

به نام خدا

Data Communication Networks, Lab #3

Instructor: Dr. MohammadReza Pakravan

MAC Sublayer Simulation Using NS3

علی فتحی ۹۴۱۰۹۲۰۵

مقدمه:

بخش ۲.۱: نمونه اجرای کد نوشته در صورت سوال، در کامپیوتر من بدین صورت است:

```
AUTHORS      README      build      scratch      testpy.sup  waf          wutils.py
CHANGES.html RELEASE_NOTES contrib      src          utils       waf-tools   wutils.pyc
LICENSE      VERSION     doc         test.py      utils.py    waf.bat
Makefile     bindings   examples   testpy-output  utils.pyc   wscript
Macintoshs-MacBook-Pro:ns-3-dev macintosh$ sudo ./waf --run "scratch/ns3Test --a=2 --b=5"
Waf: Entering directory `/usr/local/bin/ns-3-allinone/ns-3-dev/build'
Waf: Leaving directory `/usr/local/bin/ns-3-allinone/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (11.510s)
a + b = 7Macintoshs-MacBook-Pro:ns-3-dev macintosh$
```

چهار رقم سمت راست شماره دانشجویی من، ۹۲۰۵ است لذا Base Address را برابر 19.20.5.0 قرار می‌دهم.

همچنین نرخ های ترافیک بر اساس شماره دانشجویی من بدین صورت خواهد بود:

- $16 + \frac{ab}{10} \text{ Kbps} = 16 + 94/10 = 25.4 \text{ Kbps}$
- $32 + \frac{bc}{10} \text{ Kbps} = 32 + 41/10 = 36.1 \text{ Kbps}$
- $64 + \frac{cd}{10} \text{ Kbps} = 64 + 10/10 = 65.0 \text{ Kbps}$
- $128 + \frac{de}{10} \text{ Kbps} = 128 + 09/10 = 128.9 \text{ Kbps}$
- $256 + \frac{ef}{10} \text{ Kbps} = 256 + 92/10 = 265.2 \text{ Kbps}$
- $512 + \frac{fg}{10} \text{ Kbps} = 512 + 20/10 = 514 \text{ Kbps}$
- $1024 + \frac{gh}{10} \text{ Kbps} = 1024 + 05/10 = 1024.5 \text{ Kbps}$

توضیح API کد:

1 (توپولوژی:

بر اساس خواسته سوال، توپولوژی به شکل زیر است:

```
// Network Topology
//
//  n1  n2  n3  n4  n5  n6  n7  n8
//  |   |   |   |   |   |   |
//  =====
//                LAN 19.20.5.0
```

که در واقع همان CSMA است (چیزی که از این پروتکل استفاده می کند. تفاوتی در دستورات ندارد).

2 (ساختن Node ها

بصورت زیر، به کمک NodeContainer، ۸ نقطه برای استفاده از این شبکه تعریف می کنیم:

```
uint32_t nCsm = 8;

NodeContainer csmaNodes;
csmaNodes.Create (nCsm);
```

3 (ساختن Channel

به کمک CsmaHelper، کانال را تعریف کرده و ویژگی های تأخیر و DataRate آن را مشخص می کنیم:

```
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("1024Kbps"));
csma.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

4 (ساختن دستگاه ها و Install کردن نقاط روی کانال

به کمک NetDeviceContainer، این کار را انجام می دهیم:

```
NetDeviceContainer csmaDevices;
csmaDevices = csma.Install (csmaNodes);
```

5 (ساختن پروتکل اینترنت

پروتکل اینترنت را به کمک InternetStackHelper، روی نقاط شبکه قرار می دهیم:

```
InternetStackHelper stack;
stack.Install (csmaNodes);
```

6) ساختن IPv4 برای ارتباط نقاط شبکه

در اینجا Net Mask (آدرس‌های معتبر در این شبکه) و Base Address را به کمک Ipv4AddressHelper تعیین می‌کنیم. سپس آن را به لایه پایین تر (Device ها) assign می‌کنیم.

```
Ipv4AddressHelper address;  
address.SetBase ("19.20.5.0", "255.255.255.0");  
Ipv4InterfaceContainer csmaInterfaces;  
csmaInterfaces = address.Assign (csmaDevices);
```

7) آماده کردن Server ها برای دریافت بسته‌ها

از Echo UDP استفاده می‌کنیم. برای ارتباط‌ها، همگی از Port شماره ۹ استفاده می‌کنیم (پورت شماره ۹، در واقع ویژگی زیر را دارد:

Service Name and Transport Protocol Port Number Registry

Service Name	Port Number	Transport Protocol
	8	udp
discard	9	tcp
discard	9	udp
discard	9	sctp

و Discard Protocol در واقع یک سرویس روی Internet Protocol تعریف شده در RFC 863 برای اعمال testing ، debugging ، measurement و host management است). سپس سرور را روی اپلیکیشن (ApplicationContainer) راه‌اندازی می‌نماییم. زمان فعالیت سرور را نیز از زمان ۱ تا ۱۵ ثانیه در نظر می‌گیریم.

```
endTime = Seconds (15.0);  
  
UdpEchoServerHelper echoServer5 (9);  
UdpEchoServerHelper echoServer6 (9);  
UdpEchoServerHelper echoServer3 (9);  
UdpEchoServerHelper echoServer4 (9);  
  
ApplicationContainer serverApps5 = echoServer5.Install (csmaNodes.Get (4));  
serverApps5.Start (Seconds (1.0));  
serverApps5.Stop (endTime);  
  
ApplicationContainer serverApps6 = echoServer6.Install (csmaNodes.Get (5));  
serverApps6.Start (Seconds (1.0));  
serverApps6.Stop (endTime);  
  
ApplicationContainer serverApps3 = echoServer3.Install (csmaNodes.Get (2));  
serverApps3.Start (Seconds (1.0));  
serverApps3.Stop (endTime);  
  
ApplicationContainer serverApps4 = echoServer4.Install (csmaNodes.Get (3));  
serverApps4.Start (Seconds (1.0));  
serverApps4.Stop (endTime);
```

8) تعیین Client ها برای ارسال بسته‌ها

اینجا هم از Echo UDP استفاده می‌کنیم و آن را روی اپلیکیشن (ApplicationContainer) نصب می‌نماییم. پورت‌ها در اینجا نیز شماره ۹ هستند (دقت شود شماره متغیرها یکی کمتر از عدد واقعی نقاط است زیرا اندیس‌ها از ۰ شروع می‌شوند).

```
maxPackets = UIntegerValue (2000);
interval = TimeValue (Seconds (0.315));

UdpEchoClientHelper echoClient1 (csmaInterfaces.GetAddress (4), 9);
echoClient1.SetAttribute ("MaxPackets", maxPackets);
echoClient1.SetAttribute ("Interval", interval);
echoClient1.SetAttribute ("PacketSize", UIntegerValue (1024));

UdpEchoClientHelper echoClient2 (csmaInterfaces.GetAddress (5), 9);
echoClient2.SetAttribute ("MaxPackets", maxPackets);
echoClient2.SetAttribute ("Interval", interval);
echoClient2.SetAttribute ("PacketSize", UIntegerValue (1024));

UdpEchoClientHelper echoClient7 (csmaInterfaces.GetAddress (2), 9);
echoClient7.SetAttribute ("MaxPackets", maxPackets);
echoClient7.SetAttribute ("Interval", interval);
echoClient7.SetAttribute ("PacketSize", UIntegerValue (1024));

UdpEchoClientHelper echoClient8 (csmaInterfaces.GetAddress (3), 9);
echoClient8.SetAttribute ("MaxPackets", maxPackets);
echoClient8.SetAttribute ("Interval", interval);
echoClient8.SetAttribute ("PacketSize", UIntegerValue (1024));

ApplicationContainer clientApps1 = echoClient1.Install (csmaNodes.Get (0));
clientApps1.Start (Seconds (2.0));
clientApps1.Stop (endTime);

ApplicationContainer clientApps2 = echoClient2.Install (csmaNodes.Get (1));
clientApps2.Start (Seconds (2.0));
clientApps2.Stop (endTime);

ApplicationContainer clientApps7 = echoClient7.Install (csmaNodes.Get (6));
clientApps7.Start (Seconds (2.0));
clientApps7.Stop (endTime);

ApplicationContainer clientApps8 = echoClient8.Install (csmaNodes.Get (7));
clientApps8.Start (Seconds (2.0));
clientApps8.Stop (endTime);
```

در اینجا چند نکته درباره Attribute ها وجود دارد. نرخ ترافیک به عنوان ورودی به برنامه داده می‌شود. برای آن که مدل استاندارد باشد، سایز packet ها را برابر و همه را 1024 Byte یا 1 KByte می‌گیریم (زیرا اندازه packet ها معمولاً بر اساس استاندارد تعیین می‌شود و بسته به نرخ ترافیک تغییر نمی‌کند). در نتیجه Interval باید بر اساس نرخ ترافیک تعیین شود. ترافیک هر Client ، مستقلاً باید برابر با عدد خواسته شده در صورت سوال باشد (که در مقدمه محاسبه شده است). این عدد برابر نسبت سایز packet به Interval است. پس Interval از تقسیم 1024 Byte یا معادلات 8 Kbit به BitRate بدست می‌آید. تا سه رقم، این مقدار برای ترافیک‌های من بدین صورت است:

```
{[Traffic, Interval] = [25.4Kbps, 314.96ms], [36.1Kbps, 221.6ms], [65.0Kbps, 123.077ms],
[128.9Kbps, 62.06ms], [265.2Kbps, 30.166ms], [514.0Kbps, 15.56ms], [1024.5Kbps, 7.809ms]}
```

پس، maxPackets را مناسب است عددی بگیریم که همه این مقادیر، در بازه ۱۵ ثانیه‌ای ارسال جا شوند، و مقدار ۲۰۰۰ برای این مهم خوب است (حدود ۱۹۰۰ بسته در ۱۵ ثانیه، در بازه‌های ۷.۸۰۹ میلی ثانیه‌ای ارسال می‌شوند). نکته این که در کدام نرخ ترافیک هستیم، در ورودی با یک عدد بین ۱ تا ۷ مشخص می‌شود. در این باره توضیح داده می‌شود.

9) تعیین فایل‌های خروجی

در انتها، سه نوع خروجی نیز برای کد محاسبه می‌گردد؛ .tr ، .pcap و دو نوع فایل .xml . کاربری هر کدام مشخص است و در بخش خروجی‌ها نتایجشان مرور می‌شود.

```
std::string modeStr;
modeStr = std::to_string(modeNumber);
std::string fileName = "HW3/HW3_94109205_Mode_" + modeStr;
csma.EnablePcapAll (fileName);

AsciiTraceHelper asciiTraceHelper;
Ptr<OutputStreamWrapper> stream = asciiTraceHelper.CreateFileStream(fileName+".tr");
csma.EnableAsciiAll(stream);

AnimationInterface animationInterface(fileName+".xml");
animationInterface.SetConstantPosition (csmaNodes.Get (0), 50, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (1), 100, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (2), 150, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (3), 200, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (4), 250, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (5), 300, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (6), 350, 200);
animationInterface.SetConstantPosition (csmaNodes.Get (7), 400, 200);
animationInterface.EnableIpv4RouteTracking(fileName+".xml",
                                           Time (Seconds (1.0)),
                                           Time (endTime),
                                           Time (Seconds (5.0)));
```

یک نقطه لازم به ذکر است، برای انجام محاسبات delay و throughput ، از یک کتابخانه دیگر به نام FlowMonitorHelper استفاده شده است (ایده انجام این کار توسط بقیه افراد کلاس پیدا شده که با این کلاس، محاسبات دستی ساده‌تر می‌شود. همچنین در کدهای معمول نوشته شده در سایت‌های مختلف نیز برای محاسبه delay و throughput از همین کتابخانه استفاده شده است). نحوه استفاده از آن نیز در زیر آمده است:

Usage

The module usage is extremely simple. The helper will take care of about everything.

The typical use is:

```
// Flow monitor
Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();

Simulator::Stop (Seconds(stop_time));
Simulator::Run ();

flowMonitor->SerializeToXmlFile("NameOfFile.xml", true, true);
```

درباره ورودی دادن و اجرا:

کدها با دستور زیر اجرا می‌شوند:

```
Macintoshs-MacBook-Pro:HW3 macintosh$ sudo ./waf --run "scratch/csma --modeNumber=3"
```

و می‌گویید برنامه با کدام نرخ ارسال کار کند؛ که عددی بین ۱ متناظر با نرخ 25.4Kbps و ۷ متناظر با نرخ 1024.5Kbps است. برای تکلیف نیز با هر ۷ مقدار برنامه اجرا شده و نتایج در فولدر خروجی آمده است.

درباره خروجی کد:

نمونه اعداد چاپ شده در terminal به صورت زیر است:

```
At time 14.483s client received 1024 bytes from 19.20.5.4 port 9
At time 14.6312s client sent 1024 bytes to 19.20.5.6 port 9
At time 14.6312s client sent 1024 bytes to 19.20.5.7 port 9
At time 14.6312s client sent 1024 bytes to 19.20.5.4 port 9
At time 14.6312s client sent 1024 bytes to 19.20.5.5 port 9
At time 14.6416s server received 1024 bytes from 19.20.5.1 port 49153
At time 14.6416s server sent 1024 bytes to 19.20.5.1 port 49153
At time 14.6519s client received 1024 bytes from 19.20.5.6 port 9
At time 14.6729s server received 1024 bytes from 19.20.5.8 port 49153
At time 14.6729s server sent 1024 bytes to 19.20.5.8 port 49153
At time 14.6833s client received 1024 bytes from 19.20.5.5 port 9
At time 14.694s server received 1024 bytes from 19.20.5.7 port 49153
At time 14.694s server sent 1024 bytes to 19.20.5.7 port 49153
At time 14.7044s client received 1024 bytes from 19.20.5.4 port 9
At time 14.8528s client sent 1024 bytes to 19.20.5.6 port 9
At time 14.8528s client sent 1024 bytes to 19.20.5.7 port 9
At time 14.8528s client sent 1024 bytes to 19.20.5.4 port 9
At time 14.8528s client sent 1024 bytes to 19.20.5.5 port 9
At time 14.8632s server received 1024 bytes from 19.20.5.1 port 49153
At time 14.8632s server sent 1024 bytes to 19.20.5.1 port 49153
At time 14.8735s client received 1024 bytes from 19.20.5.6 port 9
At time 14.8841s server received 1024 bytes from 19.20.5.8 port 49153
At time 14.8841s server sent 1024 bytes to 19.20.5.8 port 49153
At time 14.8945s client received 1024 bytes from 19.20.5.5 port 9
At time 14.9153s server received 1024 bytes from 19.20.5.7 port 49153
At time 14.9153s server sent 1024 bytes to 19.20.5.7 port 49153
At time 14.9257s client received 1024 bytes from 19.20.5.4 port 9
```

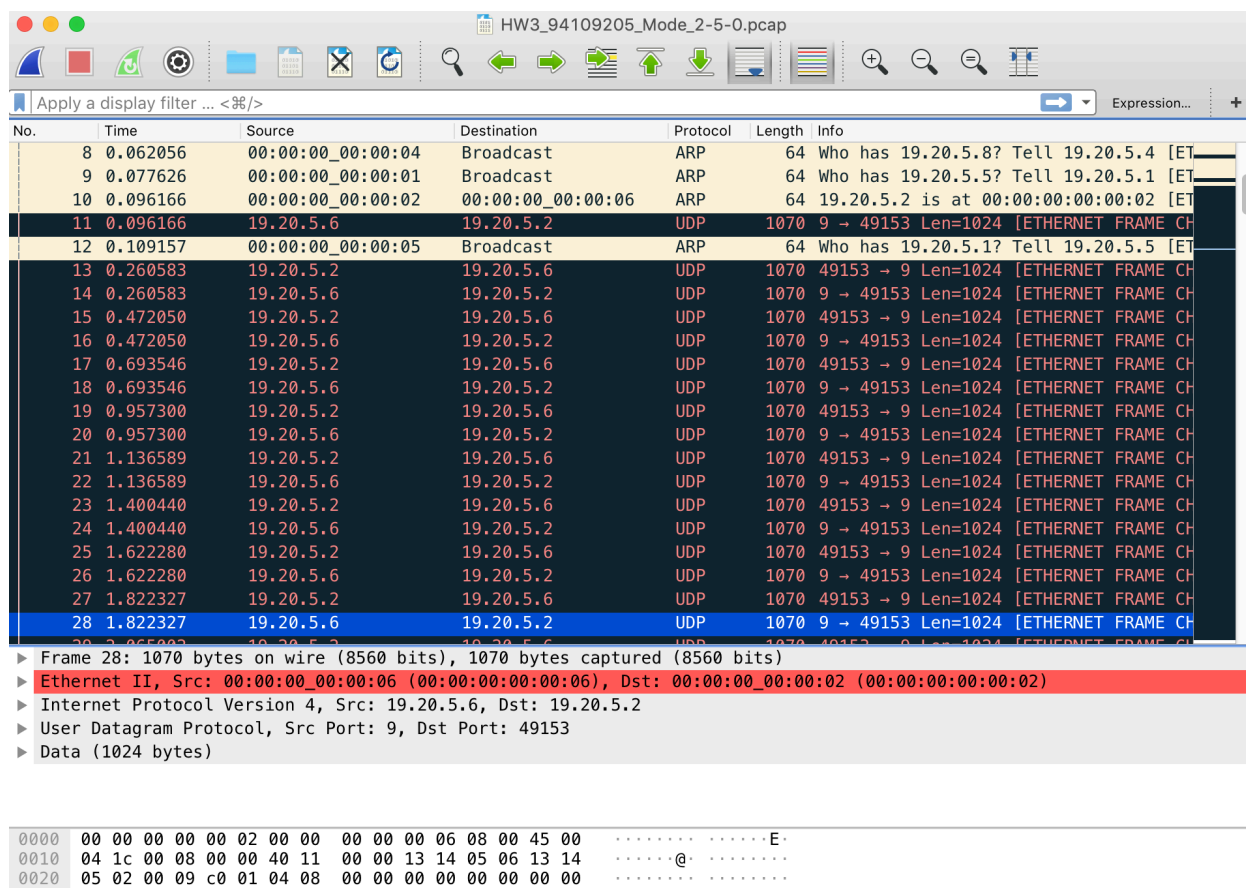
فایل‌های خروجی نمونه برای اجرای modeNumber=3 نیز برای مثال، به فرمت‌های زیر هستند:

```
Macintoshs-MacBook-Pro:HW3 macintosh$ ls
HW3_94109205_Mode_3-0-0.pcap  HW3_94109205_Mode_3-3-0.pcap  HW3_94109205_Mode_3-6-0.pcap  HW3_94109205_Mode_3.xml
HW3_94109205_Mode_3-1-0.pcap  HW3_94109205_Mode_3-4-0.pcap  HW3_94109205_Mode_3-7-0.pcap  HW3_94109205_Mode_3_flow.xml
HW3_94109205_Mode_3-2-0.pcap  HW3_94109205_Mode_3-5-0.pcap  HW3_94109205_Mode_3.tr
```

که کاربری آنها از پسوندشان مشخص است.

نمونه خروجی Pcap:

یک نمونه از این خروجی را با Wireshark باز کرده و نمایش می‌دهیم:



The screenshot shows the Wireshark interface with a packet capture file named 'HW3_94109205_Mode_2-5-0.pcap'. The packet list pane shows several packets, with packet 28 selected. The packet details pane shows the structure of packet 28: Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Data (1024 bytes). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
8	0.062056	00:00:00_00:00:04	Broadcast	ARP	64	Who has 19.20.5.8? Tell 19.20.5.4 [ET
9	0.077626	00:00:00_00:00:01	Broadcast	ARP	64	Who has 19.20.5.5? Tell 19.20.5.1 [ET
10	0.096166	00:00:00_00:00:02	00:00:00_00:00:06	ARP	64	19.20.5.2 is at 00:00:00:00:00:02 [ET
11	0.096166	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
12	0.109157	00:00:00_00:00:05	Broadcast	ARP	64	Who has 19.20.5.1? Tell 19.20.5.5 [ET
13	0.260583	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
14	0.260583	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
15	0.472050	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
16	0.472050	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
17	0.693546	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
18	0.693546	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
19	0.957300	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
20	0.957300	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
21	1.136589	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
22	1.136589	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
23	1.400440	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
24	1.400440	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
25	1.622280	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
26	1.622280	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH
27	1.822327	19.20.5.2	19.20.5.6	UDP	1070	49153 → 9 Len=1024 [ETHERNET FRAME CH
28	1.822327	19.20.5.6	19.20.5.2	UDP	1070	9 → 49153 Len=1024 [ETHERNET FRAME CH

Frame 28: 1070 bytes on wire (8560 bits), 1070 bytes captured (8560 bits)

- Ethernet II, Src: 00:00:00_00:00:06 (00:00:00:00:00:06), Dst: 00:00:00_00:00:02 (00:00:00:00:00:02)
- Internet Protocol Version 4, Src: 19.20.5.6, Dst: 19.20.5.2
- User Datagram Protocol, Src Port: 9, Dst Port: 49153
- Data (1024 bytes)

```

0000  00 00 00 00 00 02 00 00 00 00 06 08 00 45 00  .....E.
0010  04 1c 00 08 00 00 40 11 00 00 13 14 05 06 13 14  .....@.....
0020  05 02 00 09 c0 01 04 08 00 00 00 00 00 00 00 00
  
```

که ارتباط نقاط ۲ و ۶ را نشان می‌دهد (ملاحظه می‌شود در بسته‌ها information خاصی وجود ندارد).

نمونه خروجی xml (برای AnimationInterface):

ابتدای یک نمونه از این خروجی به فرمت زیر است:

```
HW3_94109205_Mode_4.xml
HW3_94109205_Mode_4.xml <anim ver="netanim-3.108" filetype="routing" >
1 <anim ver="netanim-3.108" filetype="routing" >
2 <rt t="1" id="0" info="Node: 0, Time: +1.0s, Local time: +1.0s, Ipv4ListRouting table
3   Priority: 0 Protocol: ns3::Ipv4StaticRouting
4   Node: 0, Time: +1.0s, Local time: +1.0s, Ipv4StaticRouting table
5   Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
6   127.0.0.0         0.0.0.0         255.0.0.0    U      0      -     -     0
7   19.20.5.0         0.0.0.0         255.255.255.0 U      0      -     -     1
8
9   Priority: -10 Protocol: ns3::Ipv4GlobalRouting
10  Node: 0, Time: +1.0s, Local time: +1.0s, Ipv4GlobalRouting table
11  Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
12  19.20.5.0         0.0.0.0         255.255.255.0 U      -      -     -     1
13
14 " />
15 <rt t="1" id="1" info="Node: 1, Time: +1.0s, Local time: +1.0s, Ipv4ListRouting table
16   Priority: 0 Protocol: ns3::Ipv4StaticRouting
17   Node: 1, Time: +1.0s, Local time: +1.0s, Ipv4StaticRouting table
18   Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
19   127.0.0.0         0.0.0.0         255.0.0.0    U      0      -     -     0
20   19.20.5.0         0.0.0.0         255.255.255.0 U      0      -     -     1
21
22   Priority: -10 Protocol: ns3::Ipv4GlobalRouting
23   Node: 1, Time: +1.0s, Local time: +1.0s, Ipv4GlobalRouting table
24   Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
25   19.20.5.0         0.0.0.0         255.255.255.0 U      -      -     -     1
26
27 " />
28 <rt t="1" id="2" info="Node: 2, Time: +1.0s, Local time: +1.0s, Ipv4ListRouting table
29   Priority: 0 Protocol: ns3::Ipv4StaticRouting
30   Node: 2, Time: +1.0s, Local time: +1.0s, Ipv4StaticRouting table
31   Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
32   127.0.0.0         0.0.0.0         255.0.0.0    U      0      -     -     0
33   19.20.5.0         0.0.0.0         255.255.255.0 U      0      -     -     1
34
35   Priority: -10 Protocol: ns3::Ipv4GlobalRouting
36   Node: 2, Time: +1.0s, Local time: +1.0s, Ipv4GlobalRouting table
37   Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
38   19.20.5.0         0.0.0.0         255.255.255.0 U      -      -     -     1
39
40 " />
```


نمونه خروجی xml (برای FlowMonitor):

ابتدای یک نمونه از این خروجی به فرمت زیر است:

```
<?xml version="1.0" ?>
<FlowMonitor>
  <FlowStats>
    <Flow flowId="1" timeFirstTxPacket="+2000000000.0ns" timeFirstRxPacket="+2070168375.0ns"
      timeLastTxPacket="+14971379570.0ns" timeLastRxPacket="+14997280000.0ns" delaySum="+703187750650.0ns"
      jitterSum="+8451516964.0ns" lastDelay="+7718230175.0ns" txBytes="453412" rxBytes="185152"
      txPackets="431" rxPackets="176" lostPackets="0" timesForwarded="0">
    <delayHistogram nBins="7728" >
      <bin index="70" start="0.07" width="0.001" count="1" />
      <bin index="123" start="0.123" width="0.001" count="1" />
      <bin index="165" start="0.165" width="0.001" count="1" />
      <bin index="208" start="0.208" width="0.001" count="1" />
      <bin index="210" start="0.21" width="0.001" count="1" />
      <bin index="284" start="0.284" width="0.001" count="1" />
      <bin index="294" start="0.294" width="0.001" count="1" />
      <bin index="317" start="0.317" width="0.001" count="1" />
      <bin index="339" start="0.339" width="0.001" count="1" />
      <bin index="371" start="0.371" width="0.001" count="1" />
      <bin index="466" start="0.466" width="0.001" count="1" />
      <bin index="509" start="0.509" width="0.001" count="1" />
      <bin index="585" start="0.585" width="0.001" count="1" />
      <bin index="691" start="0.691" width="0.001" count="1" />
      <bin index="849" start="0.849" width="0.001" count="1" />
    </delayHistogram>
  </Flow>
</FlowStats>
</FlowMonitor>
```

محاسبات رسم نمودار:

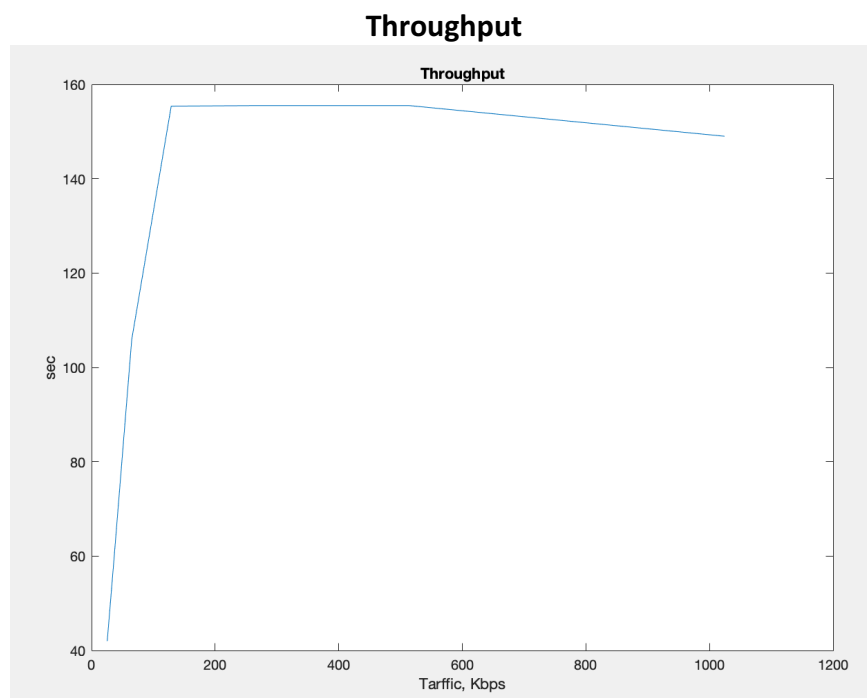
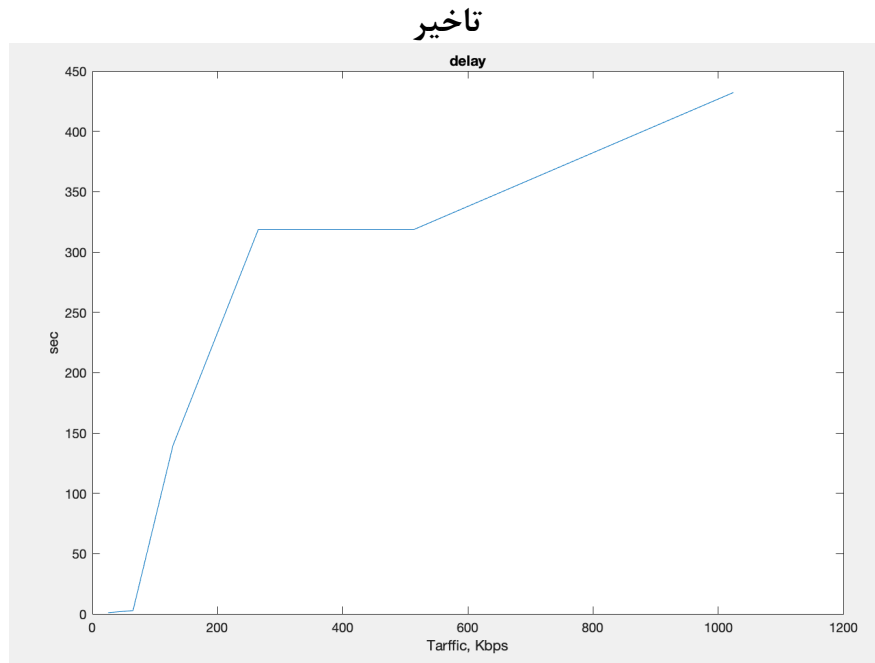
همانطور که اشاره شد، از یک روش ساده‌تر برای بدست آوردن نمودار استفاده می‌شود. به طور کلی سه راه وجود دارد:

- 1- مشاهده بسته‌های ارسالی و زمان آنها در Wireshark و محاسبه دستی مقادیر خواسته شده
- 2- نوشتن یک کد متلب برای استخراج داده‌ها از فایل tr. و انجام محاسبات روی آن و بدست آوردن مقادیر خواسته شده
- 3- استفاده از مقادیر delaySum (برای تاخیر) و rxPackets (برای throughput) که فایل xml در FlowMinitor محاسبه کرده است.

که از روش سوم استفاده می‌کنیم و با نوشتن کدی ساده نمودارهای مربوطه را رسم می‌کنیم که در ادامه می‌آیند. این کد در فولدر تمرین آمده است (با script به نام dn_plotter و تابعی به نام stc2atr).

رسم نمودار و تحلیل آن:

نمودارها به شکل زیر اند:



درمورد تاخیر انتظار داریم با شلوغ شدن کانال بیشتر شود. رابطه تاخیر ALOHA بصورت زیر است:

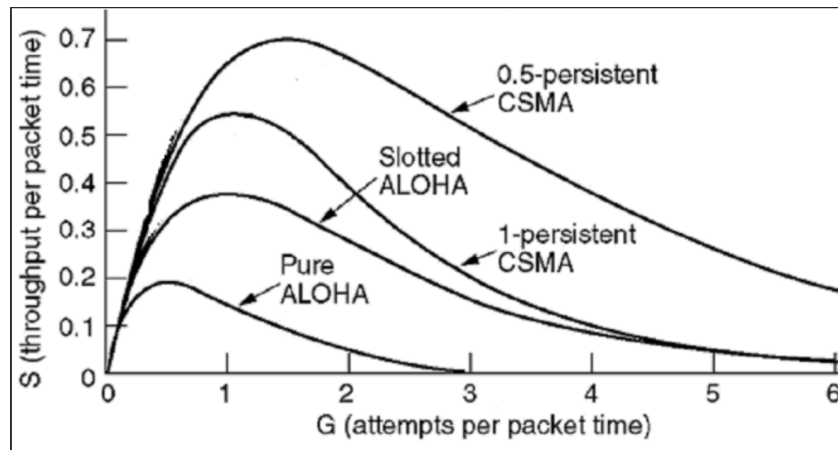
$$E[T_{ALOHA}] = X + t_{prop} + (e^{2G} - 1)(X + 2t_{prop} + B)$$

و اگر بصورت خیلی حدودی بگوییم به CSMA شبیه است، مشخص است که با افزایش بار شبکه فرم افزایشی دارد. در اینجا البته هم داده کم است و هم تعداد نقاط قابل توجه نیست پس رابطه ما دقیقاً بصورت نمایی ظاهر نمی‌شود؛ اما فرم صعودی خود را دارد.

در مورد throughput نیز انتظار داریم در ابتدا افزایش و سپس کاهش یابد زیرا اگر شباهت آن با ALOHA را در نظر بگیریم، برای آن داریم:

$$Throughput \approx Efficiency = Ge^{-G}$$

و در واقع CSMA یک 1-persistent است:



که به دلیل محدودیت‌های گفته شده، نمودار خیلی دقیق نیست اما به خوبی رفتار بالا را نشان می‌دهد، و بهترین عملکرد و بازدهی آن در حدود مقدار 128Kbps است.

***** پایان گزارش *****