



به نام خدا



دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر  
درس شبکه های کامپیوتری

پروژه دوم

نام و نام خانوادگی	دانیال سعیدی (810198571) محمد قره حسنلو
تاریخ ارسال گزارش	یکشنبه - ۷ فروردین ۱۴۰۱

## فهرست گزارش سوالات

۱. کانفیگ ..... 3
۲. تعریف توپولوژی شبکه ..... 3
۳. تعریف تابع send\_message ..... 4
۴. تعریف تابع recv ..... 5
۵. نرخ خطای ارسال ..... 5
۶. اجرا و تولید فایل های خروجی ..... 6
۷. تجزیه و تحلیل فایل های خروجی ..... 7

## ۱. کانفیگ

کانفیگ و تنظیمات هر Node به شرح ذیل است: (مقدار bandwidth\_ متفاوت در سه فایل tcp مقادیر 1.5 و 55 و 155 نسبت داده شده است.)

```
Mac/Simple set bandwidth_ 1.5Mb
set val(prop)      Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)      Phy/WirelessPhy           ;# network interface type
set val(mac)        Mac/802_11                ;# MAC type
set val(ifq)        Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)         LL                        ;# link layer type
set val(ant)        Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)     32768                     ;# max packet in ifq
set val(rp) DumbAgent
$ns node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channelType Channel/WirelessChannel \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace OFF \
                -macTrace ON \
                -movementTrace OFF
```

## ۲. تعریف توپولوژی شبکه

در کد زیر توپولوژی جدید شبکه را تعریف میکنیم:

```
set topo [new Topography]
```

باید بعد از تعریف توپولوژی جدید تعداد گره های شبکه را به create-god بدهیم:

```
create-god 9
```

موقعیت چند گره از شبکه وایلس به شکل ذیل است:

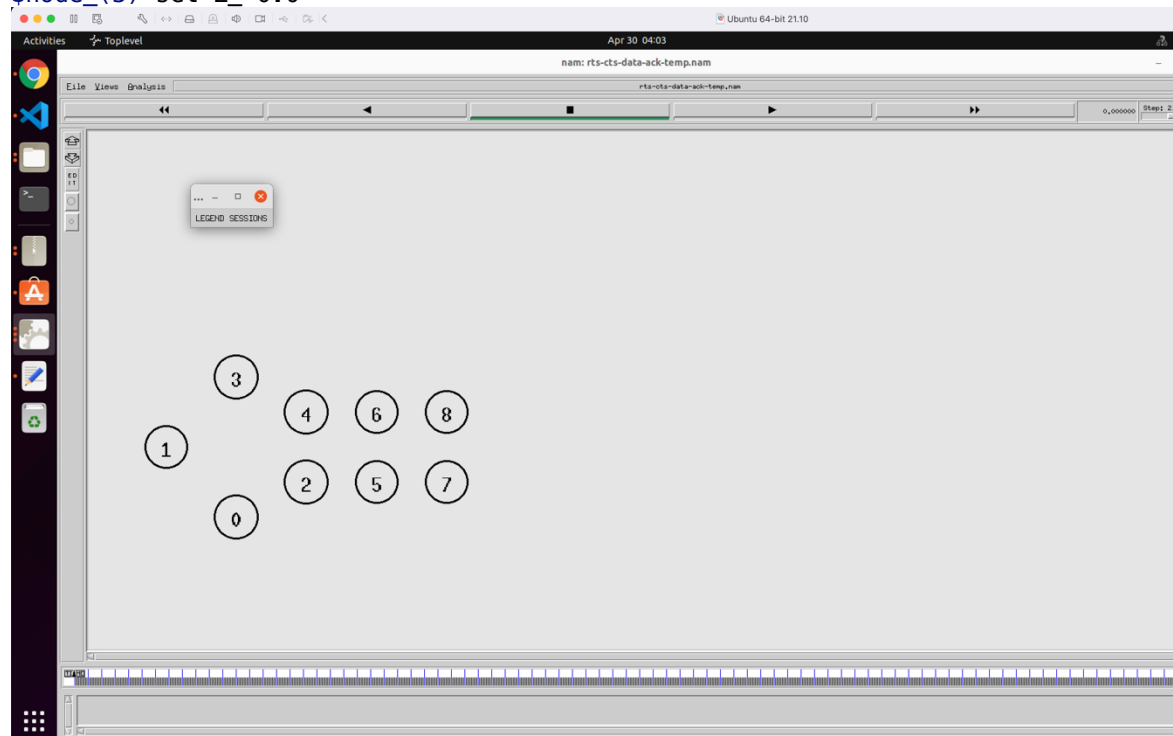
```
# A
$node_(0) set X_ 50.0
$node_(0) set Y_ 0.0
$node_(0) set Z_ 0.0

# B
$node_(1) set X_ 0.0
```

```
$node_(1) set Y_ 50.0
$node_(1) set Z_ 0.0
```

```
#D
```

```
$node_(3) set X_ 50.0
$node_(3) set Y_ 100.0
$node_(3) set Z_ 0.0
```



### ۳. تعریف تابع send\_message

این تابع پارامترهای size، message\_id، data و port را به عنوان ورودی میگیرد و پیام را به گره های مربوطه ارسال میکند.

```
Agent/MessagePassing/Flooding instproc send_message {size message_id data port}
{
  $self instvar messages_seen node_
  global ns MESSAGE_PORT 7
  global ns MESSAGE_PORT_2 8

  lappend messages_seen $message_id
  $ns trace-annotate "[$node_ node-addr] sending message $message_id"
  $self sendto $size "$message_id:$data" $port
}
```

## ۴. تعریف تابع recv

این تابع وظیفه دریافت و ارسال پکت را به عهده دارد. این تابع پارامترهای زیر را میگیرد:

- Source: جایی که پیام از آن ارسال شده است.
- Sport: همان پورت source است.
- Size: اندازه packet
- Data : داده

```
Agent/MessagePassing/Flooding instproc recv {source sport size data} {
    $self instvar messages_seen node_
    global ns 1

    # extract message ID from message
    set message_id [lindex [split $data ":"] 0]

    puts "\nNode [$node_ node-addr] got message $message_id\n"
    if {[lsearch $messages_seen $message_id] == -1} {
        lappend messages_seen $message_id

        $self sendto $size $data 7 $sport
        $self sendto $size $data 8 $sport
    }
}
```

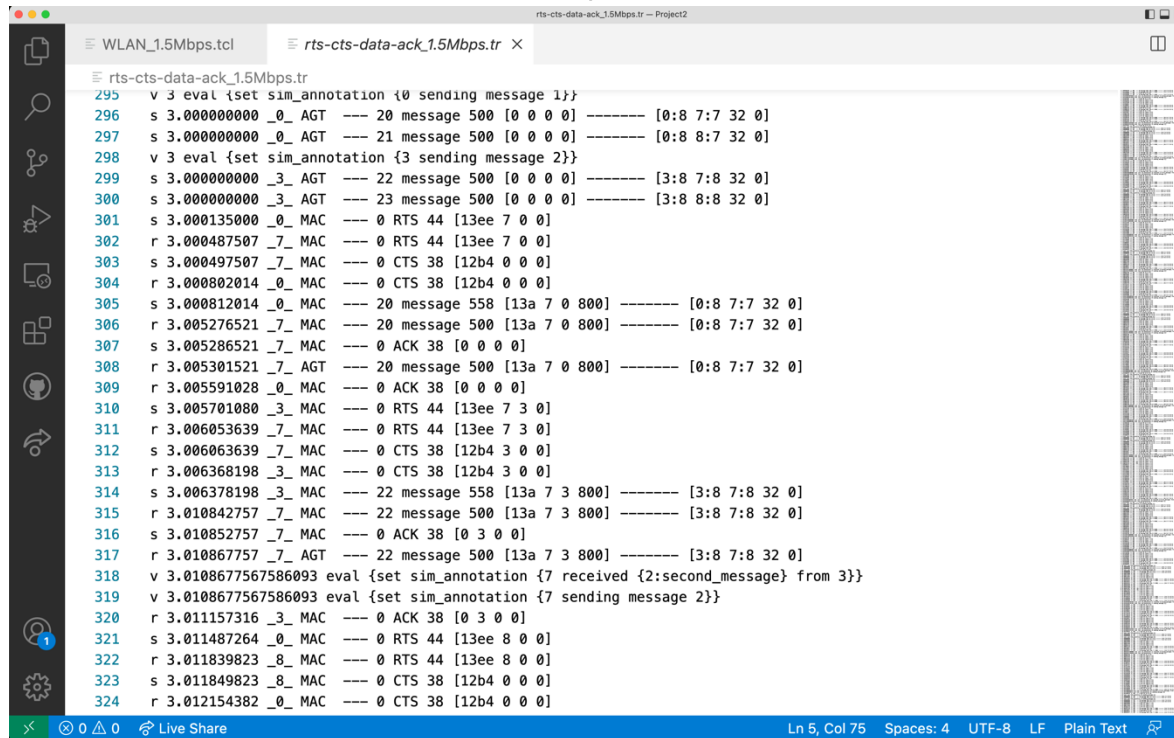
## ۵. نرخ خطای ارسال

این نرخ خطای ارسال بین 0.000001 و 0.00001 تغییر می کند.

```
set loss_module [new ErrorModel]
$loss_module set rate_ 0.000001
# set target for dropped packets
$loss_module drop-target [new Agent/Null]
```

## ۶. اجرا و تولید فایل های خروجی

با دستور ns file.tcl میتوان شبیه سازی را انجام داد و در انتها فایل tr تولید میشود.



```
rts-cts-data-ack_1.5Mbps.tr
295 v 3 eval {set sim_annotation {0 sending message 1}}
296 s 3.000000000 _0_ AGT --- 20 message 500 [0 0 0 0] ----- [0:8 7:7 32 0]
297 s 3.000000000 _0_ AGT --- 21 message 500 [0 0 0 0] ----- [0:8 8:7 32 0]
298 v 3 eval {set sim_annotation {3 sending message 2}}
299 s 3.000000000 _3_ AGT --- 22 message 500 [0 0 0 0] ----- [3:8 7:8 32 0]
300 s 3.000000000 _3_ AGT --- 23 message 500 [0 0 0 0] ----- [3:8 8:8 32 0]
301 s 3.000135000 _0_ MAC --- 0 RTS 44 [13ee 7 0 0]
302 r 3.000487507 _7_ MAC --- 0 RTS 44 [13ee 7 0 0]
303 s 3.000497507 _7_ MAC --- 0 CTS 38 [12b4 0 0 0]
304 r 3.000802014 _0_ MAC --- 0 CTS 38 [12b4 0 0 0]
305 s 3.000812014 _0_ MAC --- 20 message 558 [13a 7 0 800] ----- [0:8 7:7 32 0]
306 r 3.005276521 _7_ MAC --- 20 message 500 [13a 7 0 800] ----- [0:8 7:7 32 0]
307 s 3.005286521 _7_ MAC --- 0 ACK 38 [0 0 0 0]
308 r 3.005301521 _7_ AGT --- 20 message 500 [13a 7 0 800] ----- [0:8 7:7 32 0]
309 r 3.005591028 _0_ MAC --- 0 ACK 38 [0 0 0 0]
310 s 3.005701080 _3_ MAC --- 0 RTS 44 [13ee 7 3 0]
311 r 3.006053639 _7_ MAC --- 0 RTS 44 [13ee 7 3 0]
312 s 3.006063639 _7_ MAC --- 0 CTS 38 [12b4 3 0 0]
313 r 3.006368198 _3_ MAC --- 0 CTS 38 [12b4 3 0 0]
314 s 3.006378198 _3_ MAC --- 22 message 558 [13a 7 3 800] ----- [3:8 7:8 32 0]
315 r 3.010842757 _7_ MAC --- 22 message 500 [13a 7 3 800] ----- [3:8 7:8 32 0]
316 s 3.010852757 _7_ MAC --- 0 ACK 38 [0 3 0 0]
317 r 3.010867757 _7_ AGT --- 22 message 500 [13a 7 3 800] ----- [3:8 7:8 32 0]
318 v 3.0108677567586093 eval {set sim_annotation {7 received {2:second_message} from 3}}
319 v 3.0108677567586093 eval {set sim_annotation {7 sending message 2}}
320 r 3.011157316 _3_ MAC --- 0 ACK 38 [0 3 0 0]
321 s 3.011487264 _0_ MAC --- 0 RTS 44 [13ee 8 0 0]
322 r 3.011839823 _8_ MAC --- 0 RTS 44 [13ee 8 0 0]
323 s 3.011849823 _8_ MAC --- 0 CTS 38 [12b4 0 0 0]
324 r 3.012154382 _0_ MAC --- 0 CTS 38 [12b4 0 0 0]
```

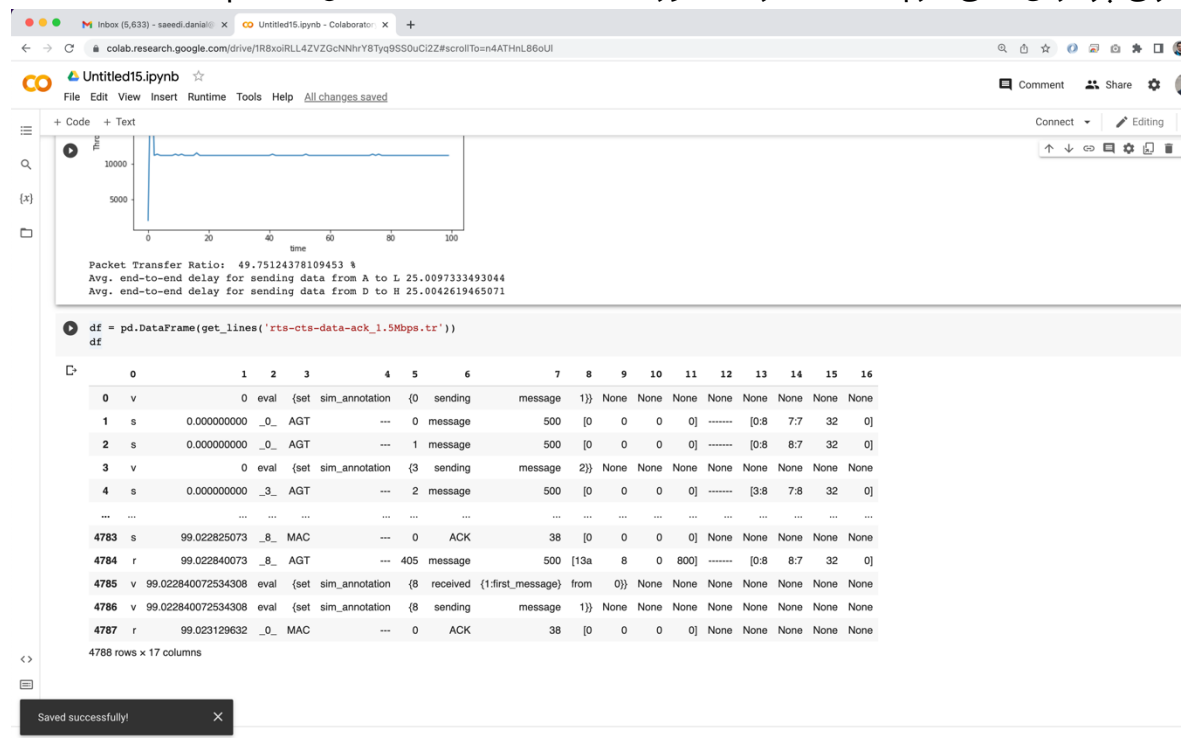
## ۷. تجزیه و تحلیل فایل های خروجی

تجزیه و تحلیل فایل های خروجی به کمک python انجام شده است.

ابتدا فایل های tr. را به صورت خط به خط میخوانیم:

```
def get_lines(filename):  
    lines = []  
    file = open(filename, 'r')  
    line = file.readline()  
    while line:  
        lines.append(line.split())  
        line = file.readline()  
    return lines
```

برای پردازش های لازم، اطلاعات را به صورت یک Data Frame تبدیل میکنیم.



### تابع رسم throughput:

```
def draw_throughput(traces):

    traces[1] = traces[1].astype(float)

    # Packet Size
    traces =
traces[~traces[7].isin(['message', '{1:first_message}', '{2:second_message}', 'redundant'])]
    traces[7] = traces[7].astype(int)

    throughput = []

    for second in range(0,100,1):
        throughput.append(traces[(traces[1] <= second) & (traces[1] >=
second-1)][7].sum())

    plt.figure(figsize=(8,5))
    plt.plot(throughput)
    plt.xlabel("time")
    plt.ylabel("Throughput")
    plt.title("Throughput per second")
    plt.show()
```

### تابع محاسبه Pack Transfer Ratio:

```
def calculate_packet_transfer_ratio(traces):
    pkt_transfer_ratio = traces[6].value_counts()['received']/(traces[6].value_counts()['sending'])
    print("Packet Transfer Ratio: ", pkt_transfer_ratio*100 , '%')
```

### تابع محاسبه Avg. end-to-end delay:

```
def calculate_avg_delay(traces):
    total = 0
    for i in range(0,100):
        received_time = df[(df[6] == 'received') & (df[7] == '{1:first_message}')] [1].iloc[i]
        send_time = df[(df[6] == 'sending') & (df[8] == '1}')] [1].iloc[i]

        total += float(received_time) - float(send_time)

    print("Avg. end-to-end delay for sending data from A to L",total/100)

    total = 0
    for i in range(0,100):
        received_time = df[(df[6] == 'received') & (df[7] == '{2:second_message}')] [1].iloc[i]
        send_time = df[(df[6] == 'sending') & (df[8] == '2}')] [1].iloc[i]

        total += float(received_time) - float(send_time)

    print("Avg. end-to-end delay for sending data from D to H",total/100)
```

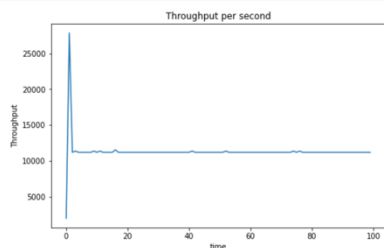


تابع محاسبه `trace_info`: این تابع اطلاعات خواسته شده را برای `trace` داده شده نمایش میدهد.

```
def trace_info(trace):  
    draw_throughput(trace)  
  
    calculate_packet_transfer_ratio(trace)  
  
    calculate_avg_delay(trace)
```

پهنای باند ۱.۵ :

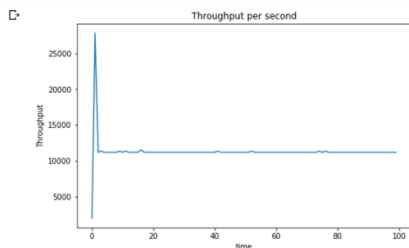
```
[ ] trace_info(pd.DataFrame(get_lines('rts-cts-data-ack_1.5Mbps.tr')))
```



Packet Transfer Ratio: 50.0 %  
Avg. end-to-end delay for sending data from A to L 25.0097333493044  
Avg. end-to-end delay for sending data from D to H 25.0042619465071

پهنای باند ۵۵ :

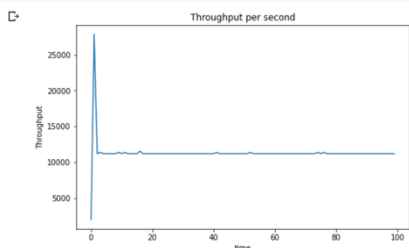
```
trace_info(pd.DataFrame(get_lines('rts-cts-data-ack_55Mbps.tr')))
```



Packet Transfer Ratio: 49.75124378109453 %  
Avg. end-to-end delay for sending data from A to L 25.0097333493044  
Avg. end-to-end delay for sending data from D to H 25.0042619465071

پهنای باند ۱۵۵ :

```
trace_info(pd.DataFrame(get_lines('rts-cts-data-ack_155Mbps.tr')))
```



Packet Transfer Ratio: 49.75124378109453 %  
Avg. end-to-end delay for sending data from A to L 25.0097333493044  
Avg. end-to-end delay for sending data from D to H 25.0042619465071