

به نام خدا

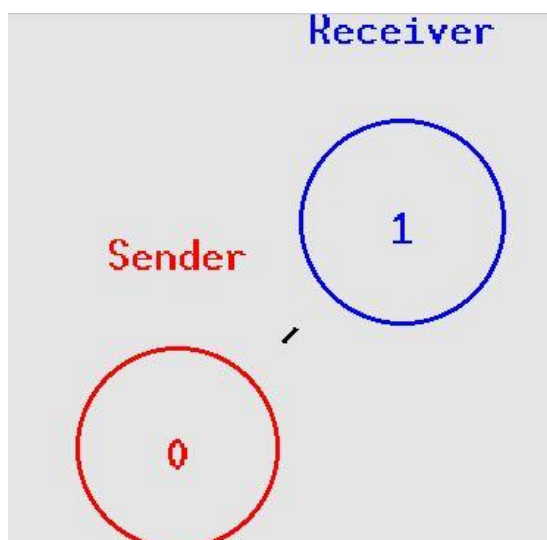
شبکه مخابرات داده

تمرین ۱

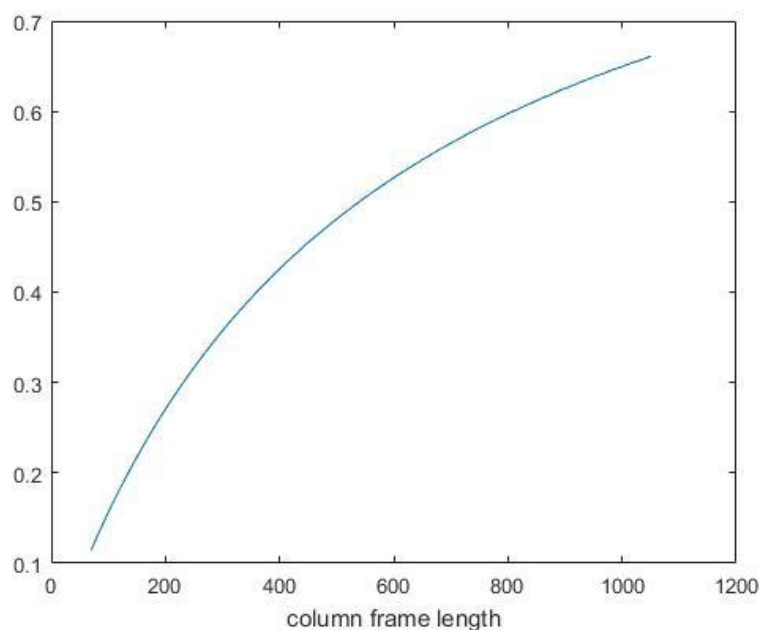
یاسمن حقیقی

۹۴۱۰۶۱۵۷

۱_در ابتدا کد صورت سوال پیاده سازی شده است خروجی فایل nam. مطابق شکل زیر است:



برای رسم نتایج در متلب با استفاده از `import data` در Home به صورت Numeric Matrix به workspace اضافه کرده و پس از تقسیم ستون دوم بر ۲MB نمودار زیر را با دستور `plot` متلب رسم میکنیم:



با افزایش طول فریم مطابق فرمول میزان استفاده از کانال افزایش می یابد و به دنبال آن U افزایش می یابد.

کد قسمت اول:

```
set ns [new Simulator]
set namf [open out.nam w]
set tracef [open trace.tr w]
$ns namtrace-all $namf
$ns trace-all $tracef
set bandwidth [open part\1.txt w]

proc finish {} {
    global ns namf bandwidth tracef
    #define global variables for this procedure
    $ns flush-trace
    close $bandwidth
    close $tracef
    close $namf
    exec nam out.nam
    exit 0
}

$ns namtrace-all $namf
$ns trace-all $tracef
set n(.) [$ns node]
set n(1) [$ns node]
$n(.) color "red"
$n(1) color "blue"
$ns at 0.0 "$n(.) label Sender"
$ns at 0.0 "$n(1) label Receiver"

set LineBW 1Mb
set LineDelay 1ms

$ns duplex-link $n(.) $n(1) $LineBW $LineDelay DropTail
$ns duplex-link-op $n(.) $n(1) orient right-up

set tcp [new Agent/TCP]
$ns attach-agent $n(.) $tcp
$tcp set window_ 1
$tcp set maxcwnd_ 1
$tcp set maxseq_ 500000000

set ftp [new Application/FTP]
$ftp attach-agent $tcp
set tcpsnk [new Agent/TCPSink]
$ns attach-agent $n(1) $tcpsnk
$ns connect $tcp $tcpsnk
$ns color 1 Blue

$tcp set fid_ 1
set headersize 40
set framelen 30
set timeinterval 100

set firstacknum [$tcp set ack_]
set lastacknum [$tcp set ack_]
$tcp set packetSize_ $framelen
proc record {} {
    global bandwidth framelen tcp ns firstacknum lastacknum headersize timeinterval n
    #get the ack number of the current frame
    set lastacknum [$tcp set ack_]
    #compares this ACK number with the one from 10 second before in order to determine
    #transmitter bandwidth
    set bw [expr ( $lastacknum - $firstacknum + 1)*($framelen + $headersize )*8 ]
    set now [$ns now]
    $tcp set packetSize_ $framelen
    #fill the text file:
    puts $bandwidth "[expr ($framelen + $headersize)] [expr (($bw)/($timeinterval))]"
}
```

```

$now* "
#record this ack number for next loop:
set firstacknum [$tcp set ack_]
#record this ack number for next loop:
set firstacknum $lastacknum
#increasing the frame length for next packet:
set framelen [expr ($framelen+10)]
$tcp set packetSize_ $framelen
#execute this proc every 10 second (timeinterval=10)
$ns at [expr $now+$timeinterval] "record"
}
$ns at 10. "ftp start"
$ns at 11. "record"
$ns at 1000. "finish"
$ns run

```

نمونه ای از فایل trace.

```

- 1 0 1 tcp 40 ----- 1 0.0 1.0 0 0
r 1.00116 0 1 tcp 40 ----- 1 0.0 1.0 0 0
+ 1.00116 1 0 ack 40 ----- 1 1.0 0.0 0 1
- 1.00116 1 0 ack 40 ----- 1 1.0 0.0 0 1
r 1.00232 1 0 ack 40 ----- 1 1.0 0.0 0 1
+ 1.00232 0 1 tcp 70 ----- 1 0.0 1.0 1 2
- 1.00232 0 1 tcp 70 ----- 1 0.0 1.0 1 2
r 1.0036 0 1 tcp 70 ----- 1 0.0 1.0 1 2
+ 1.0036 1 0 ack 40 ----- 1 1.0 0.0 1 3
- 1.0036 1 0 ack 40 ----- 1 1.0 0.0 1 3
r 1.00476 1 0 ack 40 ----- 1 1.0 0.0 1 3
+ 1.00476 0 1 tcp 70 ----- 1 0.0 1.0 2 4
- 1.00476 0 1 tcp 70 ----- 1 0.0 1.0 2 4
r 1.00604 0 1 tcp 70 ----- 1 0.0 1.0 2 4
+ 1.00604 1 0 ack 40 ----- 1 1.0 0.0 2 5
- 1.00604 1 0 ack 40 ----- 1 1.0 0.0 2 5
r 1.0072 1 0 ack 40 ----- 1 1.0 0.0 2 5
+ 1.0072 0 1 tcp 70 ----- 1 0.0 1.0 3 6
- 1.0072 0 1 tcp 70 ----- 1 0.0 1.0 3 6
r 1.00848 0 1 tcp 70 ----- 1 0.0 1.0 3 6
+ 1.00848 1 0 ack 40 ----- 1 1.0 0.0 3 7
- 1.00848 1 0 ack 40 ----- 1 1.0 0.0 3 7
r 1.00964 1 0 ack 40 ----- 1 1.0 0.0 3 7
+ 1.00964 0 1 tcp 70 ----- 1 0.0 1.0 4 8
- 1.00964 0 1 tcp 70 ----- 1 0.0 1.0 4 8
r 1.01092 0 1 tcp 70 ----- 1 0.0 1.0 4 8
+ 1.01092 1 0 ack 40 ----- 1 1.0 0.0 4 9
- 1.01092 1 0 ack 40 ----- 1 1.0 0.0 4 9
r 1.01208 1 0 ack 40 ----- 1 1.0 0.0 4 9
+ 1.01208 0 1 tcp 70 ----- 1 0.0 1.0 5 10

```

برای قسمت دوم تغییراتی در کد ایجاد کردم:

با توجه به شماره دانشجویی مقدار اولیه اینگونه تغییر دادم:

```
set LineDelay ۰,۰۰۶
```

بعد از تعریف duplex-link ، دستور زیر اضافه شده است:

```
set myLink [[ $\$ns$  link  $\$n(۰)$   $\$n(۱)$ ] link]
```

سپس به متغیرهای global متغیر myLink,LinkBW,LineDelay اضافه شده و در فایل txt. اطلاعات به فرم زیر نوشته میشود:

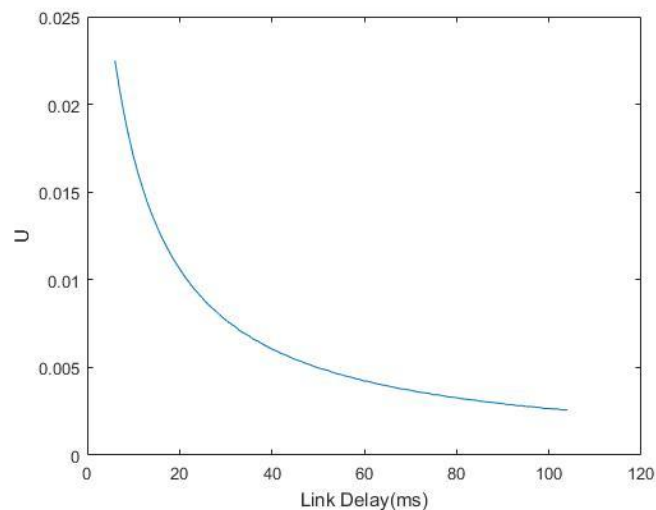
```
puts  $\$bandwidth$  "  $\$LineDelay$  [expr (( $\$bw$ )/( $\$timeinterval*\$LineBW$ ))] "
```

سپس مقدار LineDelay مطابق زیر تغییر دادم:

```
set LineDelay [expr ( $\$LineDelay$  + ۰,۰۰۹۵)]  
 $\$myLink$  set delay_  $\$LineDelay$ 
```

و تغییرات فریم را از کد حذف کردم.

سپس همانطور که در قسمت قبل توضیح دادم نمودار مربوطه را رسم میکنیم:



مشاهده میشود که با افزایش تاخیر میزان U کاهش می یابد چون در بازه زمانی مشخص کمتر از کانال استفاده میشود و اگر تاخیر کانال زیاد باشد استفاده از Stop and Wait مناسب نمیشود.

برای قسمت سوم مطابق زیر عمل کردم:

با توجه به شماره دانشجویی مقادیر اولیه کد قسمت قبل را مطابق زیر تغییر دادم:

```
set LineBW ۰.۰۰۰۰۰۰  
set LineDelay ۰,۰.۰۱
```

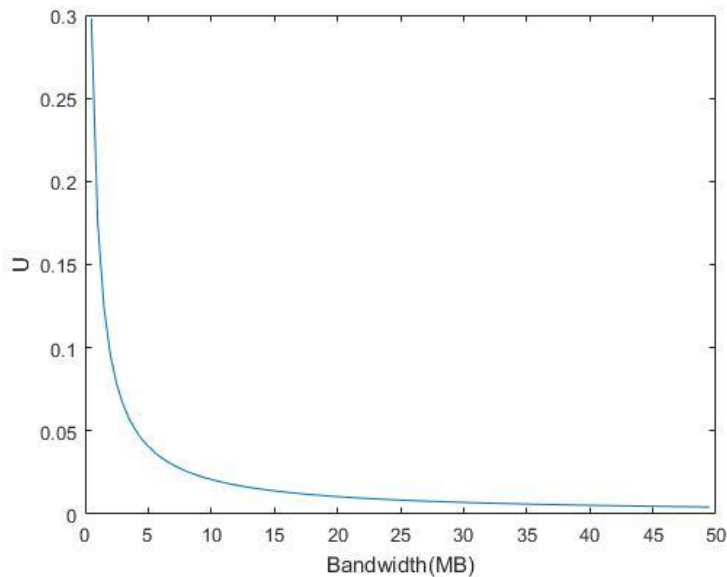
در فایل `txt`. اطلاعات به فرم زیر نوشته میشود:

```
puts $bandwidth " $LineBW [expr (($bw)/($timeinterval*$LineBW))] "
```

سپس مطابق زیر پهنای باند را تغییر دادم و تغییرات مربوط به تاخیر را حذف کردم:

```
set LineBW [expr ($LineBW + ۰.۰۰۰۰۰۰)]  
$myLink set bandwidth_ $LineBW
```

سپس همانطور که توضیح داده شد نمودار را رسم کردم:



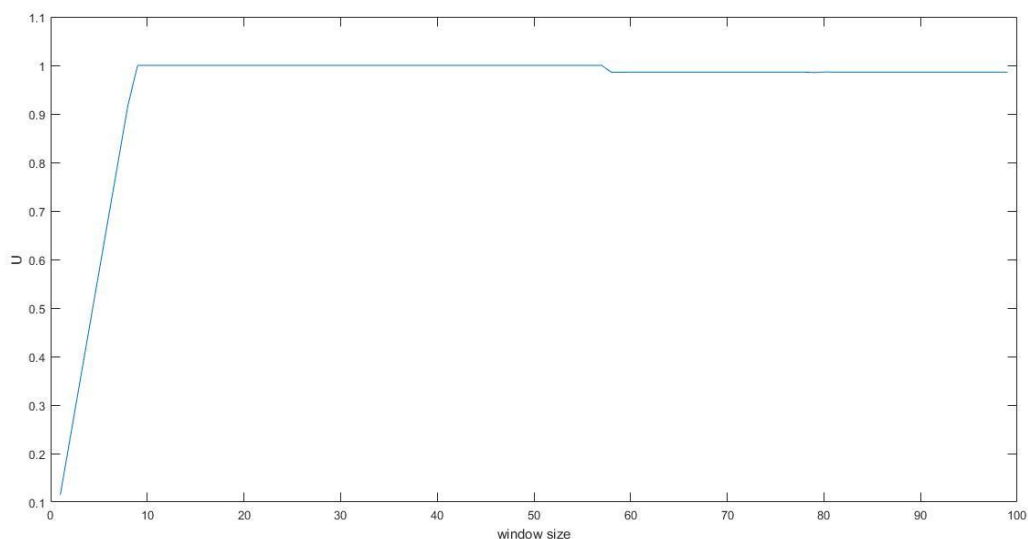
همان طور که از فرمول هم انتظار میرفت (چون پهنای باند در مخرج است) با افزایش پهنای باند U کاهش می یابد چون میزان استفاده از کانال تغییری نکرده ولی ظرفیت کانال بیشتر هم شده است.

نتیجه گیری:

با توجه به تعریف U اگر میزان استفاده از کانال افزایش یابد U هم افزایش می یابد و برعکس. همانطور که مشاهده شد با افزایش طول فریم این مقدار افزایش یافته و به دنبال آن U هم افزایش می یابد. همچنین اگر تاخیر افزایش یابد این میزان کاهش می یابد.

و اگر پهنای باند کانال افزایش یابد چون در مخرج U قرار دارد، مقدار U را کاهش می دهد.

در ابتدا کد را مانند آنچه در صورت سوال گفته شده تغییر دادم نتیجه با استفاده از داده های سوال ۱ به فرم زیر است:



هدف نهایی این است که کانال همیشه پر باشد برای این کار باید مدت زمانی که طول میکشد تا Ack مربوط به پنجره اول برسد برابر با مدت زمانی باشد که طول میکشد تا Window Size-۱ ارسال شود یعنی:

$$2 * \text{Line Delay} + \text{Ack}/\text{BW} = (\text{Window Size} - 1) * \text{Frame Length}/\text{BW}$$

با توجه به فایل nam. متوجه میشویم مقدار Frame Length Window برابر با ۷۰ بایت و مقدار Ack برابر با ۴۰ بایت است پس:

اگر مقدار تاخیر برابر با ۱ میلی ثانیه و پهنای باند برابر با ۲Mb باشد حداقل مقدار صحیح برابر طول پنجره برابر با ۹ است که در نمودار نیز مشاهده میشود.

پس میتوان نتیجه گرفت که متناسب با پهنای باند و تاخیر میتوان Window size را تعیین کرد.

کد مربوط به این سوال:

```
set ns [new Simulator]

set namf [open out۲.nam w]

set tracef [open trace۲.tr w]

$ns namtrace-all $namf

$ns trace-all $tracef

set bandwidth [open part۲.txt w]

proc finish {} {

    global ns namf bandwidth tracef

    #define global variables for this procedure

    $ns flush-trace

    close $bandwidth

    close $tracef

    close $namf

    exec nam out.nam

    exit .

}

$ns namtrace-all $namf

$ns trace-all $tracef

set n(.) [$ns node]

set n(۱) [$ns node]

$n(.) color "red"

$n(۱) color "blue"

$ns at ., "$n(.) label Sender"

$ns at ., "$n(۱) label Receiver"

set LineBW ۲Mb

set LineDelay ۱ms

$ns duplex-link $n(.) $n(۱) $LineBW $LineDelay DropTail

$ns duplex-link-op $n(.) $n(۱) orient right-up
```

```

set tcp [new Agent/TCP]

$ns attach-agent $n(0) $tcp

$tcp set window_ 1

$tcp set maxcwnd_ 1

$tcp set maxseq_ 0.....

set ftp [new Application/FTP]

$ftp attach-agent $tcp

set tcpsnk [new Agent/TCPSink]

$ns attach-agent $n(1) $tcpsnk

$ns connect $tcp $tcpsnk

$ns color 1 Blue

$tcp set fid_ 1

set headersize 40

set framelen 30

set timeinterval 100

set firstacknum [$tcp set ack_]

set lastacknum [$tcp set ack_]

$tcp set packetSize_ $framelen

proc record {} {

global bandwidth framelen tcp ns firstacknum lastacknum headersize timeinterval n

#get the ack number of the current frame

set lastacknum [$tcp set ack_]

#compares this ACK number with the one from 10 second before in order to determine

#transmitter bandwidth

set bw [expr ( $lastacknum - $firstacknum +1)*($framelen + $headersize )*8 ]

set now [$ns now]

set windowsize [$tcp set window_]

set maxwind [$tcp set maxcwnd_]

$tcp set packetSize_ $framelen

#fill the text file:

puts $bandwidth "$windowsize [expr (($bw)/($timeinterval))]"

```

```

#record this ack number for next loop:

set firstacknum [$tcp set ack_]

#record this ack number for next loop:

set firstacknum $lastacknum

#increasing the window_ and maxcwnd_ for next packet:

$tcp set window_ [expr ($window_+$windowincr)]

$tcp set maxcwnd_ [expr ($maxcwnd_+$windowincr)]

#$tcp set packetSize_ $frameLen

#execute this proc every 10 second (timeinterval=10)

$ns at [expr $now+$timeinterval] "record"

}

$ns at 10. "ftp start"

$ns at 11. "record"

$ns at 100. "finish"

$ns run

```

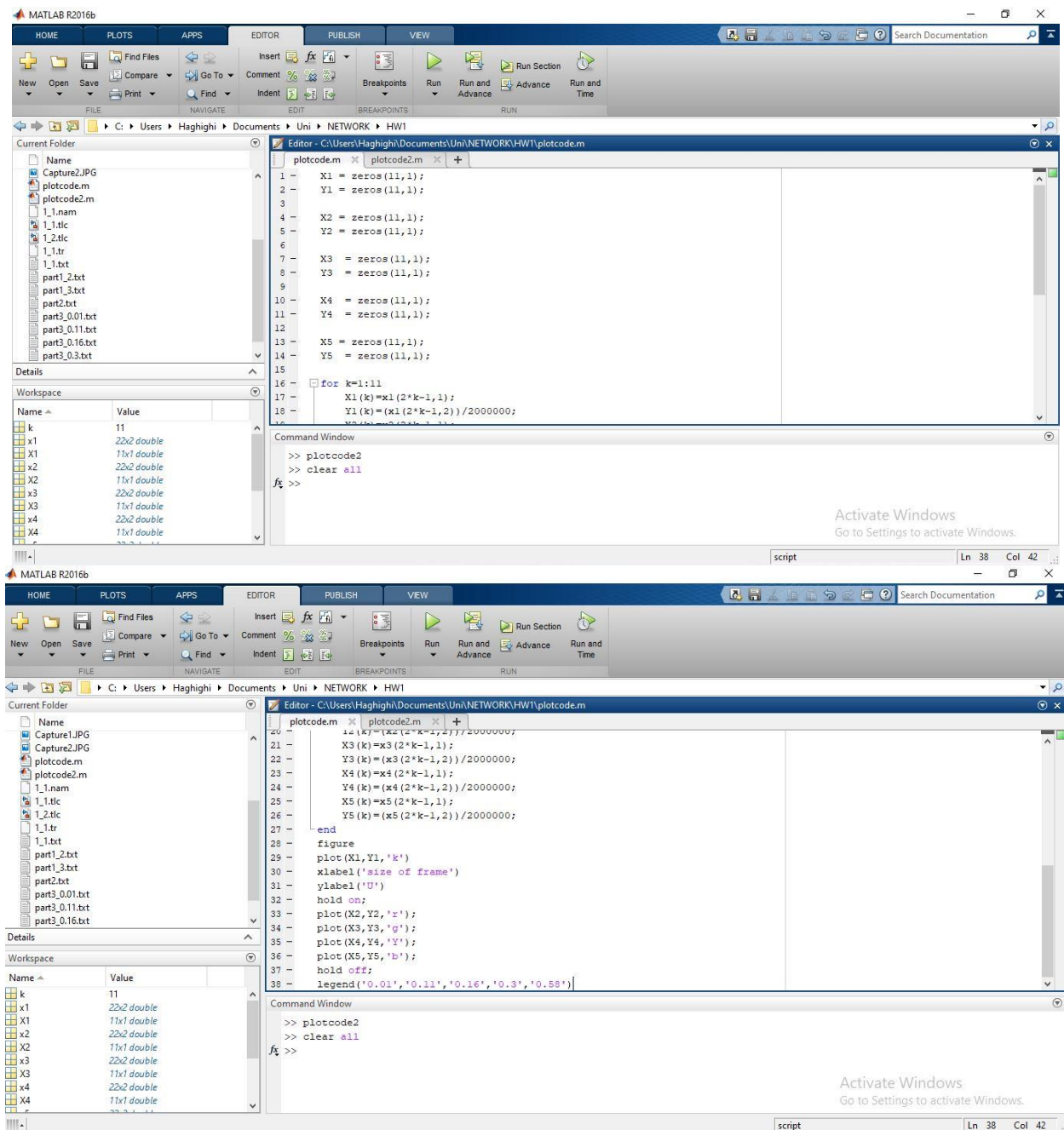
نمونه ای از فایل .trace

```

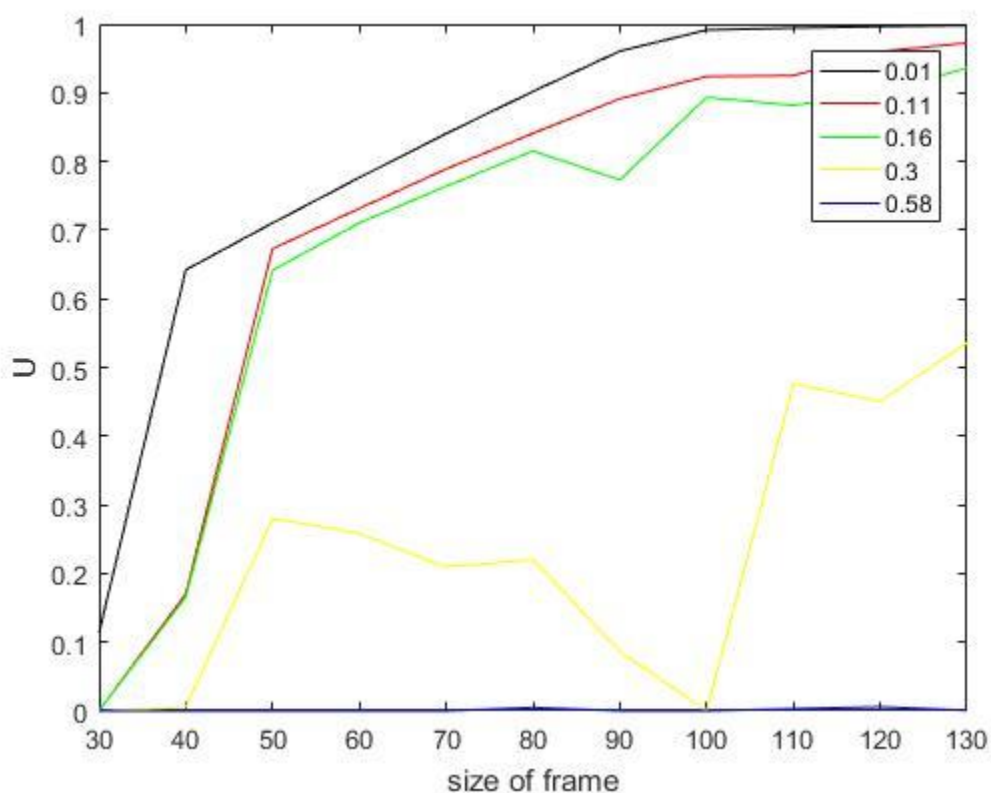
+ 1 0 1 tcp 40 ----- 1 0.0 1.0 0 0
- 1 0 1 tcp 40 ----- 1 0.0 1.0 0 0
r 1.00116 0 1 tcp 40 ----- 1 0.0 1.0 0 0
+ 1.00116 1 0 ack 40 ----- 1 1.0 0.0 0 1
- 1.00116 1 0 ack 40 ----- 1 1.0 0.0 0 1
r 1.00232 1 0 ack 40 ----- 1 1.0 0.0 0 1
+ 1.00232 0 1 tcp 70 ----- 1 0.0 1.0 1 2
- 1.00232 0 1 tcp 70 ----- 1 0.0 1.0 1 2
r 1.0036 0 1 tcp 70 ----- 1 0.0 1.0 1 2
+ 1.0036 1 0 ack 40 ----- 1 1.0 0.0 1 3
- 1.0036 1 0 ack 40 ----- 1 1.0 0.0 1 3
r 1.00476 1 0 ack 40 ----- 1 1.0 0.0 1 3
+ 1.00476 0 1 tcp 70 ----- 1 0.0 1.0 2 4
- 1.00476 0 1 tcp 70 ----- 1 0.0 1.0 2 4
r 1.00604 0 1 tcp 70 ----- 1 0.0 1.0 2 4
+ 1.00604 1 0 ack 40 ----- 1 1.0 0.0 2 5
- 1.00604 1 0 ack 40 ----- 1 1.0 0.0 2 5
r 1.0072 1 0 ack 40 ----- 1 1.0 0.0 2 5
+ 1.0072 0 1 tcp 70 ----- 1 0.0 1.0 3 6
- 1.0072 0 1 tcp 70 ----- 1 0.0 1.0 3 6
r 1.00848 0 1 tcp 70 ----- 1 0.0 1.0 3 6
+ 1.00848 1 0 ack 40 ----- 1 1.0 0.0 3 7
- 1.00848 1 0 ack 40 ----- 1 1.0 0.0 3 7
r 1.00964 1 0 ack 40 ----- 1 1.0 0.0 3 7
+ 1.00964 0 1 tcp 70 ----- 1 0.0 1.0 4 8
- 1.00964 0 1 tcp 70 ----- 1 0.0 1.0 4 8

```

۳_ ابتدا کد را مطابق آنچه در سوال گفته شده تغییر می دهیم و ۵ بار که در هر بار مقدار drop rate را تغییر می دهیم اجرا میکنیم و نمودار را مطابق زیر رسم میکنیم:



نمودار به فرم زیر است:



همان طور که مشاهده میشود با افزایش مقدار drop rate مقدار U به شدت کاهش می یابد تا حدی که برای حالتی که برابر با ۰,۵۸ است تقریباً $U=0$ میشود و کانال کاربردی نیست. که این نتیجه با فرمول ارائه در اسلاید ها نیز مطابقت دارد.

کد این سوال:

قسمت قرمز شده را تغییر میدهیم

```
set ns [new Simulator]

set namf [open out۳.nam w]

set tracef [open trace۳.tr w]

$ns namtrace-all $namf

$ns trace-all $tracef

set bandwidth [open part۳_۰,۱۱.txt w]

proc finish {} {

    global ns namf bandwidth tracef

    #define global variables for this procedure

    $ns flush-trace

    close $bandwidth

    close $tracef

    close $namf

    exec nam out.nam

    exit .

}

$ns namtrace-all $namf

$ns trace-all $tracef

set n(.) [$ns node]

set n(۱) [$ns node]

$n(.) color "red"

$n(۱) color "blue"

$ns at .۰ "$n(.) label Sender"

$ns at .۰ "$n(۱) label Receiver"

set LineBW ۲Mb

set LineDelay ۱ms

$ns duplex-link $n(.) $n(۱) $LineBW $LineDelay DropTail
```

```

$ns duplex-link-op $n(.) $n(1) orient right-up

set tcp [new Agent/TCP]

$ns attach-agent $n(.) $tcp

$tcp set window_ 1

$tcp set maxcwnd_ 1

$tcp set maxseq_ 65535

set ftp [new Application/FTP]

$ftp attach-agent $tcp

set tcpsnk [new Agent/TCPSink]

$ns attach-agent $n(1) $tcpsnk

$ns connect $tcp $tcpsnk

$ns color 1 Blue

$tcp set fid_ 1

#Loss parameters

set loss_module [new ErrorModel]

#set Drop rate

$loss_module set rate_ .1

$loss_module set min_ .

$loss_module set max_ 1

#Uniform random variable (means choosing a packet for dropping is uniform in #time):

$loss_module ranvar [new RandomVariable/Uniform]

$loss_module drop-target [new Agent/Null]

#a complete packet should be dropped not bytes or bits , etc:

$loss_module unit pkt

#Loss module starts working at "time = .1 (s)" between node . and 1:

$ns at .1 "$ns lossmodel $loss_module $n(1) $n(.)"

#Constant Window Size For Lossy Channel

proc windconst {} {

global tcp ns file\

```

```

$tcp set maxcwnd_ Δ

# maximum congestion window

$tcp set windowinit_ Δ

# initial window size

$tcp set window_ Δ

# window size

$tcp set cwnd_ Δ

#congestion window

set now [$ns now]

$ns at [expr $now+.,\] "windconst"

}


set headersize f.

set framelen r.

set timeinterval \.,

set firstacknum [$tcp set ack_]

set lastacknum [$tcp set ack_]

$tcp set packetSize_ $framelen

proc record {} {

global bandwidth framelen tcp ns firstacknum lastacknum headersize timeinterval n

#get the ack number of the current frame

set lastacknum [$tcp set ack_]

#compares this ACK number with the one from \. second before in order to determine

#transmitter bandwidth

set bw [expr ( $lastacknum - $firstacknum +1)*($framelen + $headersize )*\^ ]

set now [$ns now]

set windowSize [$tcp set window_]

set maxwind [$tcp set maxcwnd_]

$tcp set packetSize_ $framelen

#fill the text file:

puts $bandwidth "$framelen [expr (($bw)/($timeinterval))]"

$now*"

#record this ack number for next loop:

```



```

set firstacknum [$tcp set ack_]

#record this ack number for next loop:

set firstacknum $lastacknum

#increasing the window_ and maxcwnd_ for next packet:

#$tcp set window_ [expr ($window_size+1)]

#$tcp set maxcwnd_ [expr ($maxwind+1)]

set framelen [expr ($framelen+10) ]

$tcp set packetSize_ $framelen

#execute this proc every 10 second (timeinterval=10)

$ns at [expr $now+$timeinterval] "record"

}

$ns at 1. "sftp start"

$ns at 1.1 "windconst"

$ns at 2. "record"

$ns at 10. "finish"

$ns run

```

نمونه ای از فایل .trace

```

+ 1 0 1 tcp 40 ----- 1 0.0 1.0 0 0
- 1 0 1 tcp 40 ----- 1 0.0 1.0 0 0
r 1.00116 0 1 tcp 40 ----- 1 0.0 1.0 0 0
d 1.00116 1 0 ack 40 ----- 1 1.0 0.0 0 1
+ 4 0 1 tcp 40 ---A--- 1 0.0 1.0 0 2
- 4 0 1 tcp 40 ---A--- 1 0.0 1.0 0 2
r 4.00116 0 1 tcp 40 ---A--- 1 0.0 1.0 0 2
d 4.00116 1 0 ack 40 ----- 1 1.0 0.0 0 3
+ 10 0 1 tcp 40 ---A--- 1 0.0 1.0 0 4
- 10 0 1 tcp 40 ---A--- 1 0.0 1.0 0 4
r 10.00116 0 1 tcp 40 ---A--- 1 0.0 1.0 0 4
+ 10.00116 1 0 ack 40 ----- 1 1.0 0.0 0 5
- 10.00116 1 0 ack 40 ----- 1 1.0 0.0 0 5
r 10.00232 1 0 ack 40 ----- 1 1.0 0.0 0 5
+ 10.00232 0 1 tcp 80 ----- 1 0.0 1.0 1 6
- 10.00232 0 1 tcp 80 ----- 1 0.0 1.0 1 6
r 10.00364 0 1 tcp 80 ----- 1 0.0 1.0 1 6
+ 10.00364 1 0 ack 40 ----- 1 1.0 0.0 1 7
- 10.00364 1 0 ack 40 ----- 1 1.0 0.0 1 7
r 10.0048 1 0 ack 40 ----- 1 1.0 0.0 1 7
+ 10.0048 0 1 tcp 80 ----- 1 0.0 1.0 2 8
- 10.0048 0 1 tcp 80 ----- 1 0.0 1.0 2 8
+ 10.0048 0 1 tcp 80 ----- 1 0.0 1.0 3 9
- 10.00512 0 1 tcp 80 ----- 1 0.0 1.0 3 9
r 10.00612 0 1 tcp 80 ----- 1 0.0 1.0 2 8
+ 10.00612 1 0 ack 40 ----- 1 1.0 0.0 2 10
- 10.00612 1 0 ack 40 ----- 1 1.0 0.0 2 10
r 10.00644 0 1 tcp 80 ----- 1 0.0 1.0 3 9
+ 10.00644 1 0 ack 40 ----- 1 1.0 0.0 3 11

```