597
4.92464 12597 12619
```

```
4.93296 12619 12641
4.94128 12641 12663
4.9496 12663 12685
4.95792 12685 12707
4.96624 12707 12729
4.97456 12729 12751
4.98288 12751 12773
4.9912 12773 12795
4.99952 12795 12817
5.00784 12817 12839
5.01616 12839 12861
5.02448 12861 12883
5.0328 12883 12905
5.04112 12905 12927
5.04944 12927 12949
5.05776 12949 12971
5.06608 12971 12993
5.0744 12993 13015
5.08272 13015 13037
5.09104 13037 13059
5.09936 13059 13080
5.10768 13080 13101
5.116 13101 13122
5.12432 13122 13143
5.13264 13143 13164
5.14096 13164 13185
5.14928 13185 13206
5.1576 13206 13227
5.16592 13227 13248
5.17424 13248 13269
5.18256 13269 13290
5.19088 13290 13311
5.1992 13311 13332
5.20752 13332 13353
5.21584 13353 13374
5.22416 13374 13395
5.23248 13395 13416
5.2408 13416 13437
5.24912 13437 13458
5.25744 13458 13479
5.26576 13479 13500
5.27408 13500 13521
5.2824 13521 13542
5.29072 13542 13563
```

```
5.29904 13563 13584
5.30736 13584 13605
5.31568 13605 13626
5.324 13626 13647
5.33232 13647 13668
5.34064 13668 13689
5.34896 13689 13709
5.35728 13709 13729
5.3656 13729 13749
5.37392 13749 13769
5.38224 13769 13789
5.39056 13789 13809
5.39888 13809 13829
5.4072 13829 13849
5.41552 13849 13869
5.42384 13869 13889
5.43216 13889 13909
5.44048 13909 13929
5.4488 13929 13949
5.45712 13949 13969
5.46544 13969 13989
5.47376 13989 14009
5.48208 14009 14029
5.4904 14029 14049
5.49872 14049 14069
5.50704 14069 14089
5.51536 14089 14109
5.52368 14109 14129
5.532 14129 14149
5.54032 14149 14169
5.54864 14169 14189
5.55696 14189 14209
5.56528 14209 14229
5.5736 14229 14249
5.58192 14249 14269
5.59024 14269 14289
5.59856 14289 14309
5.60688 14309 14329
5.6152 14329 14349
5.62352 14349 14369
5.63184 14369 14388
5.64016 14388 14407
5.64848 14407 14426
5.6568 14426 14445
```

```
5.66512 14445 14464
5.67344 14464 14483
5.68176 14483 14502
5.69008 14502 14521
5.6984 14521 14540
5.70672 14540 14559
5.71504 14559 14578
5.72336 14578 14597
5.73168 14597 14616
5.74 14616 14635
5.74832 14635 14654
5.75664 14654 14673
5.76496 14673 14692
5.77328 14692 14711
5.7816 14711 14730
5.79736 14730 14749
5.80658 14749 1072
5.81569 1072 1340
5.8232 1340 1554
5.83152 1554 1738
5.83984 1738 1903
5.84816 1903 2053
5.85648 2053 2192
5.8648 2192 2323
5.87312 2323 2446
5.88144 2446 2563
5.88976 2563 2675
5.89808 2675 2782
5.9064 2782 2885
5.91472 2885 2984
5.92304 2984 3080
5.93136 3080 3173
5.93968 3173 3263
5.948 3263 3351
5.95632 3351 3436
5.96464 3436 3519
5.97296 3519 3600
5.98128 3600 3679
5.9896 3679 3757
5.99792 3757 3833
6.00624 3833 3907
6.01456 3907 3980
6.02288 3980 4052
6.0312 4052 4122
```

```
6.03952 4122 4191
6.04784 4191 4259
6.05616 4259 4326
6.06448 4326 4392
6.0728 4392 4457
6.08112 4457 4521
6.08944 4521 4584
6.09776 4584 4646
6.10608 4646 4707
6.1144 4707 4768
6.12272 4768 4828
6.13104 4828 4887
6.13936 4887 4945
6.14768 4945 5003
6.156 5003 5060
6.16432 5060 5116
6.17264 5116 5172
6.18096 5172 5227
6.18928 5227 5281
6.1976 5281 5335
6.20592 5335 5388
6.21424 5388 5441
6.22256 5441 5493
6.23088 5493 5545
6.2392 5545 5596
6.24752 5596 5647
6.25584 5647 5697
6.26416 5697 5747
6.27248 5747 5796
6.2808 5796 5845
6.28912 5845 5894
6.29744 5894 5942
6.30576 5942 5990
6.31408 5990 6037
6.3224 6037 6084
6.33072 6084 6131
6.33904 6131 6177
6.34736 6177 6223
6.35568 6223 6269
6.364 6269 6314
6.37232 6314 6359
6.38064 6359 6404
6.38896 6404 6448
6.39728 6448 6492
```

```
6.4056 6492 6536
6.41392 6536 6579
6.42224 6579 6622
6.43056 6622 6665
6.44722 6665 1072
6.45722 1072 1340
6.46384 1340 1554
6.47216 1554 1738
6.48048 1738 1903
6.4888 1903 2053
6.49712 2053 2192
6.50544 2192 2323
6.51376 2323 2446
6.52208 2446 2563
6.5304 2563 2675
6.53872 2675 2782
6.54704 2782 2885
6.55536 2885 2984
6.56368 2984 3080
6.572 3080 3173
6.58032 3173 3263
6.58864 3263 3351
6.59696 3351 3436
6.60528 3436 3519
6.6136 3519 3600
6.62192 3600 3679
6.63024 3679 3757
6.63856 3757 3833
6.64688 3833 3907
6.6552 3907 3980
6.66352 3980 4052
6.67184 4052 4122
6.68016 4122 4191
6.68848 4191 4259
6.6968 4259 4326
6.70512 4326 4392
6.71344 4392 4457
6.72176 4457 4521
6.73008 4521 4584
6.7384 4584 4646
6.74672 4646 4707
6.75504 4707 4768
6.76336 4768 4828
6.77168 4828 4887
```

```
6.78 4887 4945
6.78832 4945 5003
6.79664 5003 5060
6.80496 5060 5116
6.81328 5116 5172
6.8216 5172 5227
6.82992 5227 5281
6.83824 5281 5335
6.84656 5335 5388
6.85488 5388 5441
6.8632 5441 5493
6.87152 5493 5545
6.87984 5545 5596
6.88816 5596 5647
6.89648 5647 5697
6.9048 5697 5747
6.91312 5747 5796
6.92144 5796 5845
6.92976 5845 5894
6.93808 5894 5942
6.9464 5942 5990
6.95472 5990 6037
6.96304 6037 6084
6.97136 6084 6131
6.97968 6131 6177
6.988 6177 6223
6.99632 6223 6269
7.00464 6269 6314
7.01296 6314 6359
7.02128 6359 6404
7.0296 6404 6448
7.03792 6448 6492
7.04624 6492 6536
7.05456 6536 6579
7.06288 6579 6622
7.0712 6622 6665
7.07952 6665 6708
7.08784 6708 6750
7.09616 6750 6792
7.10448 6792 6834
7.1128 6834 6876
7.12112 6876 6917
7.12944 6917 6958
7.13776 6958 6999
```

```
7.14608 6999 7040
7.1544 7040 7080
7.16272 7080 7120
7.17104 7120 7160
7.17936 7160 7200
7.18768 7200 7239
7.196 7239 7278
7.20432 7278 7317
7.21264 7317 7356
7.22096 7356 7395
7.22928 7395 7433
7.2376 7433 7471
7.24592 7471 7509
7.25424 7509 7547
7.26256 7547 7585
7.27088 7585 7622
7.2792 7622 7659
7.28752 7659 7696
7.29584 7696 7733
7.30416 7733 7770
7.31248 7770 7806
7.3208 7806 7842
7.32912 7842 7878
7.33744 7878 7914
7.3532 7914 7950
7.36242 7950 1072
7.56983 1072 1340
7.58496 1340 1072
7.59596 1072 1340
7.60194 1340 1554
7.60791 1554 1738
7.61389 1738 1903
7.61892 1903 2053
7.62395 2053 2192
7.62898 2192 2323
7.63087 2323 2446
7.63495 2446 2563
7.63684 2563 2675
7.64093 2675 2782
7.64282 2782 2885
7.6469 2885 2984
7.64879 2984 3080
7.65193 3080 3173
7.65382 3173 3263
```

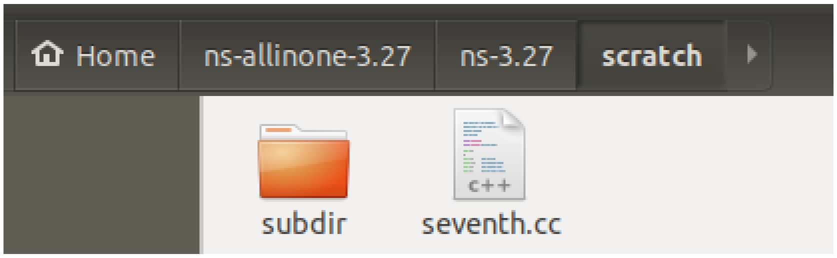
```
7.65696 3263 3351
7.65885 3351 3436
7.66074 3436 3519
7.66294 3519 3600
7.66482 3600 3679
7.66671 3679 3757
7.66891 3757 3833
7.6708 3833 3907
7.67269 3907 3980
7.67458 3980 4052
7.67646 4052 4122
7.67835 4122 4191
7.68024 4191 4259
7.68213 4259 4326
7.68402 4326 4392
7.6859 4392 4457
7.68779 4457 4521
7.68968 4521 4584
7.69157 4584 4646
7.69431 4646 4707
7.70263 4707 4768
7.71095 4768 4828
7.71927 4828 4887
7.72759 4887 4945
7.73591 4945 5003
7.74423 5003 5060
7.75255 5060 5116
7.76087 5116 5172
7.76919 5172 5227
7.77751 5227 5281
7.78583 5281 5335
7.79415 5335 5388
7.80247 5388 5441
7.81079 5441 5493
7.81911 5493 5545
7.82743 5545 5596
7.83575 5596 5647
7.84407 5647 5697
7.85239 5697 5747
7.86071 5747 5796
7.86903 5796 5845
7.87735 5845 5894
7.88567 5894 5942
7.89399 5942 5990
```

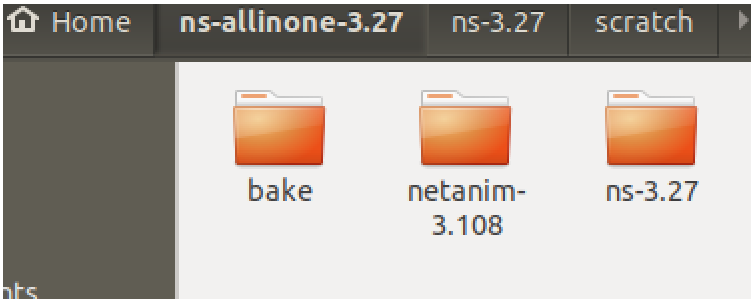
```
7.90231 5990 6037
7.91063 6037 6084
7.91895 6084 6131
7.92727 6131 6177
7.93559 6177 6223
7.94391 6223 6269
7.95223 6269 6314
7.96055 6314 6359
7.96887 6359 6404
7.97719 6404 6448
7.98551 6448 6492
7.99383 6492 6536
8.00215 6536 6579
8.01047 6579 6622
8.01879 6622 6665
8.02711 6665 6708
8.03543 6708 6750
8.04375 6750 6792
8.05207 6792 6834
8.06039 6834 6876
8.06871 6876 6917
8.07703 6917 6958
8.08535 6958 6999
8.09367 6999 7040
8.10199 7040 7080
8.11031 7080 7120
8.11863 7120 7160
8.12695 7160 7200
8.13618 7200 7239
8.14953 7239 7278
8.1528 7278 7317
8.16112 7317 7356
8.16944 7356 7395
8.17776 7395 7433
8.18608 7433 7471
8.1944 7471 7509
8.20272 7509 7547
8.21104 7547 7585
8.21936 7585 7622
8.22768 7622 7659
8.236 7659 7696
8.24432 7696 7733
8.25264 7733 7770
8.26096 7770 7806
```

```
8.26928 7806 7842
8.2776 7842 7878
8.28592 7878 7914
8.29424 7914 7950
8.30256 7950 7986
8.31088 7986 8021
8.3192 8021 8056
8.32752 8056 8091
8.33584 8091 8126
8.34416 8126 8161
8.35248 8161 8196
8.3608 8196 8231
8.36912 8231 8265
8.37744 8265 8299
8.38576 8299 8333
8.39408 8333 8367
8.4024 8367 8401
8.41072 8401 8435
8.41904 8435 8469
8.42736 8469 8502
8.43568 8502 8535
8.444 8535 8568
8.45232 8568 8601
8.46064 8601 8634
8.46896 8634 8667
8.47728 8667 8700
8.4856 8700 8733
8.49392 8733 8765
8.50224 8765 8797
8.51056 8797 8829
8.51888 8829 8861
8.5272 8861 8893
8.53552 8893 8925
8.54384 8925 8957
8.55216 8957 8989
8.56048 8989 9020
8.5688 9020 9051
8.57712 9051 9082
8.58544 9082 9113
8.59376 9113 9144
8.60208 9144 9175
8.6104 9175 9206
8.61872 9206 9237
8.62704 9237 9268
```

```
8.63536 9268 9298
8.64368 9298 9328
8.652 9328 9358
8.66032 9358 9388
8.66864 9388 9418
8.67696 9418 9448
8.68528 9448 9478
8.6936 9478 9508
8.70192 9508 9538
8.71024 9538 9568
8.71856 9568 9598
8.72688 9598 9627
8.7352 9627 9656
8.74352 9656 9685
8.75184 9685 9714
8.76016 9714 9743
8.76848 9743 9772
8.7768 9772 9801
8.78512 9801 9830
8.79344 9830 9859
8.80176 9859 9888
8.81008 9888 9917
8.8184 9917 9945
8.82672 9945 9973
8.83504 9973 10001
8.84336 10001 10029
8.85168 10029 10057
8.86 10057 10085
8.86832 10085 10113
8.87664 10113 10141
8.88496 10141 10169
8.89328 10169 10197
8.9016 10197 10225
8.90992 10225 10253
8.91824 10253 10281
8.92656 10281 10308
8.93488 10308 10335
8.9432 10335 10362
8.95152 10362 10389
8.95984 10389 10416
8.96816 10416 10443
8.97648 10443 10470
8.9848 10470 10497
8.99312 10497 10524
```

```
9.00144 10524 10551
9.00976 10551 10578
9.01808 10578 10605
9.0264 10605 10632
9.03472 10632 10659
9.04304 10659 10685
9.05136 10685 10711
9.05968 10711 10737
9.068 10737 10763
9.07632 10763 10789
9.08464 10789 10815
9.09296 10815 10841
9.10128 10841 10867
9.1096 10867 10893
9.11792 10893 10919
9.12624 10919 10945
9.13456 10945 10971
9.14288 10971 10997
9.1512 10997 11023
9.15952 11023 11049
9.16784 11049 11075
9.17616 11075 11100
9.18448 11100 11125
9.1928 11125 11150
9.20112 11150 11175
9.20944 11175 11200
9.21776 11200 11225
9.22608 11225 11250
9.2344 11250 11275
9.24272 11275 11300
9.25104 11300 11325
9.25936 11325 11350
9.26768 11350 11375
9.276 11375 11400
9.28432 11400 11425
9.29264 11425 11450
9.30096 11450 11475
9.30928 11475 11500
9.3176 11500 11524
```





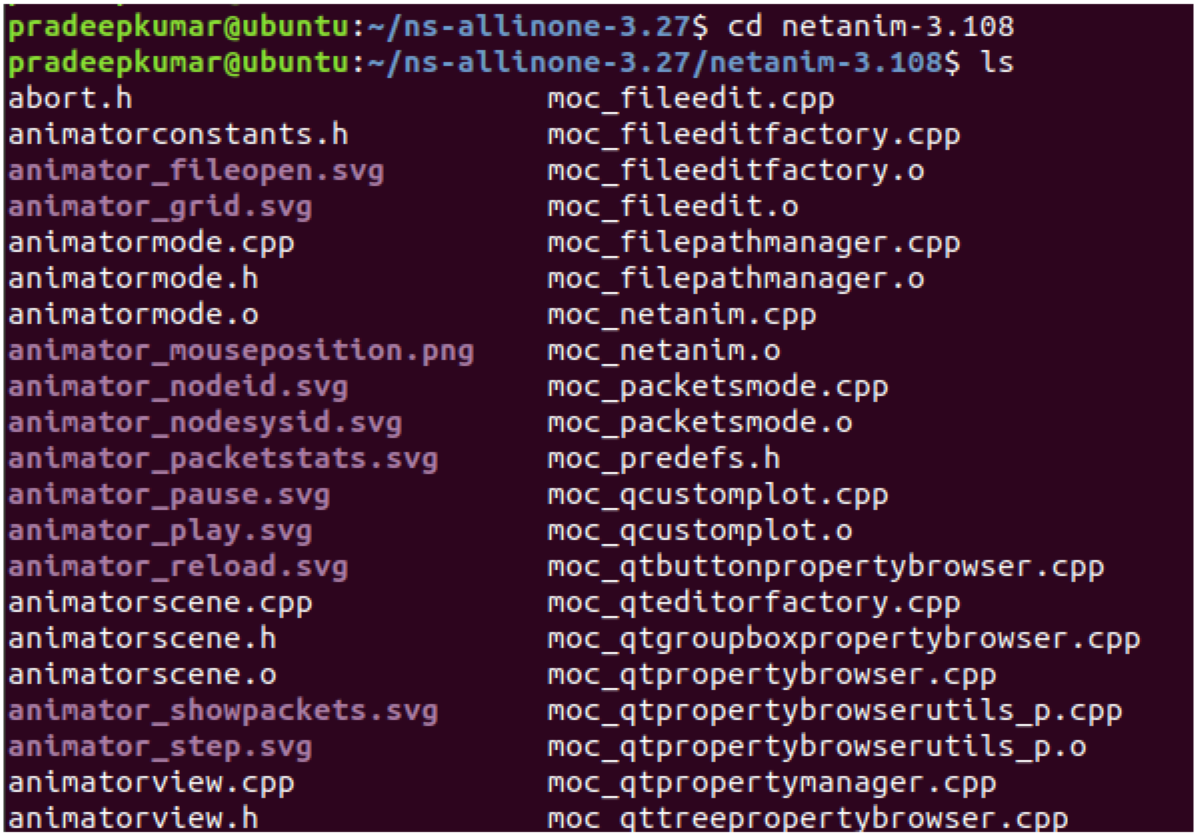
Seventh.cc

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 */

#include <fstream>
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/stats-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("SeventhScriptExample");
```

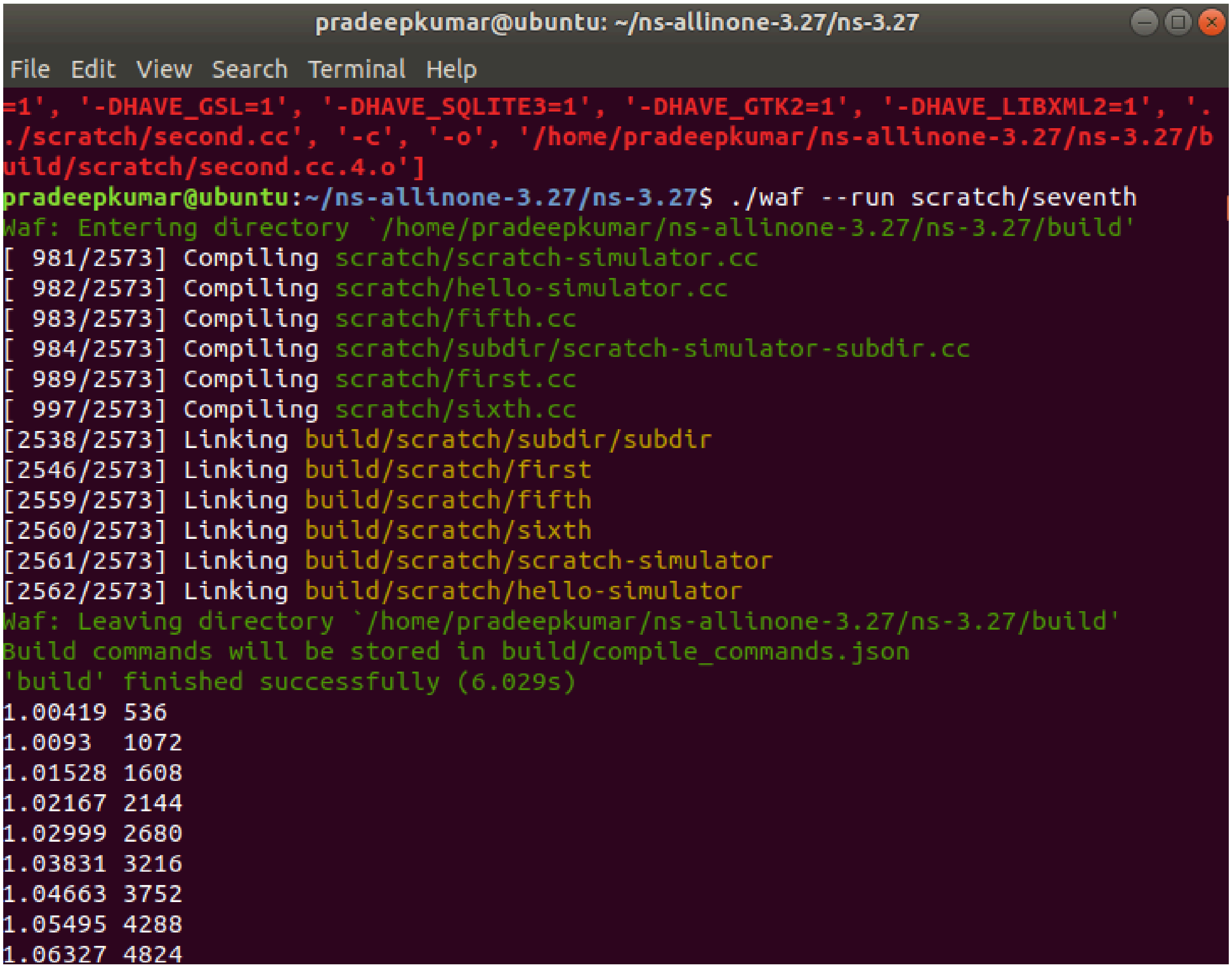
```
//
==========================================================================
===
//
//           node 0                 node 1
//   +----------------+    +----------------+
//   |    ns-3 TCP    |    |    ns-3 TCP    |
//   +----------------+    +----------------+
//   |    10.1.1.1    |    |    10.1.1.2    |
//   +----------------+    +----------------+
//   | point-to-point |    | point-to-point |
//   +----------------+    +----------------+
//           |                     |
//           +---------------------+
//                5 Mbps, 2 ms
//
//
// We want to look at changes in the ns-3 TCP congestion window.  We need
// to crank up a flow and hook the CongestionWindow attribute on the socket
// of the sender.  Normally one would use an on-off application to generate a
// flow, but this has a couple of problems.  First, the socket of the on-off
// application is not created until Application Start time, so we wouldn't be
// able to hook the socket (now) at configuration time.  Second, even if we
// could arrange a call after start time, the socket is not public so we
// couldn't get at it.
//
// So, we can cook up a simple version of the on-off application that does what
// we want.  On the plus side we don't need all of the complexity of the on-off
// application.  On the minus side, we don't have a helper, so we have to get
// a little more involved in the details, but this is trivial.
//
// So first, we create a socket and do the trace connect on it; then we pass
// this socket into the constructor of our simple application which we then
// install in the source node.
//
==========================================================================
===
//
class MyApp : public Application
{
public:
  MyApp ();
  virtual ~MyApp ();
  /**
   * Register this type.
   * \return The TypeId.
   */
  static TypeId GetTypeId (void);
  void Setup (Ptr<Socket> socket, Address address, uint32_t packetSize, uint32_t nPackets,
DataRate dataRate);

private:
  virtual void StartApplication (void);
  virtual void StopApplication (void);

  void ScheduleTx (void);
  void SendPacket (void);

  Ptr<Socket>     m_socket;
  Address         m_peer;
  uint32_t        m_packetSize;
  uint32_t        m_nPackets;
  DataRate        m_dataRate;
  EventId         m_sendEvent;
  bool            m_running;
  uint32_t        m_packetsSent;
};

MyApp::MyApp ()
  : m_socket (0),
    m_peer (),
    m_packetSize (0),
    m_nPackets (0),
    m_dataRate (0),
    m_sendEvent (),
    m_running (false),
    m_packetsSent (0)
{
}

MyApp::~MyApp ()
{
  m_socket = 0;
}

/* static */
```

```
TypeId MyApp::GetTypeId (void)
{
  static TypeId tid = TypeId ("MyApp")
    .SetParent<Application> ()
    .SetGroupName ("Tutorial")
    .AddConstructor<MyApp> ()
    ;
  return tid;
}

void
MyApp::Setup (Ptr<Socket> socket, Address address, uint32_t packetSize, uint32_t nPackets,
DataRate dataRate)
{
  m_socket = socket;
  m_peer = address;
  m_packetSize = packetSize;
  m_nPackets = nPackets;
  m_dataRate = dataRate;
}

void
MyApp::StartApplication (void)
{
  m_running = true;
  m_packetsSent = 0;
  if (InetSocketAddress::IsMatchingType (m_peer))
    {
      m_socket->Bind ();
    }
  else
    {
      m_socket->Bind6 ();
    }
  m_socket->Connect (m_peer);
  SendPacket ();
}

void
MyApp::StopApplication (void)
{
  m_running = false;

  if (m_sendEvent.IsRunning ())
```

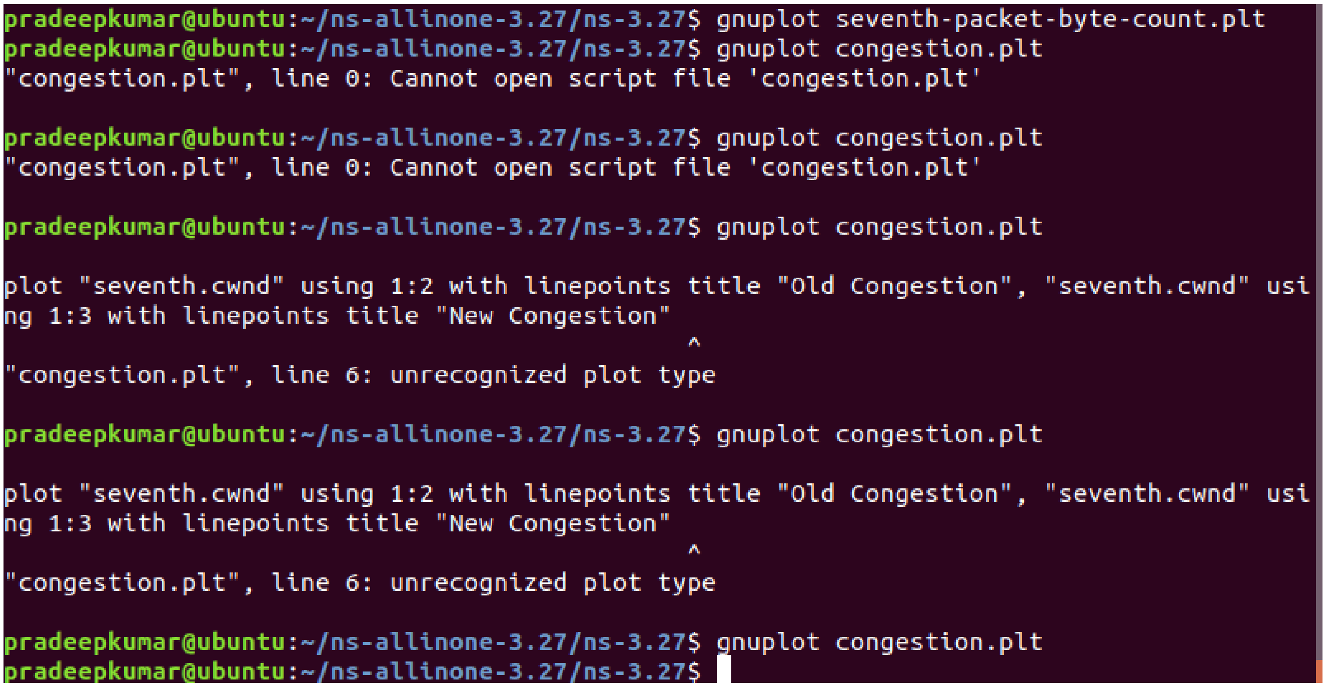
```
    {
      Simulator::Cancel (m_sendEvent);
    }

  if (m_socket)
    {
      m_socket->Close ();
    }
}

void
MyApp::SendPacket (void)
{
  Ptr<Packet> packet = Create<Packet> (m_packetSize);
  m_socket->Send (packet);

  if (++m_packetsSent < m_nPackets)
    {
      ScheduleTx ();
    }
}

void
MyApp::ScheduleTx (void)
{
  if (m_running)
    {
      Time tNext (Seconds (m_packetSize * 8 / static_cast<double> (m_dataRate.GetBitRate ())));
      m_sendEvent = Simulator::Schedule (tNext, &MyApp::SendPacket, this);
    }
}

static void
CwndChange (Ptr<OutputStreamWrapper> stream, uint32_t oldCwnd, uint32_t newCwnd)
{
  NS_LOG_UNCOND (Simulator::Now ().GetSeconds () << "\t" << newCwnd);
  *stream->GetStream () << Simulator::Now ().GetSeconds () << "\t" << oldCwnd << "\t" <<
newCwnd << std::endl;
}

static void
RxDrop (Ptr<PcapFileWrapper> file, Ptr<const Packet> p)
{
  NS_LOG_UNCOND ("RxDrop at " << Simulator::Now ().GetSeconds ());
  file->Write (Simulator::Now (), p);
}

int
main (int argc, char *argv[])
{
  bool useV6 = false;

  CommandLine cmd;
  cmd.AddValue ("useIpv6", "Use Ipv6", useV6);
  cmd.Parse (argc, argv);

  NodeContainer nodes;
  nodes.Create (2);

  PointToPointHelper pointToPoint;
  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

  NetDeviceContainer devices;
  devices = pointToPoint.Install (nodes);

  Ptr<RateErrorModel> em = CreateObject<RateErrorModel> ();
  em->SetAttribute ("ErrorRate", DoubleValue (0.00001));
  devices.Get (1)->SetAttribute ("ReceiveErrorModel", PointerValue (em));

  InternetStackHelper stack;
  stack.Install (nodes);

  uint16_t sinkPort = 8080;
  Address sinkAddress;
  Address anyAddress;
  std::string probeType;
  std::string tracePath;
  if (useV6 == false)
    {
      Ipv4AddressHelper address;
      address.SetBase ("10.1.1.0", "255.255.255.0");
      Ipv4InterfaceContainer interfaces = address.Assign (devices);
      sinkAddress = InetSocketAddress (interfaces.GetAddress (1), sinkPort);
      anyAddress = InetSocketAddress (Ipv4Address::GetAny (), sinkPort);
      probeType = "ns3::Ipv4PacketProbe";
      tracePath = "/NodeList/*/$ns3::Ipv4L3Protocol/Tx";
    }
  else
    {
      Ipv6AddressHelper address;
      address.SetBase ("2001:0000:f00d:cafe::", Ipv6Prefix (64));
      Ipv6InterfaceContainer interfaces = address.Assign (devices);
      sinkAddress = Inet6SocketAddress (interfaces.GetAddress (1,1), sinkPort);
      anyAddress = Inet6SocketAddress (Ipv6Address::GetAny (), sinkPort);
      probeType = "ns3::Ipv6PacketProbe";
      tracePath = "/NodeList/*/$ns3::Ipv6L3Protocol/Tx";
    }

  PacketSinkHelper packetSinkHelper ("ns3::TcpSocketFactory", anyAddress);
  ApplicationContainer sinkApps = packetSinkHelper.Install (nodes.Get (1));
  sinkApps.Start (Seconds (0.));
  sinkApps.Stop (Seconds (20.));

  Ptr<Socket> ns3TcpSocket = Socket::CreateSocket (nodes.Get (0),
TcpSocketFactory::GetTypeId ());

  Ptr<MyApp> app = CreateObject<MyApp> ();
  app->Setup (ns3TcpSocket, sinkAddress, 1040, 1000, DataRate ("1Mbps"));
  nodes.Get (0)->AddApplication (app);
  app->SetStartTime (Seconds (1.));
  app->SetStopTime (Seconds (20.));

  AsciiTraceHelper asciiTraceHelper;
  Ptr<OutputStreamWrapper> stream = asciiTraceHelper.CreateFileStream ("seventh.cwnd");
  ns3TcpSocket->TraceConnectWithoutContext ("CongestionWindow", MakeBoundCallback
(&CwndChange, stream));

  PcapHelper pcapHelper;
  Ptr<PcapFileWrapper> file = pcapHelper.CreateFile ("seventh.pcap", std::ios::out,
PcapHelper::DLT_PPP);
  devices.Get (1)->TraceConnectWithoutContext ("PhyRxDrop", MakeBoundCallback (&RxDrop,
file));

  // Use GnuplotHelper to plot the packet byte count over time
  GnuplotHelper plotHelper;

  // Configure the plot.  The first argument is the file name prefix
  // for the output files generated.  The second, third, and fourth
  // arguments are, respectively, the plot title, x-axis, and y-axis labels
  plotHelper.ConfigurePlot ("seventh-packet-byte-count",
                            "Packet Byte Count vs. Time",
                         "Time (Seconds)",
                         "Packet Byte Count");

  // Specify the probe type, trace source path (in configuration namespace), and
  // probe output trace source ("OutputBytes") to plot.  The fourth argument
  // specifies the name of the data series label on the plot.  The last
  // argument formats the plot by specifying where the key should be placed.
  plotHelper.PlotProbe (probeType,
                        tracePath,
                        "OutputBytes",
                        "Packet Byte Count",
                        GnuplotAggregator::KEY_BELOW);

  // Use FileHelper to write out the packet byte count over time
  FileHelper fileHelper;

  // Configure the file to be written, and the formatting of output data.
  fileHelper.ConfigureFile ("seventh-packet-byte-count",
                            FileAggregator::FORMATTED);

  // Set the labels for this formatted output file.
  fileHelper.Set2dFormat ("Time (Seconds) = %.3e\tPacket Byte Count = %.0f");

  // Specify the probe type, trace source path (in configuration namespace), and
  // probe output trace source ("OutputBytes") to write.
  fileHelper.WriteProbe (probeType,
                         tracePath,
                         "OutputBytes");

  Simulator::Stop (Seconds (20));
  Simulator::Run ();
  Simulator::Destroy ();

  return 0;
}
```



```
pradeepkumar@ubuntu:~/ns-allinone-3.27$ cd netanim-3.108
pradeepkumar@ubuntu:~/ns-allinone-3.27/netanim-3.108$ ls
abort.h                       moc_fileedit.cpp
animatorconstants.h           moc_fileeditfactory.cpp
animator_fileopen.svg         moc_fileeditfactory.o
animator_grid.svg             moc_fileedit.o
animatormode.cpp              moc_filepathmanager.cpp
animatormode.h                moc_filepathmanager.o
animatormode.o                moc_netanim.cpp
animator_mouseposition.png    moc_netanim.o
animator_nodeid.svg           moc_packetsmode.cpp
animator_nodesysid.svg        moc_packetsmode.o
animator_packetstats.svg      moc_predefs.h
animator_pause.svg            moc_qcustomplot.cpp
animator_play.svg             moc_qcustomplot.o
animator_reload.svg           moc_qtbuttonpropertybrowser.cpp
animatorscene.cpp             moc_qteditorfactory.cpp
animatorscene.h               moc_qtgroupboxpropertybrowser.cpp
animatorscene.o               moc_qtpropertybrowser.cpp
animator_showpackets.svg      moc_qtpropertybrowserutils_p.cpp
animator_step.svg             moc_qtpropertybrowserutils_p.o
animatorview.cpp              moc_qtpropertymanager.cpp
animatorview.h                moc_qttreepropertybrowser.cpp
```


pradeepkumar@ubuntu: ~/ns-allinone-3.27/ns-3.27
File Edit View Search Terminal Help

```
=1', '-DHAVE_GSL=1', '-DHAVE_SQLITE3=1', '-DHAVE_GTK2=1', '-DHAVE_LIBXML2=1', '.
./scratch/second.cc', '-c', '-o', '/home/pradeepkumar/ns-allinone-3.27/ns-3.27/b
uild/scratch/second.cc.4.o']
pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/seventh
Waf: Entering directory `/home/pradeepkumar/ns-allinone-3.27/ns-3.27/build'
[ 981/2573] Compiling scratch/scratch-simulator.cc
[ 982/2573] Compiling scratch/hello-simulator.cc
[ 983/2573] Compiling scratch/fifth.cc
[ 984/2573] Compiling scratch/subdir/scratch-simulator-subdir.cc
[ 989/2573] Compiling scratch/first.cc
[ 997/2573] Compiling scratch/sixth.cc
[2538/2573] Linking build/scratch/subdir/subdir
[2546/2573] Linking build/scratch/first
[2559/2573] Linking build/scratch/fifth
[2560/2573] Linking build/scratch/sixth
[2561/2573] Linking build/scratch/scratch-simulator
[2562/2573] Linking build/scratch/hello-simulator
Waf: Leaving directory `/home/pradeepkumar/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.029s)
1.00419 536
1.0093  1072
1.01528 1608
1.02167 2144
1.02999 2680
1.03831 3216
1.04663 3752
1.05495 4288
1.06327 4824
```



```
pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ gnuplot seventh-packet-byte-count.plt
pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ gnuplot congestion.plt
"congestion.plt", line 0: Cannot open script file 'congestion.plt'

pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ gnuplot congestion.plt
"congestion.plt", line 0: Cannot open script file 'congestion.plt'

pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ gnuplot congestion.plt

plot "seventh.cwnd" using 1:2 with linepoints title "Old Congestion", "seventh.cwnd" usi
ng 1:3 with linepoints title "New Congestion"
                                               ^
"congestion.plt", line 6: unrecognized plot type

pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ gnuplot congestion.plt

plot "seventh.cwnd" using 1:2 with linepoints title "Old Congestion", "seventh.cwnd" usi
ng 1:3 with linepoints title "New Congestion"
                                               ^
"congestion.plt", line 6: unrecognized plot type

pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$ gnuplot congestion.plt
pradeepkumar@ubuntu:~/ns-allinone-3.27/ns-3.27$
```