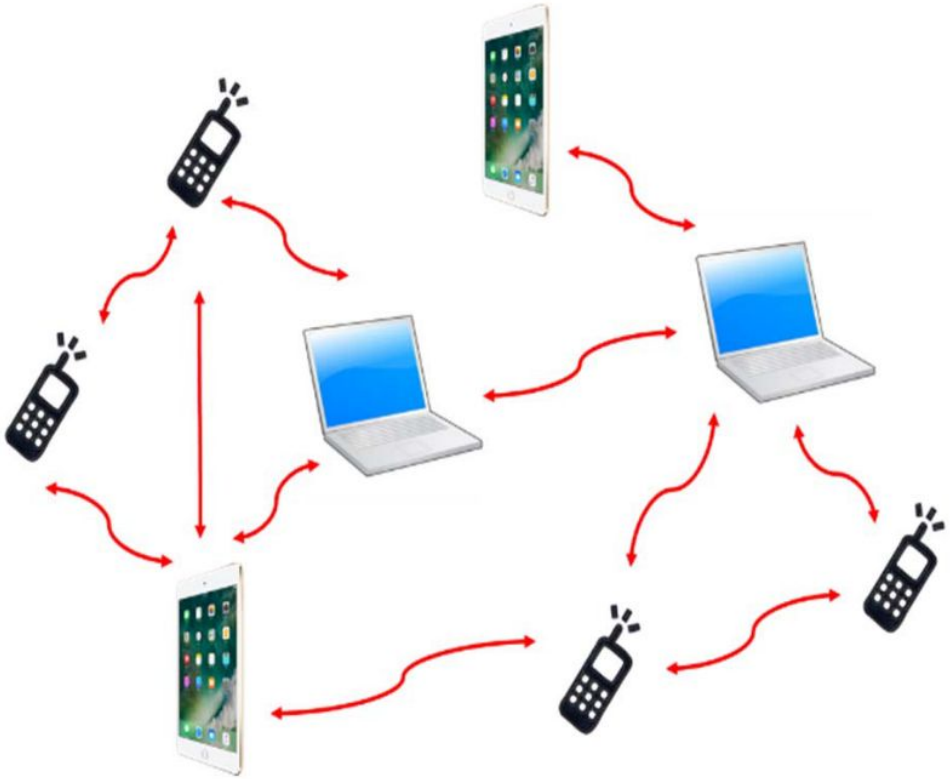
An Effective Multiple Paths Congestion Control AODV



Kawshik Kumar Paul
1705043
Undergrad Student
Dept of CSE, BUET

Conference Paper

CC-ADOV: An effective multiple paths congestion control AODV

January 2018

DOI:10.1109/CCWC.2018.8301758

Conference: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)

Authors:

Yefa Mai
California State University, Fresno

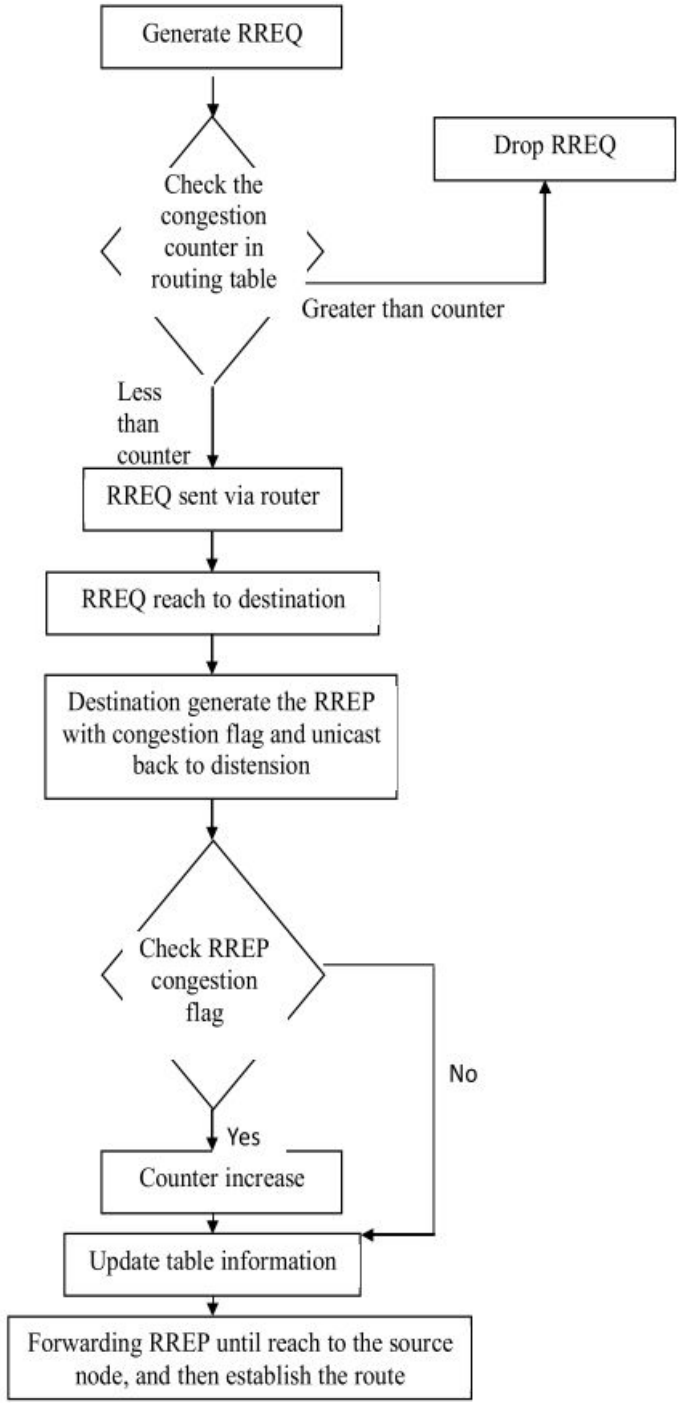
Fernando Molina Rodriguez

Nan Wang

https://www.researchgate.net/publication/323562880_CC-ADOV_An_effective_multiple_paths_congestion_control_AODV

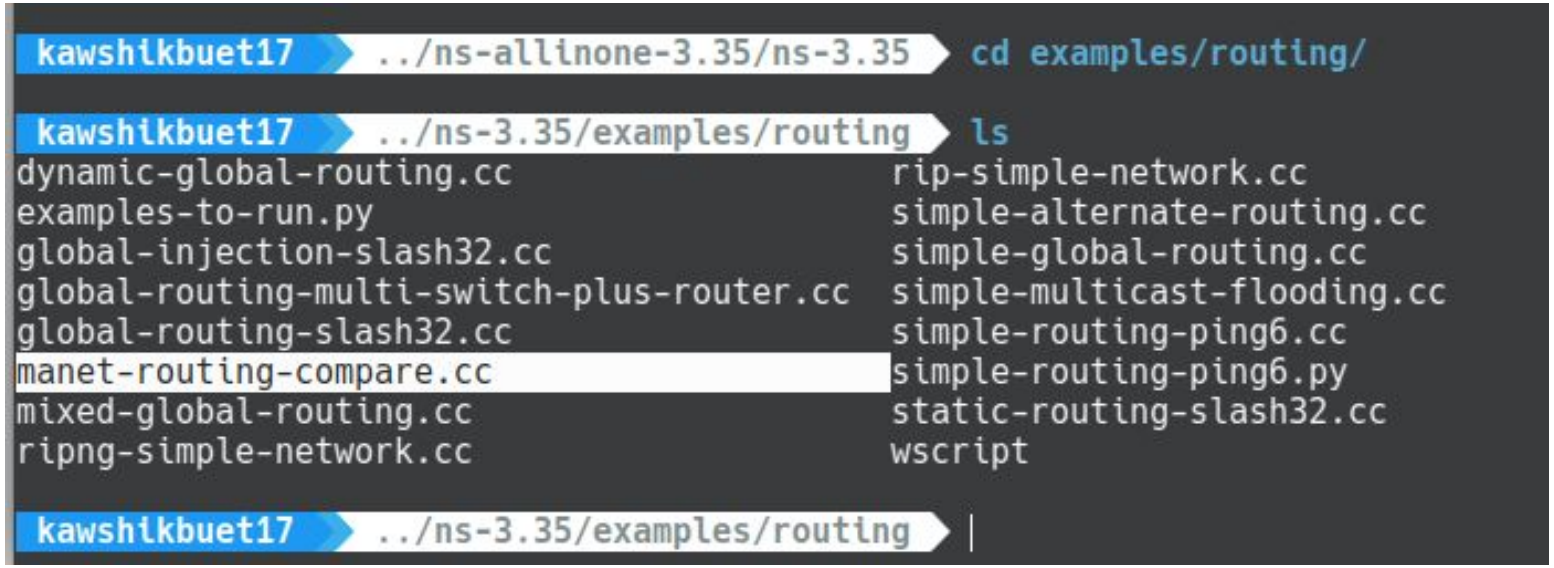
https://ieeexplore.ieee.org/document/8301758

CC-AODV Flowchart



Project Update 2

File Reference



Main Function

scratch > AODV-Topology.cc > main(int, char * [])

```
201  main (int argc, char *argv[])
202  {
203    RoutingExperiment experiment;
204    std::string CSVfileName = experiment.CommandSetup (argc,argv);
205
206    //blank out the last output file and write the column headers
207    std::ofstream out (CSVfileName.c_str ());
208    out << "SimulationSecond," <<
209    "ReceiveRate," <<
210    "PacketsReceived," <<
211    "NumberOfSinks," <<
212    "RoutingProtocol," <<
213    "TransmissionPower" <<
214    std::endl;
215    out.close ();
216
217    int nSinks = 10;
218    double txp = 7.5;
219    experiment.Run (nSinks, txp, CSVfileName);
220  }
```

scratch > AODV-Topology.cc > RoutingExperiment()

```
113
114  RoutingExperiment::RoutingExperiment ()
115    : port (9),
116      bytesTotal (0),
117      packetsReceived (0),
118      m_CSVfileName ("manet-routing-output.csv"),
119      m_traceMobility (false),
120      m_protocol (2) // AODV
121  {
122  }
```

Set Simulation Time and Params

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
222  void
223  RoutingExperiment::Run (int nSinks, double txp, std::string CSVfileName)
224  {
225    Packet::EnablePrinting ();
226    m_nSinks = nSinks;
227    m_txp = txp;
228    m_CSVfileName = CSVfileName;
229
230    int nWifis = 50;
231
232    double TotalTime = 70.0;
233    std::string rate ("2048bps");
234    std::string phyMode ("DsssRate11Mbps");
235    std::string tr_name ("AODV_Topology_Trace");
236    int nodeSpeed = 20; //in m/s
237    int nodePause = 0; //in s
238    m_protocolName = "protocol";
239
```

Creating adhocNodes and adhocDevices

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
240   Config::SetDefault  ("ns3::OnOffApplication::PacketSize",StringValue ("64"));
241   Config::SetDefault ("ns3::OnOffApplication::DataRate",  StringValue (rate));
242
243   //Set Non-unicastMode rate to unicast mode
244   Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",StringValue (phyMode));
245
246   NodeContainer adhocNodes;
247   adhocNodes.Create (nWifis);
248
249   // setting up wifi phy and channel using helpers
250   WifiHelper wifi;
251   wifi.SetStandard (WIFI_STANDARD_80211b);
252
253   YansWifiPhyHelper wifiPhy;
254   YansWifiChannelHelper wifiChannel;
255   wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
256   wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");
257   wifiPhy.SetChannel (wifiChannel.Create ());
258
259   // Add a mac and disable rate control
260   WifiMacHelper wifiMac;
261   wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
262                                 "DataMode",StringValue (phyMode),
263                                 "ControlMode",StringValue (phyMode));
264
265   wifiPhy.Set ("TxPowerStart",DoubleValue (txp));
266   wifiPhy.Set ("TxPowerEnd", DoubleValue (txp));
267
268   wifiMac.SetType ("ns3::AdhocWifiMac");
269   NetDeviceContainer adhocDevices = wifi.Install (wifiPhy, wifiMac, adhocNodes);
```

Set Mobility

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
271   MobilityHelper mobilityAdhoc;
272   int64_t streamIndex = 0; // used to get consistent mobility across scenarios
273
274   ObjectFactory pos;
275   pos.SetTypeId ("ns3::RandomRectanglePositionAllocator");
276   pos.Set ("X", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=300.0]"));
277   pos.Set ("Y", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=1500.0]"));
278
279   Ptr<PositionAllocator> taPositionAlloc = pos.Create ()->GetObject<PositionAllocator> ();
280   streamIndex += taPositionAlloc->AssignStreams (streamIndex);
281
282   std::stringstream ssSpeed;
283   ssSpeed << "ns3::UniformRandomVariable[Min=0.0|Max=" << nodeSpeed << "]";
284   std::stringstream ssPause;
285   ssPause << "ns3::ConstantRandomVariable[Constant=" << nodePause << "]";
286   mobilityAdhoc.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
287                                   "Speed", StringValue (ssSpeed.str ()),
288                                   "Pause", StringValue (ssPause.str ()),
289                                   "PositionAllocator", PointerValue (taPositionAlloc));
290   mobilityAdhoc.SetPositionAllocator (taPositionAlloc);
291   mobilityAdhoc.Install (adhocNodes);
292   streamIndex += mobilityAdhoc.AssignStreams (adhocNodes, streamIndex);
293   NS_UNUSED (streamIndex); // From this point, streamIndex is unused
```

Adding Network Address

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
336
337     Ipv4AddressHelper addressAdhoc;
338     addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");
339     Ipv4InterfaceContainer adhocInterfaces;
340     adhocInterfaces = addressAdhoc.Assign (adhocDevices);
```

Packet Send Receive

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
342   OnOffHelper onoff1 ("ns3::UdpSocketFactory",Address ());
343   onoff1.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"));
344   onoff1.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0.0]"));
345
346   for (int i = 0; i < nSinks; i++)
347     {
348       Ptr<Socket> sink = SetupPacketReceive (adhocInterfaces.GetAddress (i), adhocNodes.Get (i));
349
350       AddressValue remoteAddress (InetSocketAddress (adhocInterfaces.GetAddress (i), port));
351       onoff1.SetAttribute ("Remote", remoteAddress);
352
353       Ptr<UniformRandomVariable> var = CreateObject<UniformRandomVariable> ();
354       ApplicationContainer temp = onoff1.Install (adhocNodes.Get (i + nSinks));
355       temp.Start (Seconds (var->GetValue (50.0,51.0)));
356       temp.Stop (Seconds (TotalTime));
357     }
```

.tr file generating

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
378   AsciiTraceHelper ascii;
379   Ptr<OutputStreamWrapper> osw = ascii.CreateFileStream ( (tr_name + ".tr").c_str());
380   wifiPhy.EnableAsciiAll (osw);
381   // AsciiTraceHelper ascii;
382   // MobilityHelper::EnableAsciiAll (ascii.CreateFileStream (tr_name + ".mob"));
383
```

Flow Monitor

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
383
384     uint32_t rxPacketsum = 0;
385     double Delaysum = 0;
386     double rxTimeSum = 0, txTimeSum = 0;
387     uint32_t txPacketsum = 0;
388     uint32_t txBytessum = 0;
389     uint32_t rxBytessum = 0;
390     uint32_t txTimeFirst = 0;
391     uint32_t rxTimeLast = 0;
392     uint32_t lostPacketssum = 0;
393
394     FlowMonitorHelper flowmon;
395     Ptr<FlowMonitor> monitor = flowmon.InstallAll();
396
```

Run Simulation

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
397
398   NS_LOG_INFO ("Run Simulation.");
399
400   CheckThroughput ();
401
402   Simulator::Stop (Seconds (TotalTime));
403   Simulator::Run ();
```

Calculating Flow Packet Properties

scratch > AODV-Topology.cc > Run(int, double, std::string)

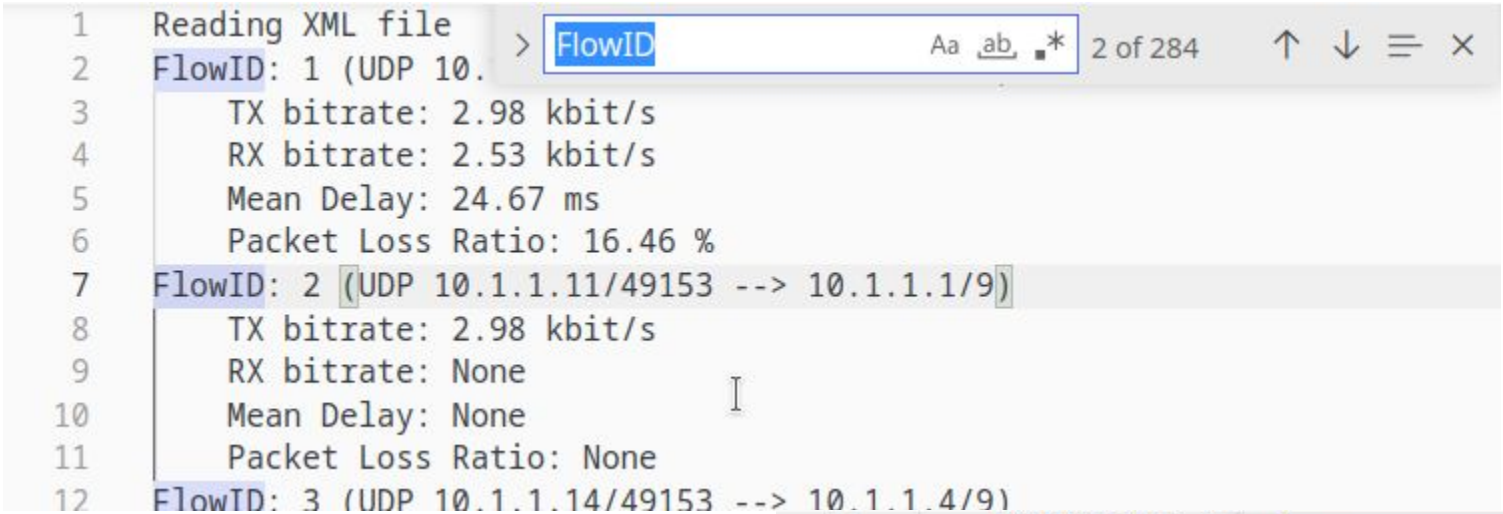
```
405  Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier> (flowmon.GetClassifier ());
406  std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats ();
407
408  for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator i = stats.begin (); i != stats.end (); ++i)
409  {
410    Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (i->first);
411    if(t.sourcePort==654){
412      continue;
413    }
414
415    rxPacketsum += i->second.rxPackets;
416    txPacketsum += i->second.txPackets;
417    txBytessum += i->second.txBytes;
418    rxBytessum += i->second.rxBytes;
419    Delaysum += i->second.delaySum.GetSeconds();
420    lostPacketssum += i->second.lostPackets;
421
422    if(txTimeFirst == 0)
423    {
424      txTimeFirst = i->second.timeFirstTxPacket.GetSeconds();
425    }
426
427    rxTimeLast = i->second.timeLastRxPacket.GetSeconds();
428    lostPacketssum += i->second.lostPackets;
429    Delaysum += i->second.delaySum.GetSeconds();
430  }
431
432  monitor->SerializeToXmlFile ((tr_name + ".flowmon").c_str(), false, false);
```

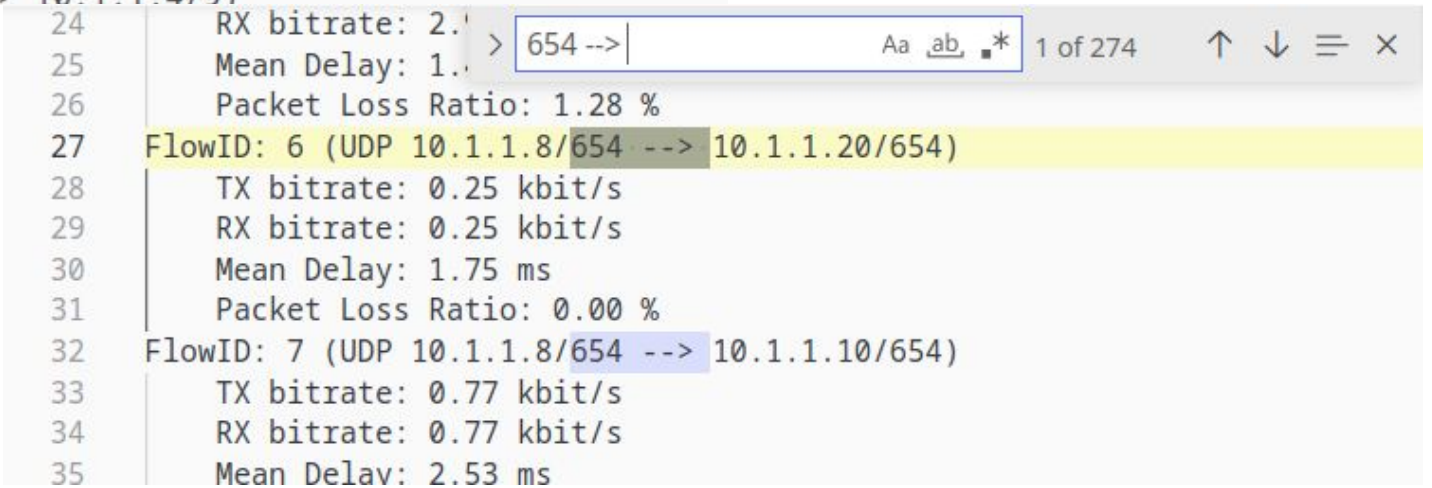
Getting Output and Store in a File

scratch > AODV-Topology.cc > Run(int, double, std::string)

```
434   uint64_t timeDiff = (rxTimeLast - txTimeFirst);
435   double timeDiff2 = (rxTimeSum - txTimeSum) / rxPacketsum;
436
437   std::cout << "\n\n";
438   std::cout << "Total Tx Packets: " << txPacketsum << "\n";
439   std::cout << "Total Rx Packets: " << rxPacketsum << "\n";
440   std::cout << "Total Packets Lost: " << (txPacketsum - rxPacketsum) << "\n";
441   std::cout << "Average Round trip time of Packet: " << timeDiff2 << "\n";
442   std::cout << "Throughput: " << ((rxBytessum * 8.0) / timeDiff)/1024<<" Kbps"<<"\n";
443   std::cout << "Packets Loss Ratio: " << (((txPacketsum - rxPacketsum) * 100) /txPacketsum) << "%" << "\n";
444   std::cout << "Packets Delivery Ratio: " << ((rxPacketsum * 100) /txPacketsum) << "%" << "\n";
445   std::cout << "Avg End to End Delay: " << Delaysum/rxPacketsum << "\n";
446
447   std::ofstream myfile;
448   myfile.open ("DATA_NSINKS.txt", std::ios::app);
449   myfile<<nSinks<<" "<<txPacketsum<<" "<<rxPacketsum<<" "<<(txPacketsum - rxPacketsum)<<" "<<((rxBytessum * 8.0) / timeDiff)/1024<<" "<<(
      ((txPacketsum - rxPacketsum) * 100) /txPacketsum)<<" "<<((rxPacketsum * 100) /txPacketsum)<<""<<Delaysum<<std::endl;
450   myfile.close();
451
452   Simulator::Destroy ();
453 }
```

Flow (For nSinks=10)





Running Code (nSinks=1)

scratch > AODV-Topology.cc > main(int, char * [])

```
217   int nSinks = 1;
218   double txp = 7.5;
219   experiment.Run (nSinks, txp, CSVfileName);
220 }
221
222 void
223 RoutingExperiment::Run (int nSinks, double txp, std::string CSVfileName)
224 {
225   Packet::EnablePrinting ();
226   m_nSinks = nSinks;
227   m_txp = txp;
228   m_CSVfileName = CSVfileName;
229
230   int nWifis = 50;
231
232   double TotalTime = 70.0;
233   std::string rate ("2048bps");
234   std::string phyMode ("DsssRate11Mbps");
235   std::string tr_name ("AODV_Topology_Trace");
236   int nodeSpeed = 20; //in m/s
```

ns-3.35 : bash — Konsole

File Edit View Bookmarks Settings Help

```
kawshikbuet17 ../ns-allinone-3.35/ns-3.35 ./waf --run scratch/AODV-Topology
Waf: Entering directory `/home/kawshikbuet17/Documents/Coding/L3-T2/CSE-322-Computer-Net
ional/ns-3/ns-download/ns-allinone-3.35/ns-3.35/build'
[2946/2996] Compiling scratch/AODV-Topology.cc
[2957/2996] Linking build/scratch/AODV-Topology
Waf: Leaving directory `/home/kawshikbuet17/Documents/Coding/L3-T2/CSE-322-Computer-Netw
onal/ns-3/ns-download/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.061s)
50.5389 0 received one packet from 10.1.1.2
50.7799 0 received one packet from 10.1.1.2
51.0299 0 received one packet from 10.1.1.2
51.2799 0 received one packet from 10.1.1.2
51.5299 0 received one packet from 10.1.1.2
51.7799 0 received one packet from 10.1.1.2
52.0299 0 received one packet from 10.1.1.2
52.2799 0 received one packet from 10.1.1.2
52.5299 0 received one packet from 10.1.1.2
52.7799 0 received one packet from 10.1.1.2
53.0299 0 received one packet from 10.1.1.2
53.2799 0 received one packet from 10.1.1.2
```

Demo Output for nSinks=1

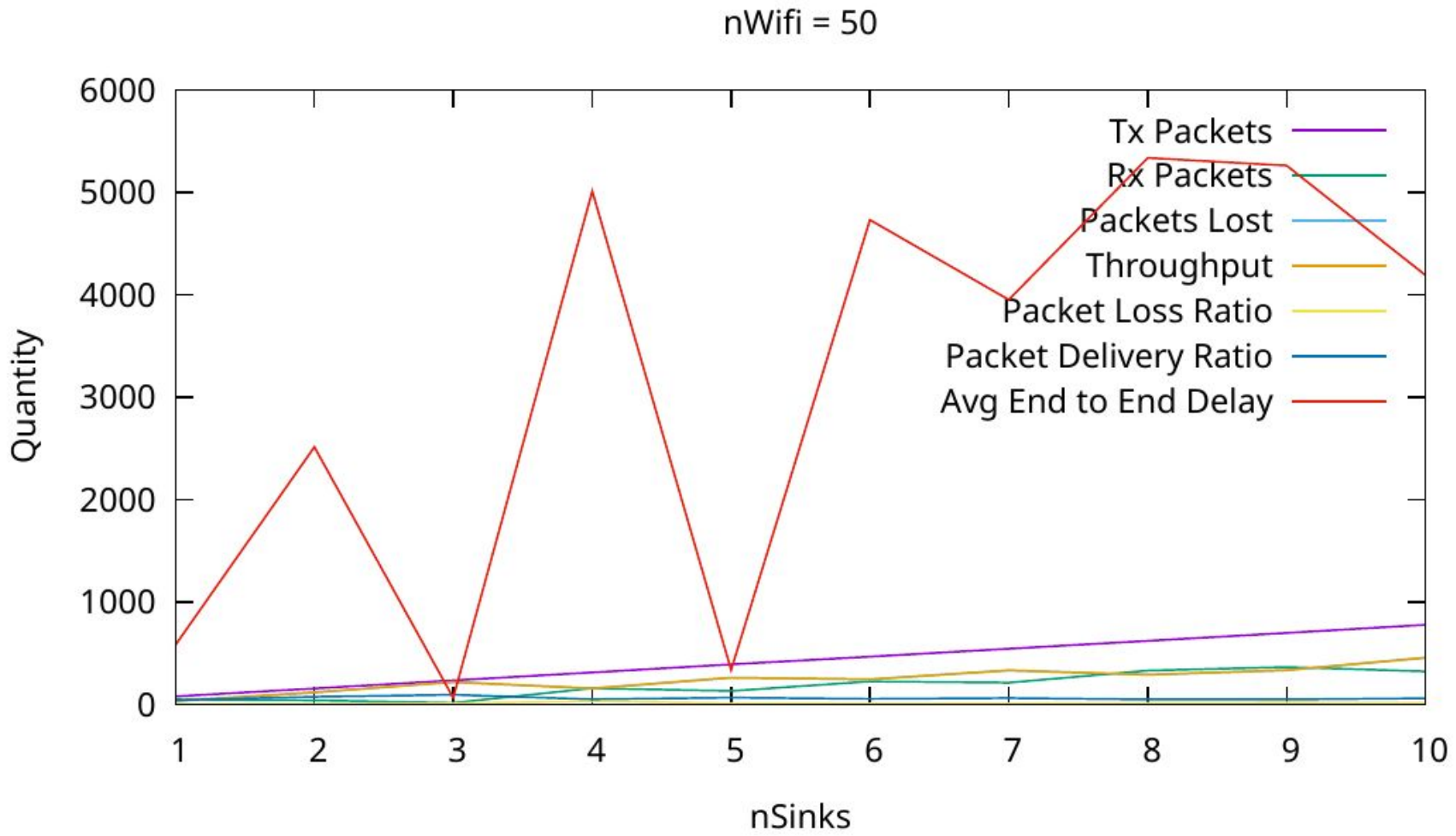
```
59.7799 0 received one packet from 10.1.1.2
60.0299 0 received one packet from 10.1.1.2
60.2799 0 received one packet from 10.1.1.2
60.5299 0 received one packet from 10.1.1.2
60.7799 0 received one packet from 10.1.1.2
61.0299 0 received one packet from 10.1.1.2
61.2799 0 received one packet from 10.1.1.2
61.5299 0 received one packet from 10.1.1.2
61.7799 0 received one packet from 10.1.1.2


Total Tx Packets: 78
Total Rx Packets: 46
Total Packets Lost: 32
Average Round trip time of Packet: 0
Throughput: 3.00568 Kbps
Packets Loss Ratio: 41%
Packets Delivery Ratio: 58%
Avg End to End Delay: 0.00106457

kawshikbuet17  ../ns-allinone-3.35/ns-3.35  |
```

Generate Data and Plot

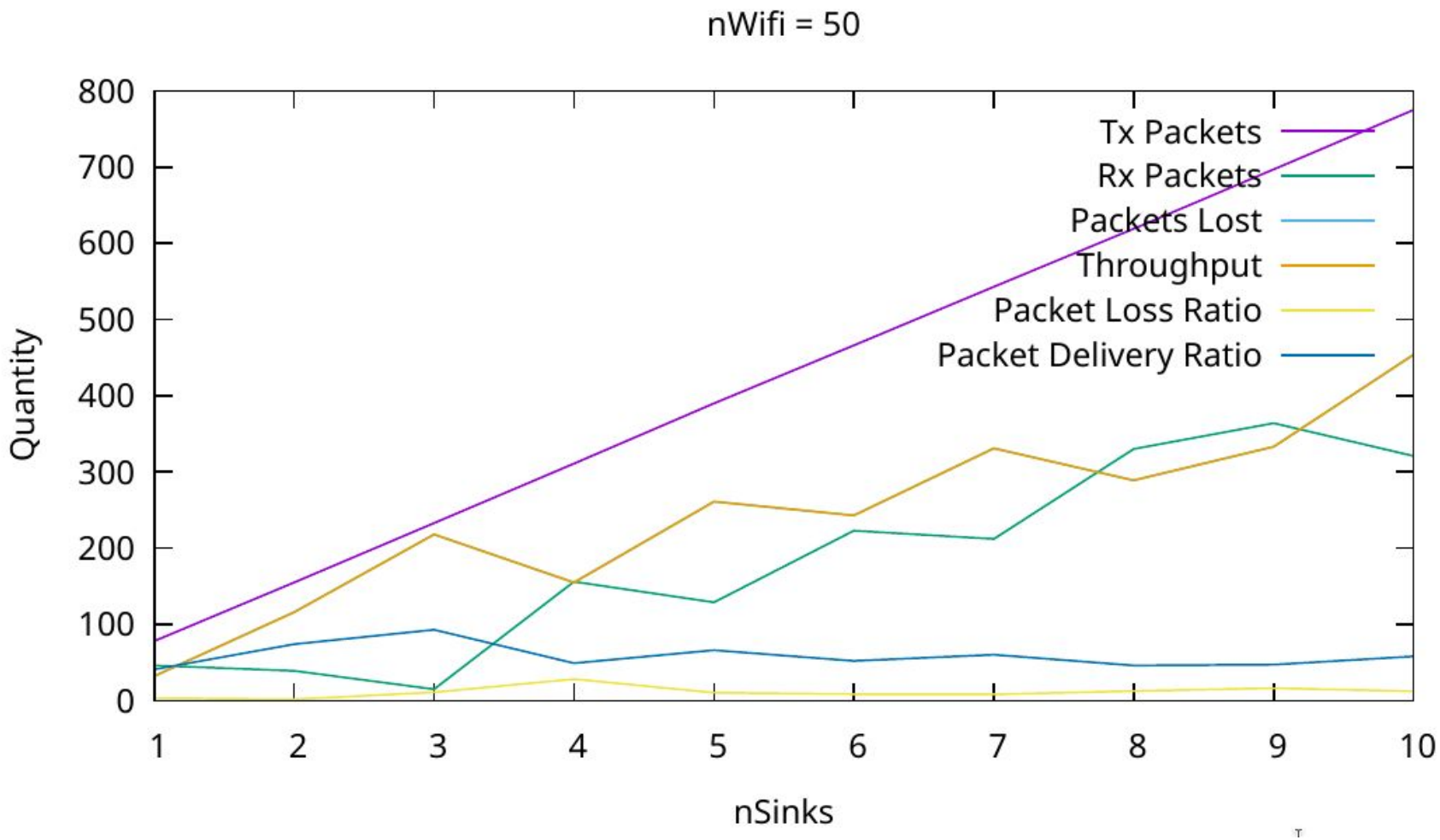
```
kawshikbuet17  ../ns-allinone-3.35/ns-3.35  cat DATA_NSINKS.txt
1 78 46 32 3.00568 41 580.0489703
2 155 39 116 1.75195 74 2511.7714
3 233 15 218 10.7812 93 63.06061
4 311 156 155 28.0312 49 5011.7358
5 390 129 261 10.3021 66 334.67104
7 543 212 331 8.01974 60 3949.7577
6 466 223 243 8.43586 52 4730.7937
8 619 330 289 12.4836 46 5336.3943
9 697 364 333 16.3516 47 5260.6265
10 775 321 454 12.1431 58 4191.0462

kawshikbuet17  ../ns-allinone-3.35/ns-3.35
```

```
kawshikbuet17  ../ns-allinone-3.35/ns-3.35  cat gnucode.plt
set terminal pdf
set output "aodv_report.pdf"
set title "nWifi = 50"
set xlabel "nSinks"
set ylabel "Quantity"
plot "DATA_NSINKS.txt" using 1:2 with lines title "Tx Packets", "DATA_NSINKS.txt" using 1:3 with l
ines title "Rx Packets", "DATA_NSINKS.txt" using 1:4 with lines title "Packets Lost",  "DATA_NSINK
S.txt" using 1:4 with lines title "Throughput", "DATA_NSINKS.txt" using 1:5 with lines title "Pack
et Loss Ratio", "DATA_NSINKS.txt" using 1:6 with lines title "Packet Delivery Ratio", "DATA_NSINKS
.txt" using 1:7 with lines title "Avg End to End Delay"
kawshikbuet17  ../ns-allinone-3.35/ns-3.35
```

Plot Graph 1


nWifi = 50
Quantity
nSinks
0
1000
2000
3000
4000
5000
6000
1
2
3
4
5
6
7
8
9
10
Tx Packets
Rx Packets
Packets Lost
Throughput
Packet Loss Ratio
Packet Delivery Ratio
Avg End to End Delay

Plot Graph 2 (Without E2E Delay)



Thank You