

Name – Pranav Tiwari  
Batch – Cse  
Roll no. - 201127

## Network Laboratory

### Programming Assignment-5

#### Ques:

##### TCP variants

---

- **Level:** Introductory
- **Expected learning outcome:** TCP internals and the difference between each of the variants. NS-3 tracing mechanism.
- **Experiment:**
  1. Create a simple dumbbell topology, two client Node1 and Node2 on the left side of the dumbbell and server nodes Node3 and Node4 on the right side of the dumbbell. Let Node5 and Node6 form the bridge of the dumbbell. Use point to point links.
  2. Install a TCP socket instance on Node1 that will connect to Node3.
  3. Install a UDP socket instance on Node2 that will connect to Node4.
  4. Start the TCP application at time 1s.
  5. Start the UDP application at time 20s at rate Rate1 such that it clogs half the dumbbell bridge's link capacity.
  6. Increase the UDP application's rate at time 30s to rate Rate2 such that it clogs the whole of the dumbbell bridge's capacity.
  7. Use the ns-3 tracing mechanism to record changes in congestion window size of the TCP instance over time. Use gnuplot/matplotlib to visualise plots of cwnd vs time.
  8. Mark points of fast recovery and slow start in the graphs.
  9. Perform the above experiment for TCP variants Tahoe, Reno and New Reno, all of which are available with ns-3.

#### Source Code

q2.cc

```
#include <fstream>
#include <string>
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/internet-module.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/ipv4-global-routing-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("TCP_router_queues");

class MyApp : public Application
{
public:

    MyApp ();
    virtual ~MyApp();

    void Setup (Ptr<Socket> socket, Address address, uint32_t packetSize, uint32_t nPackets,
    DataRate dataRate);
    void ChangeRate(DataRate newrate);

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;  
}
```

```
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;  
  m_socket->Bind ();  
  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);
```

```
}
```

```
}
```

```
void
```

```
MyApp::ChangeRate(DataRate newrate)
```

```
{
```

```
m_dataRate = newrate;
```

```
return;
```

```
}
```

```
static void
```

```
CwndChange (uint32_t oldCwnd, uint32_t newCwnd)
```

```
{
```

```
std::cout << Simulator::Now ().GetSeconds () << "\t" << newCwnd << "\n";  
}
```

```
void  
IncRate (Ptr<MyApp> app, DataRate rate)  
{  
app->ChangeRate(rate);  
return;  
}
```

```
int main (int argc, char *argv[])  
{  
std::string lat = "2ms";  
std::string rate = "500kb/s";  
bool enableFlowMonitor = false;
```

```
CommandLine cmd;  
cmd.AddValue ("latency", "P2P link Latency in milliseconds", lat);  
cmd.AddValue ("rate", "P2P data rate in bps", rate);  
cmd.AddValue ("EnableMonitor", "Enable Flow Monitor", enableFlowMonitor);  
  
cmd.Parse (argc, argv);
```

```
NS_LOG_INFO ("Create nodes.");  
NodeContainer c;  
c.Create(6);
```

```
NodeContainer n0n4 = NodeContainer (c.Get (0), c.Get (4));  
NodeContainer n1n4 = NodeContainer (c.Get (1), c.Get (4));  
NodeContainer n2n5 = NodeContainer (c.Get (2), c.Get (5));  
NodeContainer n3n5 = NodeContainer (c.Get (3), c.Get (5));  
NodeContainer n4n5 = NodeContainer (c.Get (4), c.Get (5));
```

```
InternetStackHelper internet;  
internet.Install (c);
```

```
NS_LOG_INFO ("Create channels.");  
PointToPointHelper p2p;  
p2p.SetDeviceAttribute ("DataRate", StringValue (rate));  
p2p.SetChannelAttribute ("Delay", StringValue (lat));  
NetDeviceContainer d0d4 = p2p.Install (n0n4);  
NetDeviceContainer d1d4 = p2p.Install (n1n4);  
NetDeviceContainer d4d5 = p2p.Install (n4n5);  
NetDeviceContainer d2d5 = p2p.Install (n2n5);  
NetDeviceContainer d3d5 = p2p.Install (n3n5);
```

```
NS_LOG_INFO ("Assign IP Addresses.");
```

```
Ipv4AddressHelper ipv4;  
ipv4.SetBase ("10.10.1.0", "255.255.255.0");  
Ipv4InterfaceContainer i0i4 = ipv4.Assign (d0d4);
```

```
ipv4.SetBase ("10.10.2.0", "255.255.255.0");  
Ipv4InterfaceContainer i1i4 = ipv4.Assign (d1d4);
```

```
ipv4.SetBase ("10.10.3.0", "255.255.255.0");  
Ipv4InterfaceContainer i4i5 = ipv4.Assign (d4d5);
```

```
ipv4.SetBase ("10.10.4.0", "255.255.255.0");  
Ipv4InterfaceContainer i2i5 = ipv4.Assign (d2d5);
```

```
ipv4.SetBase ("10.10.5.0", "255.255.255.0");  
Ipv4InterfaceContainer i3i5 = ipv4.Assign (d3d5);
```

```
NS_LOG_INFO ("Enable static global routing.");  
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

```
NS_LOG_INFO ("Create Applications.");
```

```
uint16_t sinkPort = 8080;  
Address sinkAddress (InetSocketAddress (i2i5.GetAddress (0), sinkPort));  
PacketSinkHelper packetSinkHelper ("ns3::TcpSocketFactory", InetSocketAddress  
(Ipv4Address::GetAny (), sinkPort));  
ApplicationContainer sinkApps = packetSinkHelper.Install (c.Get (2));  
sinkApps.Start (Seconds (0.));  
sinkApps.Stop (Seconds (100.));
```

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

```
ns3TcpSocket->TraceConnectWithoutContext ("CongestionWindow", MakeCallback  
(&CwndChange));
```

```
Ptr<MyApp> app = CreateObject<MyApp> ();  
app->Setup (ns3TcpSocket, sinkAddress, 1040, 100000, DataRate ("250Kbps"));  
c.Get (0)->AddApplication (app);  
app->SetStartTime (Seconds (1.));  
app->SetStopTime (Seconds (100.));
```

```
uint16_t sinkPort2 = 6;  
Address sinkAddress2 (InetSocketAddress (i3i5.GetAddress (0), sinkPort2));  
PacketSinkHelper packetSinkHelper2 ("ns3::UdpSocketFactory", InetSocketAddress  
(Ipv4Address::GetAny (), sinkPort2));  
ApplicationContainer sinkApps2 = packetSinkHelper2.Install (c.Get (3)); //n3 as sink  
sinkApps2.Start (Seconds (0.));  
sinkApps2.Stop (Seconds (100.));
```

```

Ptr<Socket> ns3UdpSocket = Socket::CreateSocket (c.Get (1),
UdpSocketFactory::GetTypeId ());

Ptr<MyApp> app2 = CreateObject<MyApp> ();
app2->Setup (ns3UdpSocket, sinkAddress2, 1040, 100000, DataRate ("250Kbps"));
c.Get (1)->AddApplication (app2);
app2->SetStartTime (Seconds (20.));
app2->SetStopTime (Seconds (100.));

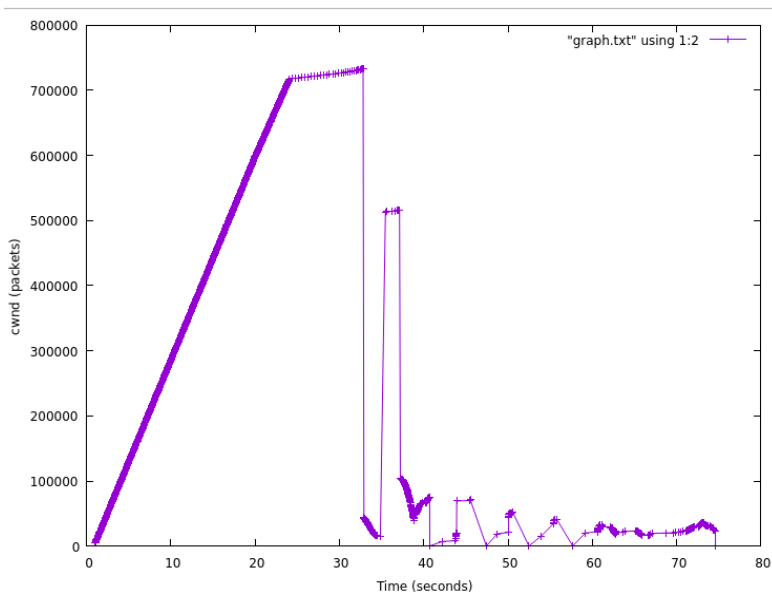
Simulator::Schedule (Seconds(30.0), &IncRate, app2, DataRate("500kbps"));

Ptr<FlowMonitor> flowmon;
if (enableFlowMonitor)
{
FlowMonitorHelper flowmonHelper;
flowmon = flowmonHelper.InstallAll ();
}

NS_LOG_INFO ("Run Simulation.");
Simulator::Stop (Seconds(100.0));
Simulator::Run ();
if (enableFlowMonitor)
{
flowmon->CheckForLostPackets ();
flowmon->SerializeToXmlFile("lab-2.flowmon", true, true);
}
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
}

```

### Gnu Plot:



## Terminal:

```
emperor-kautilya@pop-os:~/Programs/ns-allinone-3.37/ns-3.37$ ./ns3 run
scratch/q2.cc
[  0%] Building CXX object scratch/CMakeFiles/scratch_q2.dir/q2.cc.o
[ 100%] Linking CXX executable ../../build/scratch/ns3.37-q2-default
1.01757 5360
1.06134 5896
1.07971 6432
1.10947 7504
1.14275 8576
1.17603 9648
1.20931 10720
1.24259 11792
1.27587 12864
1.30915 13936
1.34243 15008
1.37571 16080
1.40899 17152
1.44227 18224
1.47555 19296
1.50883 20368
1.54211 21440
1.57539 22512
1.60867 23048
1.64195 24120
1.67523 25192
1.70851 26264
1.74179 27336
1.77507 28408
1.80835 29480
1.84163 30552
1.87491 31624
1.90819 32696
1.94147 33768
1.97475 34840
2.00803 35912
2.04131 36984
2.07459 38056
2.10787 39128
2.14115 40200
2.17443 40736
2.20771 41808
2.24099 42880
2.27427 43952
2.30755 45024
2.34083 46096
2.37411 47168
2.40739 48240
2.44067 49312
2.47395 50384
2.50723 51456
2.54051 52528
2.57379 53600
2.60707 54672
2.64035 55744
2.67363 56816
2.70691 57888
2.74019 58424
2.77347 59496
2.80675 60568
2.84003 61640
2.87331 62712
```

2.90659 63784  
2.93987 64856  
2.97315 65928  
3.00643 67000  
3.03971 68072  
3.07299 69144  
3.10627 70216  
3.13955 71288  
3.17283 72360  
3.20611 73432  
3.23939 74504  
3.27267 75576  
3.30595 76112  
3.33923 77184  
3.37251 78256  
3.40579 79328  
3.43907 80400  
3.47235 81472  
3.50563 82544  
3.53891 83616  
3.57219 84688  
3.60547 85760  
3.63875 86832  
3.67203 87904  
3.70531 88976  
3.73859 90048  
3.77187 91120  
3.80515 92192  
3.83843 92728  
3.87171 93800  
3.90499 94872  
3.93827 95944  
3.97155 97016  
4.00483 98088  
4.03811 99160  
4.07139 100232  
4.10467 101304  
4.13795 102376  
4.17123 103448  
4.20451 104520  
4.23779 105592  
4.27107 106664  
4.30435 107736  
4.33763 108808  
4.37091 109880  
4.40419 110416  
4.43747 111488  
4.47075 112560  
4.50403 113632  
4.53731 114704  
4.57059 115776  
4.60387 116848  
4.63715 117920  
4.67043 118992  
4.70371 120064  
4.73699 121136  
4.77027 122208  
4.80355 123280  
4.83683 124352  
4.87011 125424  
4.90339 126496  
4.93667 127568  
4.96995 128104  
5.00323 129176



5.03651 130248  
5.06979 131320  
5.10307 132392  
5.13635 133464  
5.16963 134536  
5.20291 135608  
5.23619 136680  
5.26947 137752  
5.30275 138824  
5.33603 139896  
5.36931 140968  
5.40259 142040  
5.43587 143112  
5.46915 144184  
5.50243 145256  
5.53571 145792  
5.56899 146864  
5.60227 147936  
5.63555 149008  
5.66883 150080  
5.70211 151152  
5.73539 152224  
5.76867 153296  
5.80195 154368  
5.83523 155440  
5.86851 156512  
5.90179 157584  
5.93507 158656  
5.96835 159728  
6.00163 160800  
6.03491 161872  
6.06819 162408  
6.10147 163480  
6.13475 164552  
6.16803 165624  
6.20131 166696  
6.23459 167768  
6.26787 168840  
6.30115 169912  
6.33443 170984  
6.36771 172056  
6.40099 173128  
6.43427 174200  
6.46755 175272  
6.50083 176344  
6.53411 177416  
6.56739 178488  
6.60067 179560  
6.63395 180096  
6.66723 181168  
6.70051 182240  
6.73379 183312  
6.76707 184384  
6.80035 185456  
6.83363 186528  
6.86691 187600  
6.90019 188672  
6.93347 189744  
6.96675 190816  
7.00003 191888  
7.03331 192960  
7.06659 194032  
7.09987 195104  
7.13315 196176

7.16643 197248  
7.19971 197784  
7.23299 198856  
7.26627 199928  
7.29955 201000  
7.33283 202072  
7.36611 203144  
7.39939 204216  
7.43267 205288  
7.46595 206360  
7.49923 207432  
7.53251 208504  
7.56579 209576  
7.59907 210648  
7.63235 211720  
7.66563 212792  
7.69891 213864  
7.73219 214936  
7.76547 215472  
7.79875 216544  
7.83203 217616  
7.86531 218688  
7.89859 219760  
7.93187 220832  
7.96515 221904  
7.99843 222976  
8.03171 224048  
8.06499 225120  
8.09827 226192  
8.13155 227264  
8.16483 228336  
8.19811 229408  
8.23139 230480  
8.26467 231552  
8.29795 232088  
8.33123 233160  
8.36451 234232  
8.39779 235304  
8.43107 236376  
8.46435 237448  
8.49763 238520  
8.53091 239592  
8.56419 240664  
8.59747 241736  
8.63075 242808  
8.66403 243880  
8.69731 244952  
8.73059 246024  
8.76387 247096  
8.79715 248168  
8.83043 249240  
8.86371 249776  
8.89699 250848  
8.93027 251920  
8.96355 252992  
8.99683 254064  
9.03011 255136  
9.06339 256208  
9.09667 257280  
9.12995 258352  
9.16323 259424  
9.19651 260496  
9.22979 261568  
9.26307 262640

9.29635	263712
9.32963	264784
9.36291	265856
9.39619	266928
9.42947	267464
9.46275	268536
9.49603	269608
9.52931	270680
9.56259	271752
9.59587	272824
9.62915	273896
9.66243	274968
9.69571	276040
9.72899	277112
9.76227	278184
9.79555	279256
9.82883	280328
9.86211	281400
9.89539	282472
9.92867	283544
9.96195	284616
9.99523	285152
10.0285	286224
10.0618	287296
10.0951	288368
10.1284	289440
10.1616	290512
10.1949	291584
10.2282	292656
10.2615	293728
10.2948	294800
10.328	295872
10.3613	296944
10.3946	298016
10.4279	299088
10.4612	300160
10.4944	301232
10.5277	301768
10.561	302840
10.5943	303912
10.6276	304984
10.6608	306056
10.6941	307128
10.7274	308200
10.7607	309272
10.794	310344
10.8272	311416
10.8605	312488
10.8938	313560
10.9271	314632
10.9604	315704
10.9936	316776
11.0269	317848
11.0602	318920
11.0935	319456
11.1268	320528
11.16	321600
11.1933	322672
11.2266	323744
11.2599	324816
11.2932	325888
11.3264	326960
11.3597	328032
11.393	329104

11.4263 330176  
11.4596 331248  
11.4928 332320  
11.5261 333392  
11.5594 334464  
11.5927 335536  
11.626 336608  
11.6592 337144  
11.6925 338216  
11.7258 339288  
11.7591 340360  
11.7924 341432  
11.8256 342504  
11.8589 343576  
11.8922 344648  
11.9255 345720  
11.9588 346792  
11.992 347864  
12.0253 348936  
12.0586 350008  
12.0919 351080  
12.1252 352152  
12.1584 353224  
12.1917 354296  
12.225 354832  
12.2583 355904  
12.2916 356976  
12.3248 358048  
12.3581 359120  
12.3914 360192  
12.4247 361264  
12.458 362336  
12.4912 363408  
12.5245 364480  
12.5578 365552  
12.5911 366624  
12.6244 367696  
12.6576 368768  
12.6909 369840  
12.7242 370912  
12.7575 371448  
12.7908 372520  
12.824 373592  
12.8573 374664  
12.8906 375736  
12.9239 376808  
12.9572 377880  
12.9904 378952  
13.0237 380024  
13.057 381096  
13.0903 382168  
13.1236 383240  
13.1568 384312  
13.1901 385384  
13.2234 386456  
13.2567 387528  
13.29 388600  
13.3232 389136  
13.3565 390208  
13.3898 391280  
13.4231 392352  
13.4564 393424  
13.4896 394496  
13.5229 395568

13.5562 396640  
13.5895 397712  
13.6228 398784  
13.656 399856  
13.6893 400928  
13.7226 402000  
13.7559 403072  
13.7892 404144  
13.8224 405216  
13.8557 406288  
13.889 406824  
13.9223 407896  
13.9556 408968  
13.9888 410040  
14.0221 411112  
14.0554 412184  
14.0887 413256  
14.122 414328  
14.1552 415400  
14.1885 416472  
14.2218 417544  
14.2551 418616  
14.2884 419688  
14.3216 420760  
14.3549 421832  
14.3882 422904  
14.4215 423976  
14.4548 424512  
14.488 425584  
14.5213 426656  
14.5546 427728  
14.5879 428800  
14.6212 429872  
14.6544 430944  
14.6877 432016  
14.721 433088  
14.7543 434160  
14.7876 435232  
14.8208 436304  
14.8541 437376  
14.8874 438448  
14.9207 439520  
14.954 440592  
14.9872 441128  
15.0205 442200  
15.0538 443272  
15.0871 444344  
15.1204 445416  
15.1536 446488  
15.1869 447560  
15.2202 448632  
15.2535 449704  
15.2868 450776  
15.32 451848  
15.3533 452920  
15.3866 453992  
15.4199 455064  
15.4532 456136  
15.4864 457208  
15.5197 458280  
15.553 458816  
15.5863 459888  
15.6196 460960  
15.6528 462032

15.6861 463104  
15.7194 464176  
15.7527 465248  
15.786 466320  
15.8192 467392  
15.8525 468464  
15.8858 469536  
15.9191 470608  
15.9524 471680  
15.9856 472752  
16.0189 473824  
16.0522 474896  
16.0855 475968  
16.1188 476504  
16.152 477576  
16.1853 478648  
16.2186 479720  
16.2519 480792  
16.2852 481864  
16.3184 482936  
16.3517 484008  
16.385 485080  
16.4183 486152  
16.4516 487224  
16.4848 488296  
16.5181 489368  
16.5514 490440  
16.5847 491512  
16.618 492584  
16.6512 493656  
16.6845 494192  
16.7178 495264  
16.7511 496336  
16.7844 497408  
16.8176 498480  
16.8509 499552  
16.8842 500624  
16.9175 501696  
16.9508 502768  
16.984 503840  
17.0173 504912  
17.0506 505984  
17.0839 507056  
17.1172 508128  
17.1504 509200  
17.1837 510272  
17.217 510808  
17.2503 511880  
17.2836 512952  
17.3168 514024  
17.3501 515096  
17.3834 516168  
17.4167 517240  
17.45 518312  
17.4832 519384  
17.5165 520456  
17.5498 521528  
17.5831 522600  
17.6164 523672  
17.6496 524744  
17.6829 525816  
17.7162 526888  
17.7495 527960  
17.7828 528496

17.816 529568  
17.8493 530640  
17.8826 531712  
17.9159 532784  
17.9492 533856  
17.9824 534928  
18.0157 536000  
18.049 537072  
18.0823 538144  
18.1156 539216  
18.1488 540288  
18.1821 541360  
18.2154 542432  
18.2487 543504  
18.282 544576  
18.3152 545648  
18.3485 546184  
18.3818 547256  
18.4151 548328  
18.4484 549400  
18.4816 550472  
18.5149 551544  
18.5482 552616  
18.5815 553688  
18.6148 554760  
18.648 555832  
18.6813 556904  
18.7146 557976  
18.7479 559048  
18.7812 560120  
18.8144 561192  
18.8477 562264  
18.881 563336  
18.9143 563872  
18.9476 564944  
18.9808 566016  
19.0141 567088  
19.0474 568160  
19.0807 569232  
19.114 570304  
19.1472 571376  
19.1805 572448  
19.2138 573520  
19.2471 574592  
19.2804 575664  
19.3136 576736  
19.3469 577808  
19.3802 578880  
19.4135 579952  
19.4468 580488  
19.48 581560  
19.5133 582632  
19.5466 583704  
19.5799 584776  
19.6132 585848  
19.6464 586920  
19.6797 587992  
19.713 589064  
19.7463 590136  
19.7796 591208  
19.8128 592280  
19.8461 593352  
19.8794 594424  
19.9127 595496

19.946	596568
19.9792	597640
20.0125	598176
20.0458	599248
20.0813	600320
20.1168	601392
20.1523	602464
20.1877	603536
20.2232	604608
20.2587	605680
20.2942	606752
20.3297	607824
20.3652	608896
20.4007	609968
20.4362	611040
20.4716	612112
20.5071	613184
20.5426	614256
20.5781	615328
20.6136	615864
20.6491	616936
20.6846	618008
20.7201	619080
20.7556	620152
20.791	621224
20.8265	622296
20.862	623368
20.8975	624440
20.933	625512
20.9685	626584
21.004	627656
21.0395	628728
21.0749	629800
21.1104	630872
21.1459	631944
21.1814	633016
21.2169	633552
21.2524	634624
21.2879	635696
21.3234	636768
21.3588	637840
21.3943	638912
21.4298	639984
21.4653	641056
21.5008	642128
21.5363	643200
21.5718	644272
21.6073	645344
21.6428	646416
21.6782	647488
21.7137	648560
21.7492	649632
21.7847	650168
21.8202	651240
21.8557	652312
21.8912	653384
21.9267	654456
21.9621	655528
21.9976	656600
22.0331	657672
22.0686	658744
22.1041	659816
22.1396	660888
22.1751	661960



22.2106 663032  
22.246 664104  
22.2815 665176  
22.317 666248  
22.3525 667320  
22.388 667856  
22.4235 668928  
22.459 670000  
22.4945 671072  
22.53 672144  
22.5654 673216  
22.6009 674288  
22.6364 675360  
22.6719 676432  
22.7074 677504  
22.7429 678576  
22.7784 679648  
22.8139 680720  
22.8493 681792  
22.8848 682864  
22.9203 683936  
22.9558 685008  
22.9913 685544  
23.0268 686616  
23.0623 687688  
23.0978 688760  
23.1332 689832  
23.1687 690904  
23.2042 691976  
23.2397 693048  
23.2752 694120  
23.3107 695192  
23.3462 696264  
23.3817 697336  
23.4172 698408  
23.4526 699480  
23.4881 700552  
23.5236 701624  
23.5591 702696  
23.5946 703232  
23.6301 704304  
23.6656 705376  
23.7011 706448  
23.7365 707520  
23.772 708592  
23.8075 709664  
23.843 710736  
23.8785 711808  
23.914 712880  
23.9495 713952  
23.985 715024  
24.0204 716096  
24.0559 717168  
24.4463 717704  
24.8012 718240  
25.1561 718776  
25.5464 719312  
25.9013 719848  
26.2562 720384  
26.6466 720920  
27.0014 721456  
27.3563 721992  
27.7112 722528  
28.1016 723064

28.4564 723600  
28.8113 724136  
29.2017 724672  
29.5566 725208  
29.9115 725744  
30.2308 726280  
30.5148 726816  
30.7803 727352  
31.0433 727888  
31.3064 728424  
31.5523 728960  
31.7627 729496  
31.9732 730032  
32.131 730568  
32.2888 731104  
32.4821 731640  
32.6228 732176  
32.7109 732712  
32.8174 733248  
32.9153 43680  
32.9242 43648  
32.9508 43144  
32.9692 42640  
33.0316 41632  
33.0571 41600  
33.0671 41096  
33.1019 40592  
33.1119 40088  
33.1556 39552  
33.1739 38544  
33.1995 38512  
33.2084 38008  
33.2526 37504  
33.2615 36968  
33.2709 36464  
33.3064 35960  
33.3409 35456  
33.3508 34952  
33.3935 34416  
33.4024 33912  
33.4124 33408  
33.4389 32872  
33.4916 31864  
33.5005 31832  
33.5099 31328  
33.5454 30824  
33.5799 30320  
33.5898 29816  
33.6325 29280  
33.6414 28776  
33.6514 28272  
33.6769 27736  
33.6858 27232  
33.6958 26728  
33.7384 26192  
33.7484 25688  
33.7839 25184  
33.8194 24680  
33.8709 24176  
33.8809 23672  
33.8898 23136  
33.9164 22632  
33.9342 22128  
33.9779 21624

33.9868	21088
34.0134	20584
34.0318	20080
34.0673	19576
34.1199	19072
34.1554	18568
34.1737	18064
34.2253	17560
34.2524	17592
34.3068	17088
34.5323	16584
34.8547	16080
35.471	512952
35.6628	513488
36.2041	514024
36.586	514560
36.8995	515096
37.0417	515632
37.1308	516168
37.2298	104112
37.3116	103576
37.3745	103040
37.4105	102504
37.4294	101968
37.4825	101432
37.492	100896
37.5545	100360
37.564	99824
37.5734	99288
37.6265	98752
37.6531	98216
37.6625	97680
37.672	97144
37.7251	96608
37.7516	95536
37.7611	95000
37.7705	94464
37.7971	93928
37.8502	92856
37.8691	91248
37.9051	90712
37.9145	89640
37.9582	89104
37.9942	87496
38.0131	86424
38.0396	85888
38.0491	85352
38.0928	84280
38.1022	83744
38.1065	82136
38.1321	81384
38.142	80344
38.1515	79808
38.1951	78768
38.2046	78264
38.2311	76720
38.2406	76216
38.25	75712
38.2766	75208
38.286	74704
38.2955	74200
38.3381	73696
38.3481	72688
38.3736	70608

38.3825 70104  
38.3915 68560  
38.4356 67520  
38.4446 66952  
38.4711 64368  
38.4806 64336  
38.49 63296  
38.5166 62792  
38.5255 62224  
38.5344 60176  
38.5781 59672  
38.5876 59104  
38.6131 56992  
38.622 55984  
38.6309 54440  
38.6751 53400  
38.6846 52832  
38.7101 50720  
38.7201 49712  
38.7295 48640  
38.755 47568  
38.764 46560  
38.7729 45016  
38.8161 43976  
38.825 42936  
38.8521 39816  
38.8615 48920  
38.8709 48040  
38.9146 48576  
38.9241 49112  
38.9335 49648  
38.9601 48608  
38.9695 49080  
38.9961 49584  
39.0055 50120  
39.0149 49584  
39.0586 50120  
39.0681 50656  
39.0946 51192  
39.1041 51728  
39.1135 52264  
39.1572 52736  
39.1666 53240  
39.1932 53776  
39.2026 54280  
39.2121 54816  
39.2386 55320  
39.2481 55824  
39.2917 56360  
39.3012 56864  
39.3106 57400  
39.3372 57936  
39.3466 58472  
39.3561 59008  
39.3997 59544  
39.4092 60080  
39.4529 60552  
39.4623 61088  
39.4889 61624  
39.5154 62128  
39.5249 62632  
39.5343 63168  
39.5609 63704  
39.5703 64240

39.614 64776  
39.7251 65280  
39.7782 65312  
39.7877 65848  
39.8122 66320  
39.8901 66352  
40.2447 66384  
40.2541 66920  
40.2807 67456  
40.2891 67928  
40.2985 67992  
40.3241 68496  
40.334 68528  
40.3777 69064  
40.3871 69600  
40.3966 70136  
40.4231 70672  
40.4326 71208  
40.4763 71744  
40.4857 72280  
40.4951 72816  
40.5207 73320  
40.5306 73352  
40.5401 73888  
40.5656 74392  
40.5756 74424  
40.6182 74928  
40.6282 74960  
40.6371 75464  
40.6726 75496  
40.7152 76000  
40.7212 536  
42.2271 7504  
43.7603 8576  
43.7868 11792  
43.7963 12864  
43.8228 15544  
43.8323 17688  
43.8588 19296  
43.8854 20368  
43.8948 20904  
43.9214 69680  
45.4286 70216  
45.473 70752  
45.5284 71288  
47.4318 536  
48.7398 19296  
50.0685 21440  
50.078 45024  
50.1045 46632  
50.114 48776  
50.1405 49848  
50.237 50384  
50.309 50920  
50.3795 51456  
50.5126 51992  
50.5663 52528  
52.4809 536  
53.9621 16080  
55.4096 34840  
55.419 35912  
55.4456 39664  
55.4545 40200  
55.5872 40736

55.7541	41272
55.7641	41808
57.6787	536
59.1696	20368
60.6171	22512
60.6266	23584
60.6531	24656
60.6797	27336
60.6891	28408
60.7149	31624
60.7964	32160
60.9215	32696
60.9309	33232
61.1015	32696
61.128	32345
61.1375	32169
61.1815	31993
61.1914	31817
61.2008	31105
61.2274	30393
61.2368	30217
61.2805	30042
62.1288	29866
62.1382	29690
62.1477	29514
62.1571	29338
62.1666	29162
62.176	28986
62.2026	28810
62.212	28635
62.2557	28459
62.2651	28283
62.2746	28107
62.3011	27931
62.3106	27755
62.3371	27579
62.3466	26867
62.356	26692
62.3997	26516
62.4091	26340
62.4357	25628
62.4451	25452
62.4546	24740
62.4982	24564
62.5077	24388
62.5342	23048
62.5531	22512
62.5986	21976
62.6782	20904
62.6971	20368
62.7768	19296
62.7862	23048
62.8754	21976
63.6038	21440
63.8429	23048
65.0844	23584
65.1106	24120
65.2168	23727
65.2434	23530
65.27	23333
65.3136	23136
65.3231	22939
65.3325	22742
65.342	22545

65.3514	22348
65.3608	22152
65.3874	21955
65.414	21758
65.4234	21561
65.4842	21364
65.5279	21167
65.5373	20970
65.5639	20773
65.5733	20576
65.5828	20379
65.6264	20183
65.6359	19986
65.6624	19789
65.689	19592
65.6984	19395
65.725	19198
65.7344	19001
65.761	18804
65.7876	18607
65.797	18410
65.8064	18214
65.8501	18017
65.8596	17820
65.8861	17623
65.9127	16616
66.6153	17152
66.7027	17688
66.8072	18224
66.8432	18760
66.8963	19296
67.071	19832
68.6966	20368
69.624	20904
69.8913	21440
70.21	21976
70.4568	22512
70.7	23048
71.1297	23584
71.1846	24120
71.2737	24656
71.3971	25192
71.4691	25728
71.5051	26264
71.56	26800
71.7004	27336
71.7193	27872
71.8427	28408
71.93	28944
71.9849	29480
72.1579	30016
72.5971	30552
72.6331	31088
72.652	31624
72.688	32160
72.724	32696
72.7771	33232
72.8302	33768
72.8491	34304
72.8851	34840
72.9211	35376
72.9571	35912
72.976	36448
73.0291	36984

73.0749	36448
73.1186	36101
73.128	35928
73.1546	35754
73.164	35580
73.1735	35407
73.2	35233
73.2095	35059
73.2535	34886
73.263	34712
73.2724	34003
73.299	33829
73.3084	33655
73.3179	33482
73.3615	33308
73.371	33134
73.3975	32961
73.407	32787
73.4164	32613
73.443	32440
73.4524	32266
73.4961	32093
73.5398	31919
73.5492	31745
73.5587	31572
73.7222	31398
73.9713	31224
73.9807	31051
73.9902	30877
73.9996	30703
74.0091	30530
74.0185	30356
74.0451	30183
74.0545	30009
74.0982	29835
74.1076	29662
74.1171	29488
74.1436	29314
74.1531	29141
74.1967	28967
74.2066	28794
74.216	28620
74.2426	27910
74.252	27737
74.2957	27563
74.3051	27389
74.3146	27216
74.3411	27042
74.3506	25728
74.396	25192
74.4586	24656
74.5382	24120
74.5571	23584
74.5931	23048
74.6127	536



