

باسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس شبکه مخابرات داده

تمرین سری سوم

علی محرابیان 96102331

استاد: دکتر پاکروان

بهار 1399



در ابتدا توپولوژی شبکه را به صورت زیر می سازیم.

```
CommandLine cmd;  
cmd.Parse (argc, argv);  
  
NodeContainer csmaNodes;  
csmaNodes.Create(8);  
CsmaHelper csma;  
csma.SetChannelAttribute ("DataRate", StringValue ("1024kbps"));  
csma.SetChannelAttribute ("Delay", StringValue ("2ms"));  
NetDeviceContainer csmaDevices;  
csmaDevices = csma.Install (csmaNodes);  
InternetStackHelper stack;  
stack.Install(csmaNodes);  
Ipv4AddressHelper address;  
address.SetBase ("20.5.1.0", "255.255.255.0");  
Ipv4InterfaceContainer csmaInterfaces;  
csmaInterfaces = address.Assign (csmaDevices);
```

سپس پارامترهای مربوط به flow داده ها را تعیین می کنیم.

```
float datarate[7] = {16 + 9.5 , 32 + 5.1 , 64 + 1 , 128 + 0.2 , 256 + 20 , 512 + 5 , 1024 + 51};  
float packetSize = 512;  
float interval = packetSize * 8 / (datarate[6]*1024);  
  
UdpEchoServerHelper echoServer1(3);  
ApplicationContainer serverApps1 = echoServer1.Install (csmaNodes.Get (0));  
serverApps1.Start (Seconds (1.0));  
serverApps1.Stop (Seconds (10.0));  
  
UdpEchoClientHelper echoClient1 (csmaInterfaces.GetAddress (0), 3);  
echoClient1.SetAttribute ("MaxPackets", UIntegerValue (1000));  
echoClient1.SetAttribute ("Interval", TimeValue (Seconds (interval)));  
echoClient1.SetAttribute ("PacketSize", UIntegerValue (512));  
  
ApplicationContainer clientApps1 = echoClient1.Install (csmaNodes.Get (4));  
clientApps1.Start (Seconds (1.0));  
clientApps1.Stop (Seconds (11.0));
```



خروجی های به دست آمده را به صورت زیر ذخیره می کنیم.

```
Ptr<FlowMonitor> flowMonitor;  
FlowMonitorHelper flowHelper;  
flowMonitor = flowHelper.InstallAll();  
AnimationInterface::SetConstantPosition(csmaNodes.Get(0),10.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(1),20.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(2),30.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(3),40.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(4),50.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(5),60.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(6),70.0,0.0);  
AnimationInterface::SetConstantPosition(csmaNodes.Get(7),80.0,0.0);  
AnimationInterface anim (animname);  
Simulator::Stop (Seconds(20));  
Simulator::Run ();  
flowMonitor->SerializeToXmlFile(flowname, true, true);
```

فایل های pcap. را نیز به کمک دستور زیر ذخیره می کنیم.

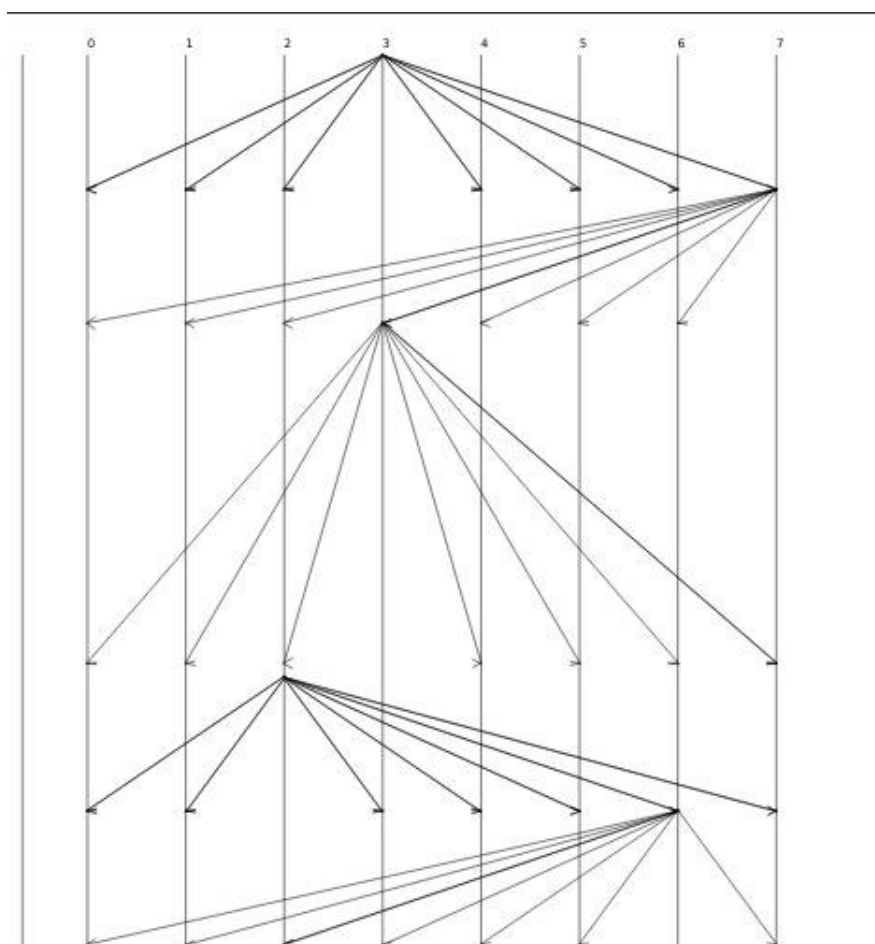
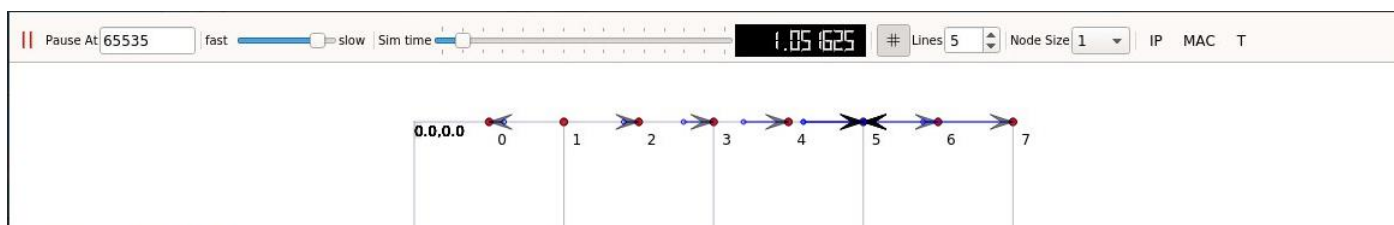
```
if(v==0) csma.EnablePcapAll (pcapname);
```

فایل های pcap. در wireshark قابل مشاهده هستند. فایل های anim در netAnim قابل مشاهده هستند.

خروجی های flowmonitor را نیز در متلب استفاده می کنیم.



شکل های زیر خروجی NetAnim هستند. خروجی زیر در ثانیه 1 ثبت شده است.





فایل های xml را به صورت struct در متلب ذخیره کرده و از آن ها برای نمایش خروجی های خواسته شده استفاده می کنیم.

```
x_nonpersistent = [xml2struct('flow512kB_00.xml')  
xml2struct('flow512kB_01.xml')  
xml2struct('flow512kB_02.xml')  
xml2struct('flow512kB_03.xml')  
xml2struct('flow512kB_04.xml')  
xml2struct('flow512kB_05.xml')  
xml2struct('flow512kB_06.xml')];
```

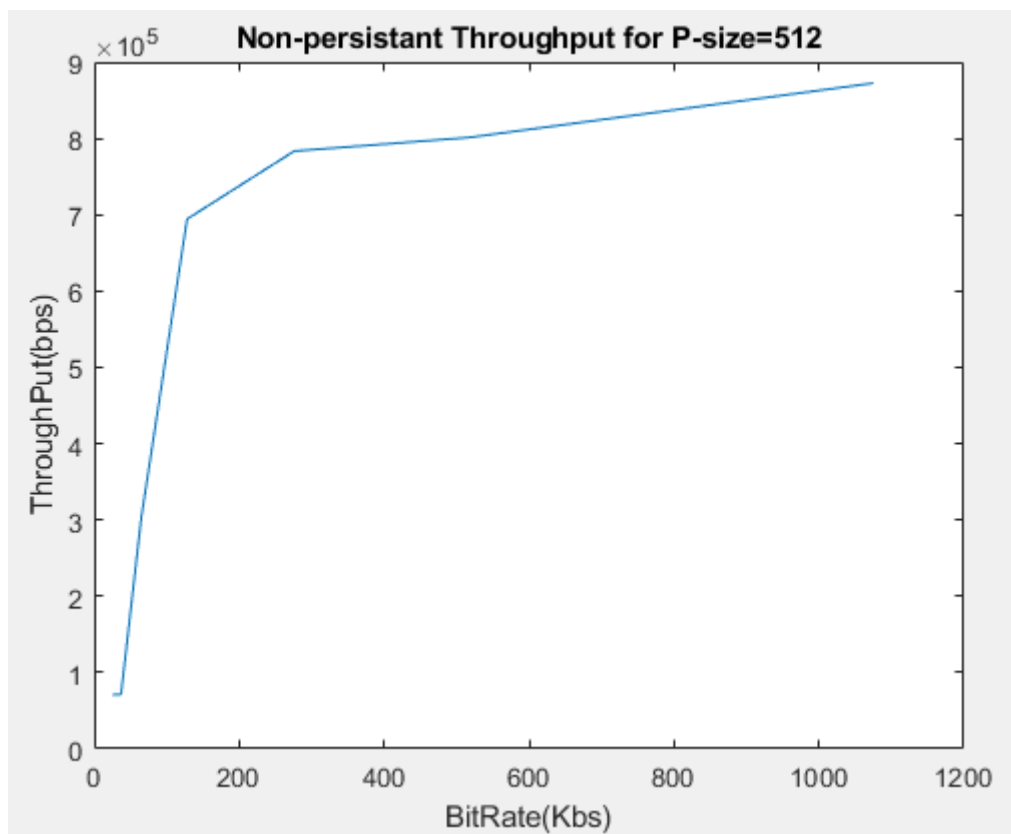
مقادیر delay و throughput به صورت زیر به دست می آیند.

```
BitRate = [25.5,37.1,65,128.2,276,517,1075]*1024;  
RxBytes_non = zeros(8,length(BitRate));  
Delay_non = zeros(8,length(BitRate));  
for j=1:length(BitRate)  
for i = 2:2:16  
RxBytes_non(i/2,j)= str2num(x_nonpersistent(j).Children(2).Children(i).Attributes(6).Value);  
Delay_non(i/2,j)= str2num(regexprep(x_nonpersistent(j).Children(2).Children(i).Attributes(1).Value,  
'ns','','ignorecase'));  
end  
end  
Throughput_non= sum(RxBytes_non)*8/9;  
Delay_non = sum(Delay_non)./Throughput_non;
```

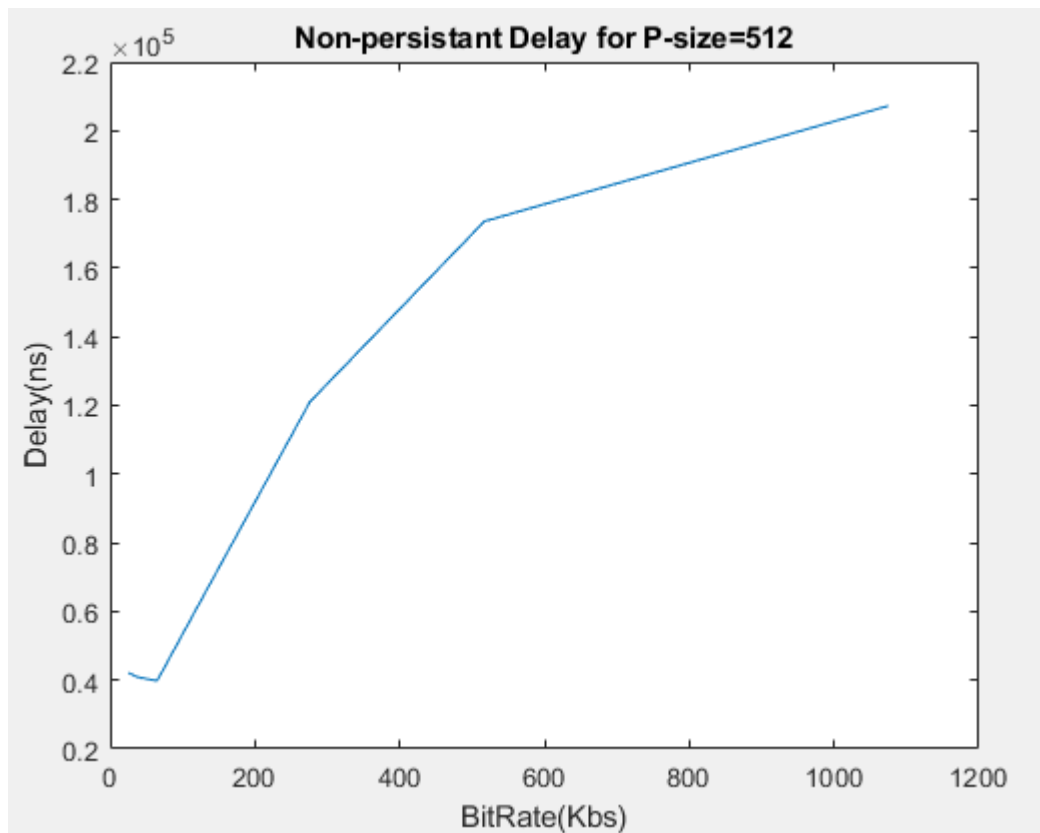
لازم به ذکر است که در تمامی قسمت ها، packet-size=512Kb در نظر گرفتیم و نمودار های زیر را به دست آوردیم.



:CSMA Non-Persistent



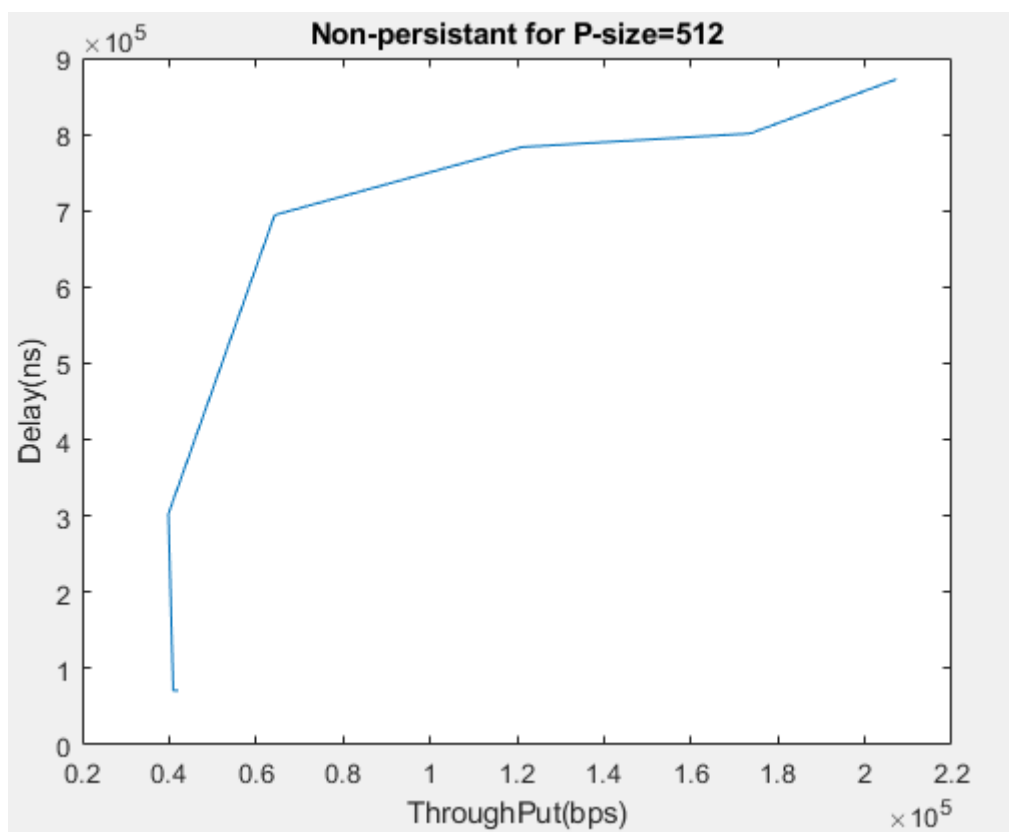
مقدار throughput به ازای ترافیک های مختلف به صورت بالا است. طبق انتظار، مقدار throughput در ابتدا با افزایش ریت کانال، افزایش می یابد و در ریت های بالا، تقریباً به حالت اشباع می رسد.



تاخیر شبکه با افزایش ریت کانال به صورت کاملاً صعودی افزایش می یابد که کاملاً مطابق انتظار است.



در نمودار زیر نیز تاخیر بر اساس throughput شبکه رسم می شود.



طبق انتظار، در throughput های بالا، تاخیر شبکه افزایش می یابد.



:ALOHA

در قسمت در csmaNetdevice تغییراتی ایجاد می کنیم. در ابتدا خاصیت carrier sense کانال را خنثی کرده به صورتی که هیچ وقت کانال را چک نکند.

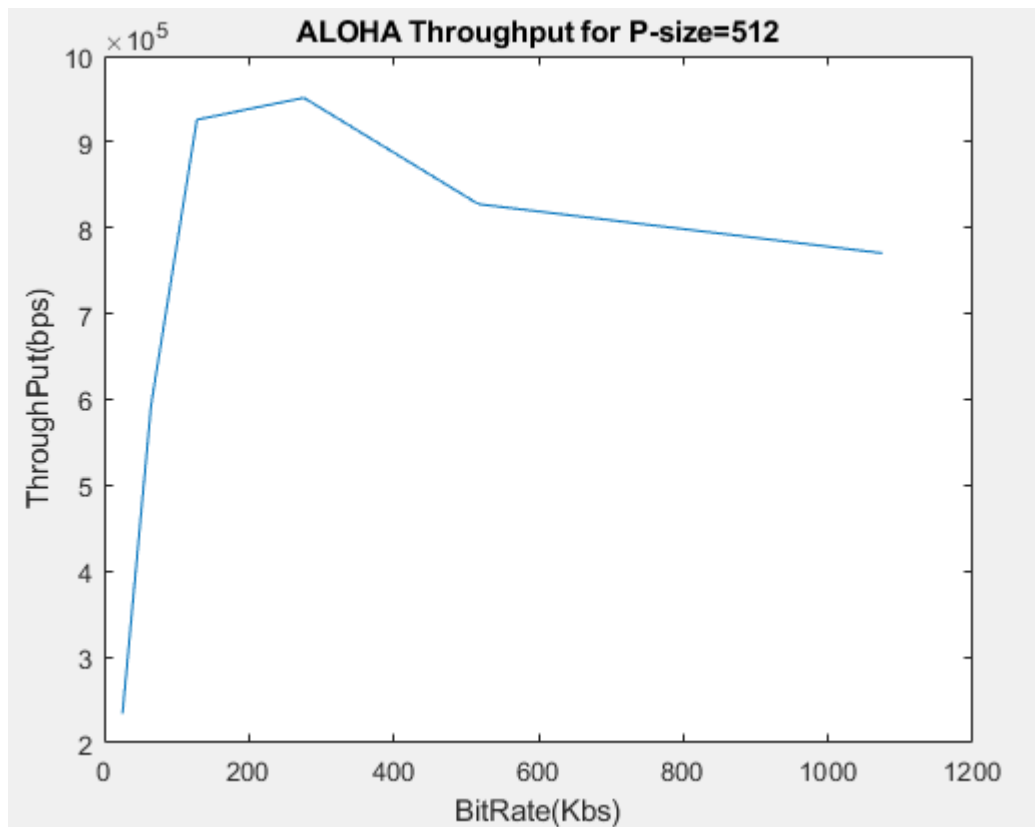
```
if (0 && (m_channel->GetState () != IDLE)) //0 for aloha because we don't need channel sensing
{
    //
    // The channel is busy -- backoff and rechedule TransmitStart() unless
    // we have exhausted all of our retries.
    //
    m_txMachineState = BACKOFF;
```

سپس عملیات انجام شده در مواقعی که کانال شلوع است را به قسمتی که برای اختلال است، منتقل می کنیم.

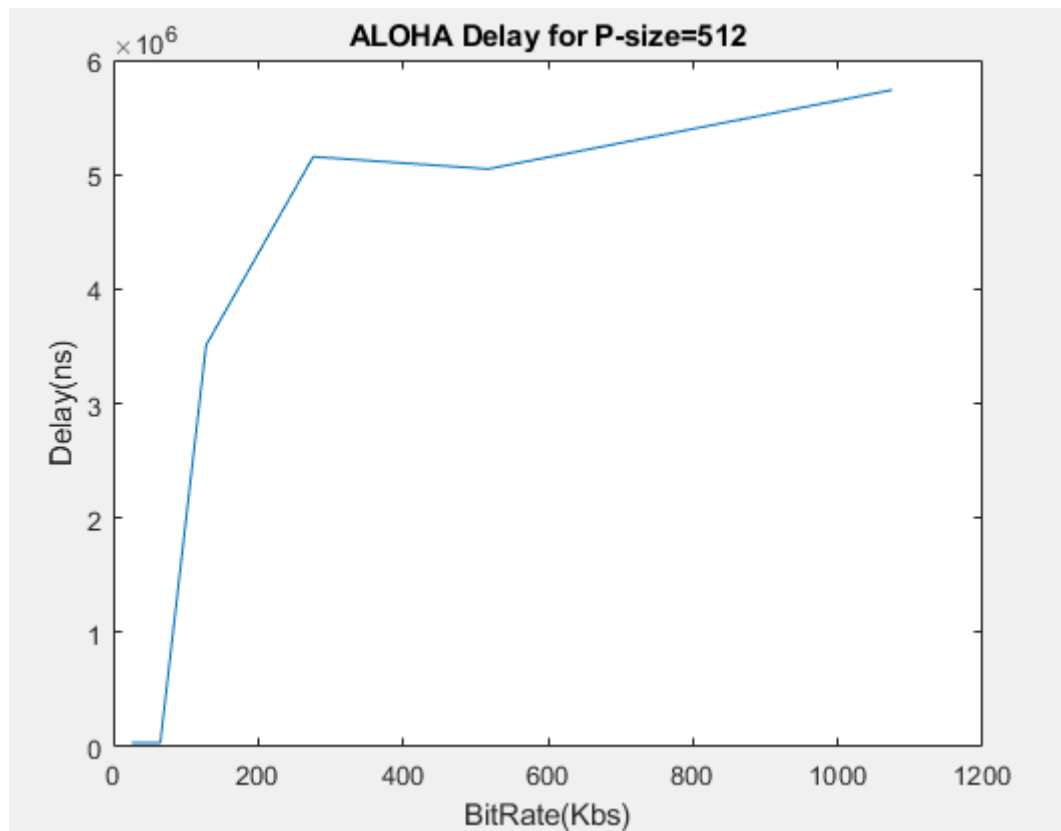
```
else
{
    //
    // The channel is free, transmit the packet
    //
    m_phyTxBeginTrace (m_currentPkt);
    if (m_channel->TransmitStart (m_currentPkt, m_deviceId) == false)
    {
        //NS_LOG_WARN ("Channel TransmitStart returns an error");
        //m_phyTxDropTrace (m_currentPkt);
        //m_currentPkt = 0;
        //m_txMachineState = READY;

        //aloha
        m_txMachineState = BACKOFF;
    }
    if (m_backoff.MaxRetriesReached ())
    {
        //
        // Too many retries, abort transmission of packet
        //
        TransmitAbort ();
    }
    else
    {
        m_macTxBackoffTrace (m_currentPkt);
        m_backoff.IncrNumRetries ();
        Time backoffTime = m_backoff.GetBackoffTime ();

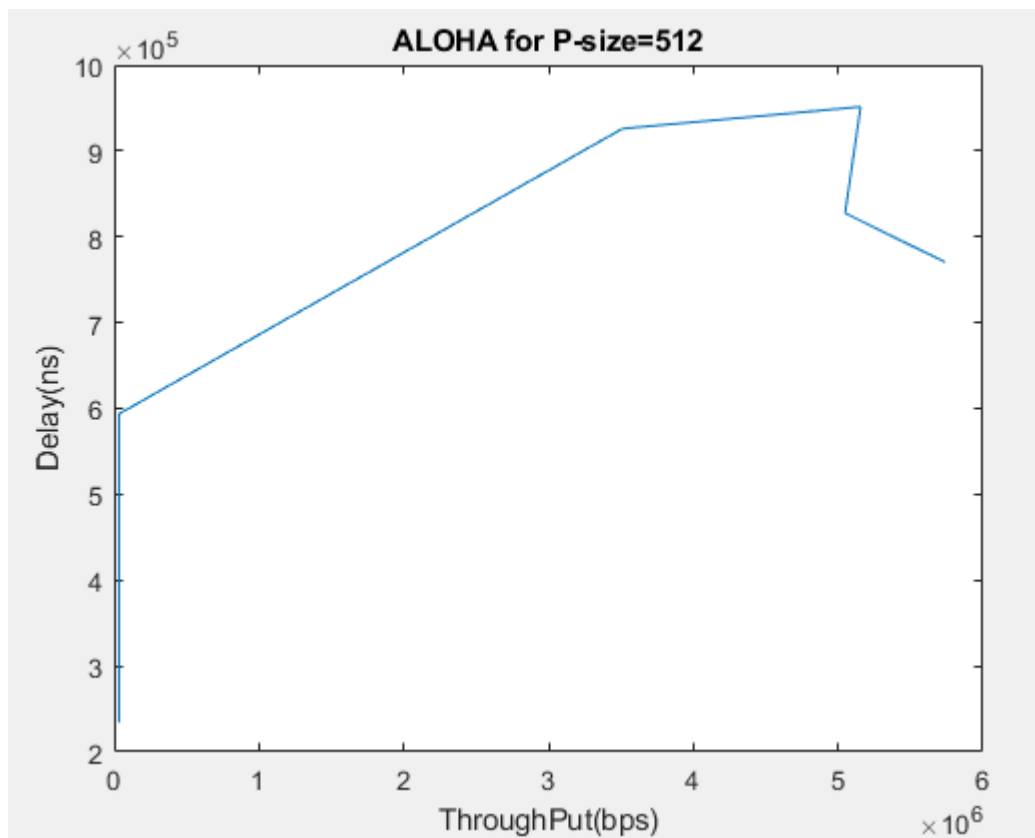
        NS_LOG_LOGIC ("Channel busy, backing off for " << backoffTime.GetSeconds () << " sec");
        Simulator::Schedule (backoffTime, &CsmaNetDevice::TransmitStart, this);
    }
    //aloha
}
```



همان طور که مشاهده می شود، با افزایش ریت کانال، در ابتدا throughput کانال افزایش می یابد و پس از رسیدن به یک ماکزیمم، به صورت نزولی کاهش می یابد.



در این حالت نیز با افزایش ریت کانال، تاخیر به صورت صعودی افزایش می یابد. ولی در این حالت نسبت به حالت قبل چون خاصیت carrier sense را نداریم، متوسط تاخیر بیشتر است.



در نهایت هم نمودار تاخیر براساس throughput به صورت بالاست.



:CSMA P-Persistent

در این قسمت تفاوت به این صورت است که در صورت خلوت بودن کانال، یک عدد random تولید کرده و در صورتی که از p کمتر باشد، پیام را ارسال می کنیم. در غیر این صورت، مانند حالت non-persistent به صورت Random صبر کرده و دوباره اقدام به ارسال پیام می کنیم.

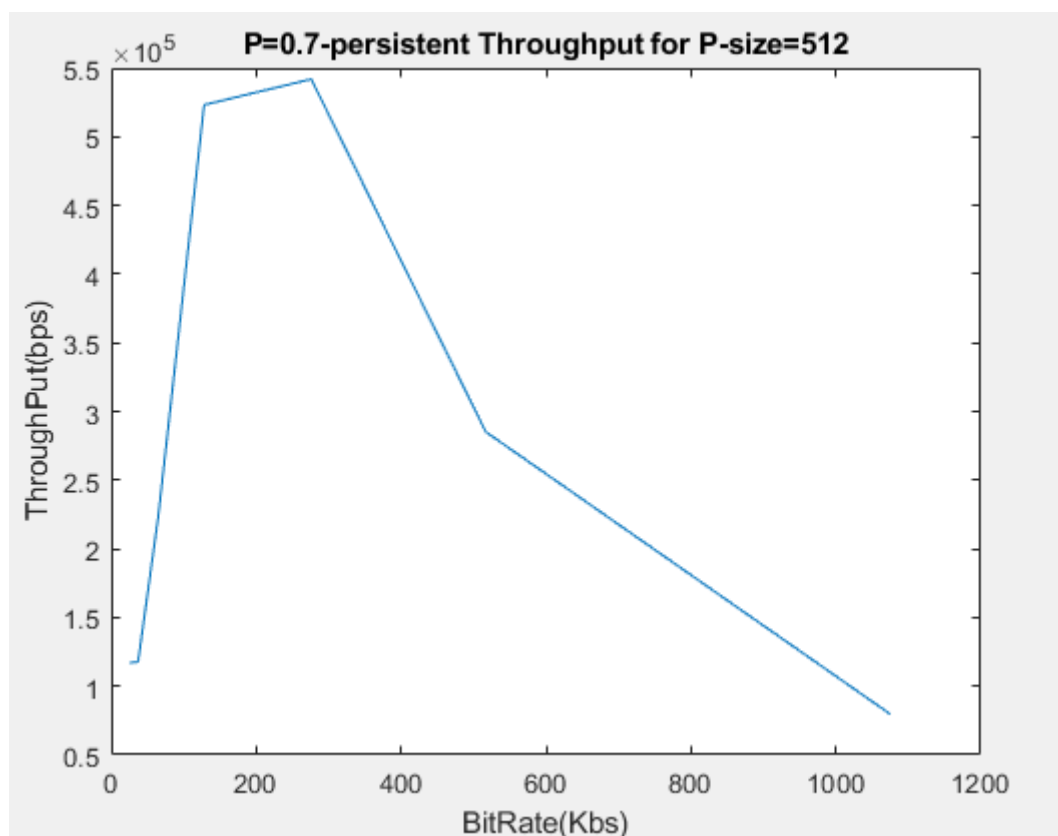
```
double p = 0.5;
double randNum = ((double) rand() / (RAND_MAX));
if (randNum < p)
{
    m_phyTxBeginTrace (m_currentPkt);
    if (m_channel->TransmitStart (m_currentPkt, m_deviceId) == false)
    {
        NS_LOG_WARN ("Channel TransmitStart returns an error");
        m_phyTxDropTrace (m_currentPkt);
        m_currentPkt = 0;
        m_txMachineState = READY;
    }
    else
    {
        //
        // Transmission succeeded, reset the backoff time parameters and
        // schedule a transmit complete event.
        //
        m_backoff.ResetBackoffTime ();
        m_txMachineState = BUSY;

        Time tEvent = m_bps.CalculateBytesTxTime (m_currentPkt->GetSize ());
        NS_LOG_LOGIC ("Schedule TransmitCompleteEvent in " << tEvent.GetSeconds () << "sec");
        Simulator::Schedule (tEvent, &CsmaNetDevice::TransmitCompleteEvent, this);
    }
}
else
{
    m_phyTxDropTrace (m_currentPkt);
    m_currentPkt = 0;
    m_txMachineState = READY;
}
```

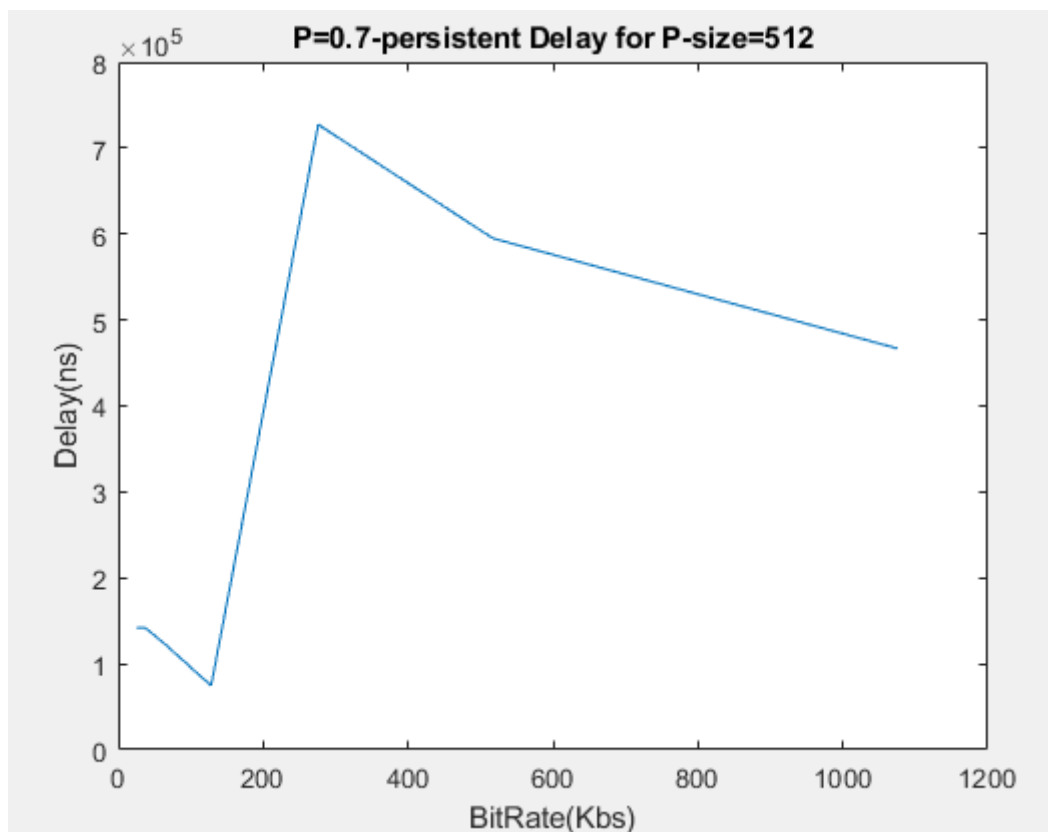
در این قسمت متاسفانه با p های کم به نتایج مورد انتظار نمی رسیدیم. با بررسی backoff نود ها متوجه شدم که Collision اتفاق نمی افتد. علت آن احتمالا به دلیل تاخیر کم در شبکه است.



بنابراین تصمیم گرفتیم که حالت $(1-P)$ را بررسی کنیم. خروجی های زیر برای حالت $P=0.7$ رسم شده اند.



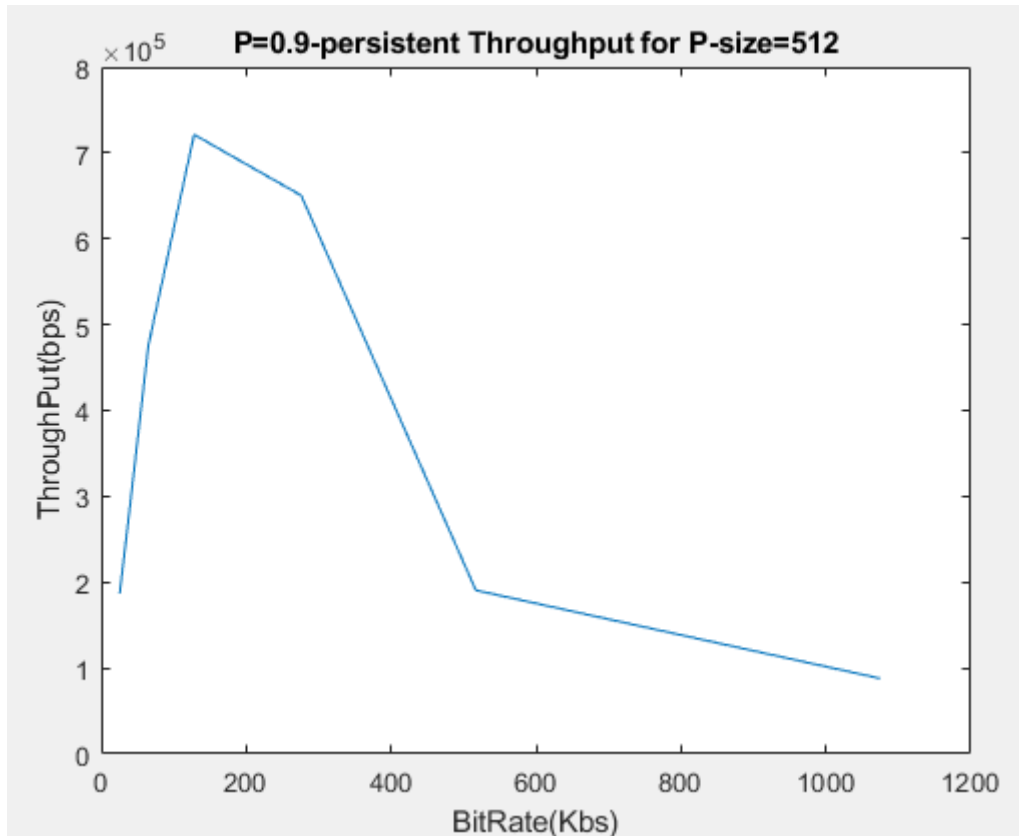
مشاهده می شود که با افزایش ریت کانال، throughput ابتدا افزایش یافته و سپس شروع به کاهش می کند.



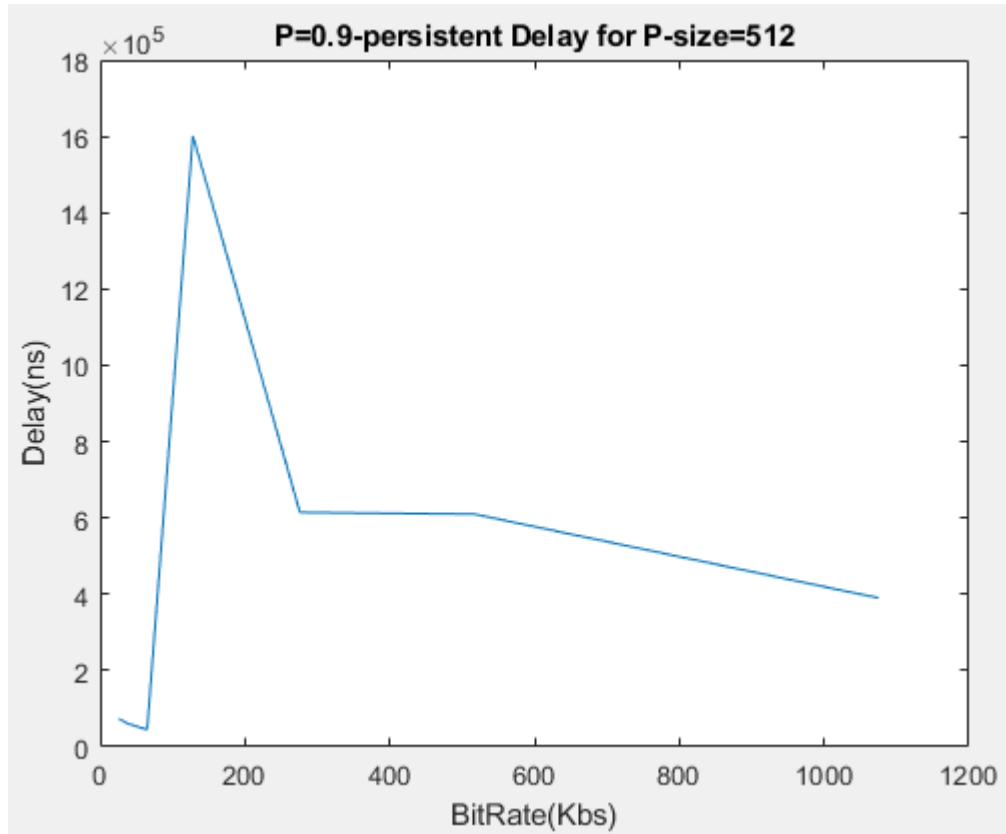
با افزایش ریت کانال، تاخیر در ابتدا افزایش و سپس کاهش می یابد.



برای حالت $P=0.9$ نیز خروجی ها به صورت زیر است.



در این حالت نیز با افزایش ریت کانال، در ابتدا شاهد افزایش throughput بوده و سپس می بینیم که روند نزولی را طی می کند. به طور کلی در حالت حدی انتظار داریم که با زیاد شدن P ، شکل نمودارها به شکل حالت ALOHA نزدیک شده و با کاهش P ، نمودارها به حالت non-persistent میل می کنند.



در نهایت هم می بینیم که برای ریت های بالا، تاخیر روند کاهشی را دنبال می کند.