

گزارش پروژه دوم درس شبکه‌های کامپیوتری (NS2 Simulation)

کیمیا محمدطاهری (۸۱۰۱۹۸۵۳۵)

دانشور امراللهی (۸۱۰۱۹۷۶۸۵)

● تعریف توپولوژی

توپولوژی در نظر گرفته شده در این پروژه مطابق شکل (۱) است که شامل ۹ راس است. در این شبکه، راس صفر به راس هفت با پروتوکول TCP و همچنین راس دو به راس هشت با پروتوکول UDP پکت ارسال می‌کند.



شکل (۱): توپولوژی شبکه

پارامترهای تنظیم شده برای شبکه به صورت زیر هستند:

```
set val(chan)      Channel/WirelessChannel      ;
set val(prop)      Propagation/TwoRayGround     ;
set val(netif)     Phy/WirelessPhy             ;
set val(mac)       Mac/802_11                  ;
set val(ifq)       Queue/DropTail/PriQueue      ;
set val(ll)        LL                          ;
set val(ant)       Antenna/OmniAntenna         ;
set val(ifqlen)    50                          ;
set val(nn)        9                          ;
set val(rp)        AODV                        ;
set val(x)         500                         ;
set val(y)         500                         ;
set val(dr)        1.5e6                       ;#1.5Mbps
```

تنظیمات پروتوکول‌های ارسالی به شکل زیر است:

```
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns attach-agent $n0 $tcp
$ns attach-agent $n7 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"

set udp [new Agent/UDP]
set null [new Agent/Null]
$ns attach-agent $n2 $udp
$ns attach-agent $n8 $null
$ns connect $udp $null
```

```
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
```

● تجزیه و تحلیل فایل‌های خروجی

a. Throughput:

$$\text{Throughput} = \frac{\text{Sum of packet sizes received}}{\text{time}}$$

برای محاسبه throughput از اسکریپت زیر استفاده شده است:

```
import sys

file = open(sys.argv[1], 'r')
lines = file.readlines()

total_time = int(float(lines[-1].split()[1]))
receiver_node = "_" + sys.argv[2] + "_"

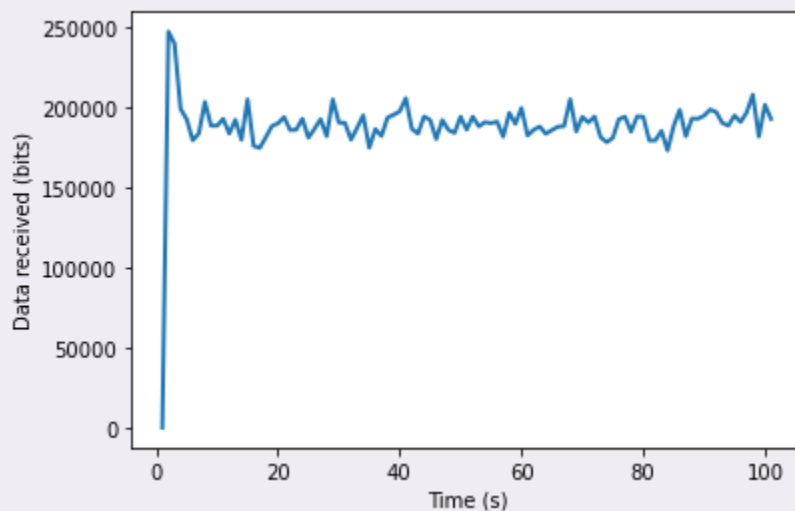
data = 0
for line in lines:
    words = line.split()
    if words[0] == "r" and words[2] == receiver_node:
        data += int(words[7])

print(f"Total Data Received by Node {receiver_node} = {data}")
print(f"Total Time = {total_time}")
print(f"Throughput = {data/total_time} bits/sec")
```

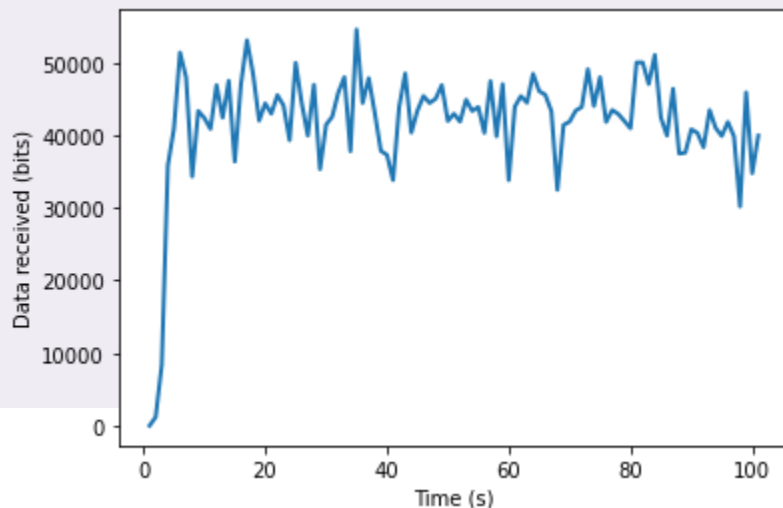
در صورتی که ستون صفر نشان‌دهنده receive باشد و ستون دوم راس دریافت‌کننده مورد نظر باشد کافی هست حجم پکت دریافت‌شده (ستون هفت) را با مقدار کل حجم دریافت‌شده جمع کنیم.
برای رسم نمودار هم بازه زمانی ۰ تا ۱۰۰ را به ۱۰۰ بازه ۱-ثانیه‌ای تقسیم شده است و در هر بازه زمانی ۱-ثانیه‌ای، مقدار throughput محاسبه شده است. بنابراین نمودارها تغییرات throughput را در طول زمان نشان می‌دهند.

i. Bandwidth = 1.5Mbps

```
$ python throughput.py 1p5mbps/wireless.tr 7  
Total Data Received by Node _7_ = 19053008  
Total Time = 100  
Throughput = 190530.08 bits/sec
```

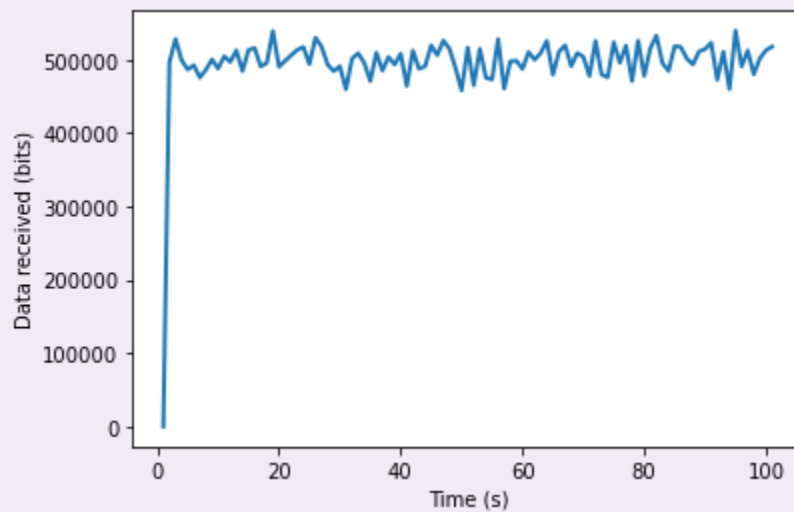


```
$ python throughput.py 1p5mbps/wireless.tr 8  
Total Data Received by Node _8_ = 4234364  
Total Time = 100  
Throughput = 42343.64 bits/sec
```

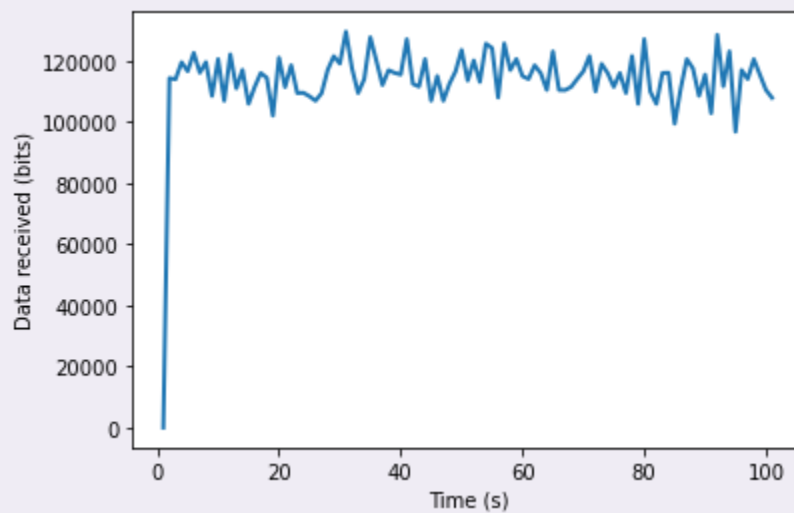


ii. Bandwidth = 55Mbps

```
$ python throughput.py 55mbps/wireless.tr 7  
Total Data Received by Node _7_ = 49981796  
Total Time = 100  
Throughput = 499817.96 bits/sec
```

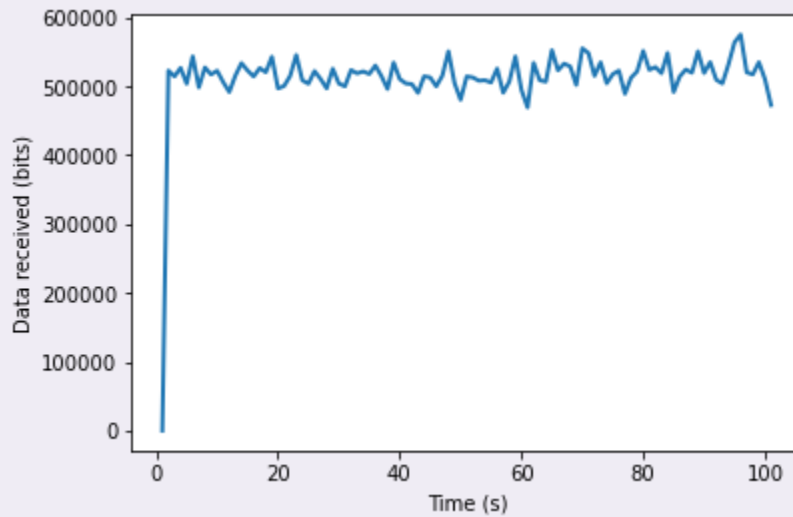


```
$ python throughput.py 55mbps/wireless.tr 8  
Total Data Received by Node _8_ = 11493472  
Total Time = 100  
Throughput = 114934.72 bits/sec
```

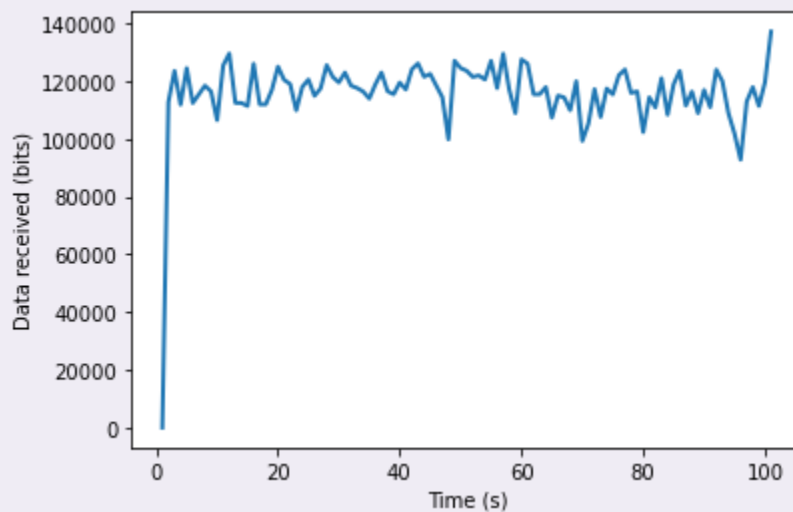


iii. Bandwidth = 155Mbps

```
$ python throughput.py 155mbps/wireless.tr 7  
Total Data Received by Node _7_ = 51801588  
Total Time = 100  
Throughput = 518015.88 bits/sec
```



```
$ python throughput.py 155mbps/wireless.tr 8  
Total Data Received by Node _8_ = 11698416  
Total Time = 100  
Throughput = 116984.16 bits/sec
```



b. Packet Transfer Ratio

برای محاسبه packet delivery ratio یا همان packet transfer ratio کافی است از رابطه زیر استفاده کنیم:

$$\text{Packet Delivery Ratio} = \frac{\text{Received packets by destination}}{\text{Sent packets by source}}$$

```
import sys

file = open(sys.argv[1], 'r')
lines = file.readlines()

sender_node = "_" + sys.argv[2] + "_"
receiver_node = "_" + sys.argv[3] + "_"

sent = dict()
packet_received=0;
packet_sent=0;

for line in lines[10:]:
    words = line.split()

    current_node = words[2]
    packet_id = int(words[5])

    if packet_id == 0:
        continue

    if words[0] == "s" and words[2] == sender_node:
        sent[packet_id] = True
        packet_sent+=1

    if words[0] == "r" and packet_id in sent:
        packet_received+=1

print(f"packets sent from {sender_node} = {packet_sent}")
print(f"packets received from {receiver_node} = {packet_received}")
```



```
print(f"Transfer Ratio = {packet_received/packet_sent}")
```

i. Bandwidth = 1.5Mbps

```
$ python3 transfer_ratio.py 1p5mbps/wireless.tr 0 7
packets sent from _0_ = 25546
packets received from _7_ = 25530
Transfer Ratio = 0.9993736788538323
```

```
$ python3 transfer_ratio.py 1p5mbps/wireless.tr 2 8
packets sent from _2_ = 61283
packets received from _8_ = 43427
Transfer Ratio = 0.7086304521645481
```

ii. Bandwidth = 55Mbps

```
$ python3 transfer_ratio.py 55mbps/wireless.tr 0 7
packets sent from _0_ = 66953
packets received from _7_ = 66946
Transfer Ratio = 0.999895449046346
```

```
$ python3 transfer_ratio.py 55mbps/wireless.tr 2 8
packets sent from _2_ = 76124
packets received from _8_ = 72247
Transfer Ratio = 0.9490699385213599
```

iii. Bandwidth = 155Mbps

```
$ python3 transfer_ratio.py 155mbps/wireless.tr 0 7
packets sent from _0_ = 69413
packets received from _7_ = 69394
Transfer Ratio = 0.999726276057799
```

```
$ python3 transfer_ratio.py 155mbps/wireless.tr 2 8
packets sent from _2_ = 76535
packets received from _8_ = 73069
Transfer Ratio = 0.9547135297576272
```

c. Average End-to-End delay

برای محاسبه End-to-End delay برای هر پکت کافی است اختلاف زمان دریافت آن و فرستادن آن را محاسبه کنیم. نهایتاً میانگین این عدد را به ازای تمام پکت‌ها محاسبه می‌کنیم. برای محاسبه این مقدار از اسکریپت زیر استفاده شده است:

```
import sys

file = open(sys.argv[1], 'r')
lines = file.readlines()

sender_node = "_" + sys.argv[2] + "_"
receiver_node = "_" + sys.argv[3] + "_"

sent = dict()
total_delay = 0
cnt = 0

for line in lines[10:]:
    words = line.split()

    trace_level = words[3]
    packet_size = int(words[7])
    payload_type = words[6]
    current_node = words[2]
    event_type = words[0]
    packet_id = int(words[5])
    time = float(words[1])

    if packet_id == 0:
        continue

    if words[0] == "s" and words[2] == sender_node:
        sent[(sender_node, packet_id, payload_type, trace_level,
        packet_size)] = time

    if words[0] == "r" and (sender_node, packet_id, payload_type,
    trace_level, packet_size) in sent:
        delay = time - sent[(sender_node, packet_id, payload_type,
        trace_level, packet_size)]

        print(sent[(sender_node, packet_id, payload_type, trace_level,
        packet_size)], time)
```

```

total_delay += delay
cnt += 1

print(f"Total Delay from {receiver_node} to {sender_node} = {total_delay}")
print(f"Number of packets transferred from {sender_node} and
{receiver_node} = {cnt}")
print(f"Average End-to-End Delay = {total_delay/cnt}")

```

مشخصات هر پکتی که از فرستنده send می‌شود را در یک dictionary ذخیره می‌کنیم. با مشاهده هر پکت receive شده توسط دریافت‌کننده، زمان ارسال آن را از دیکشنری دریافت کرده و اختلاف آن‌ها را محاسبه می‌کنیم.

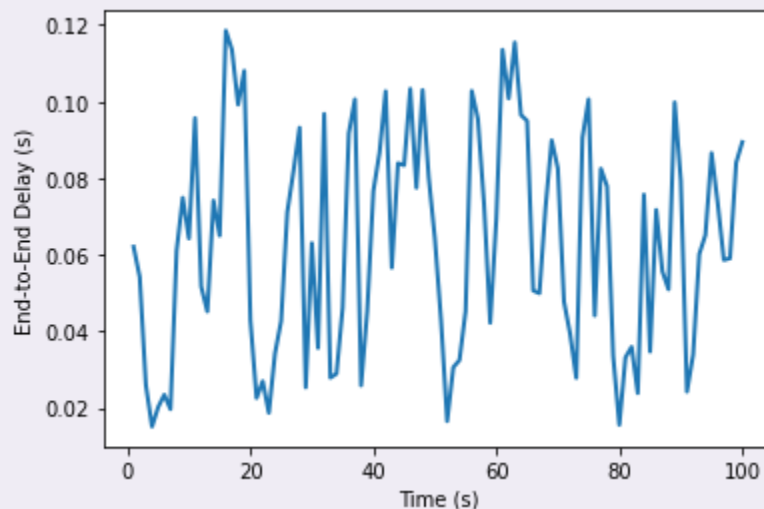
برای رسم نمودار هم بازه زمانی ۰ تا ۱۰۰ را به ۱۰۰ بازه ۱-ثانیه‌ای تقسیم شده است و در هر بازه زمانی ۱-ثانیه‌ای، میانگین مقدار End-to-End Delay محاسبه شده است. بنابراین نمودارها تغییرات End-to-End Delay را در طول زمان نشان می‌دهند.

i. Bandwidth = 1.5Mbps

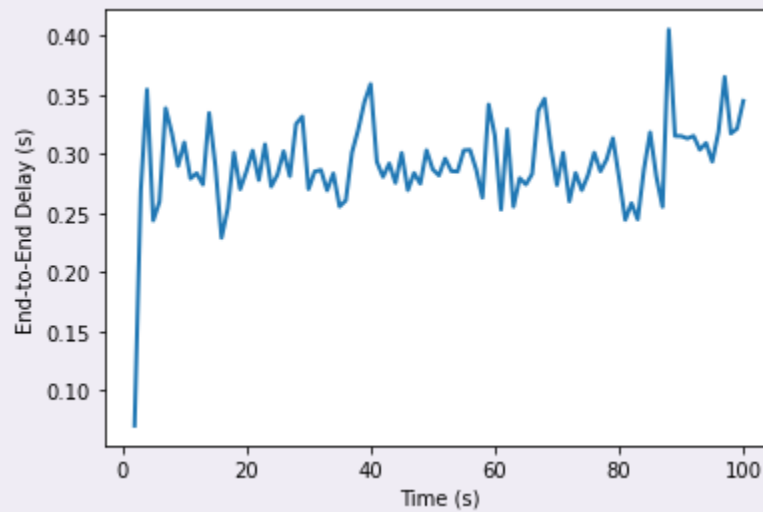
```

$ python e2edelay.py 1p5mbps/wireless.tr 0 7
Total Delay from _0_ to _7_ = 1071.7622814289496s
Number of packets transferred from _0_ and _7_ = 17010
Average End-to-End Delay = 0.06300777668600527s

```

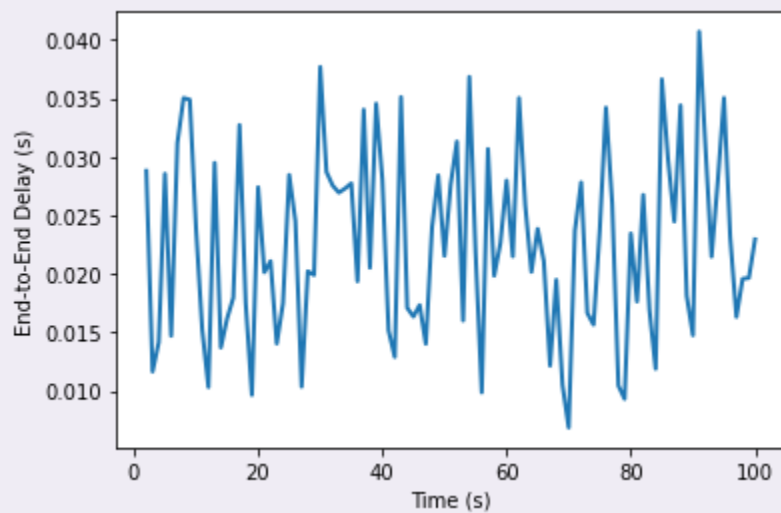


```
$ python e2edelay.py 1p5mbps/wireless.tr 2 8
Total Delay from _2_ to _8_ = 4888.598235852653s
Number of packets transferred from _2_ and _8_ = 16760
Average End-to-End Delay = 0.29168247230624417s
```

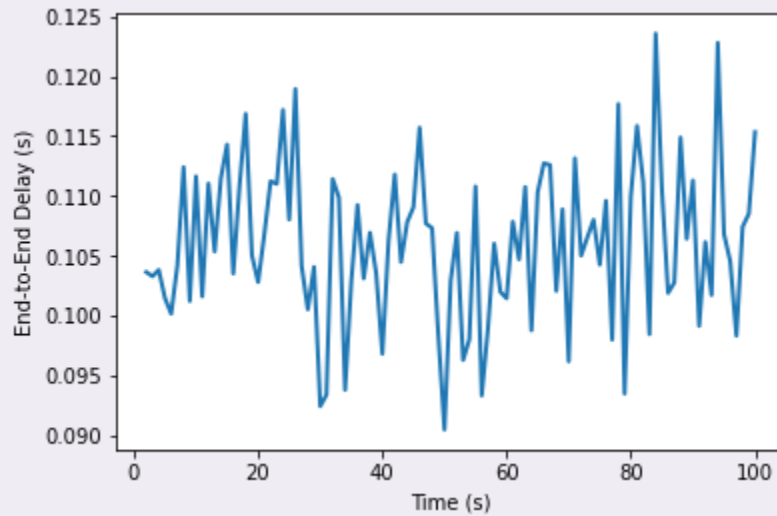


ii. Bandwidth = 55Mbps

```
$ python e2edelay.py 55mbps/wireless.tr 0 7
Total Delay from _0_ to _7_ = 1014.3540029538822s
Number of packets transferred from _0_ and _7_ = 44626
Average End-to-End Delay = 0.022730112556668357s
```

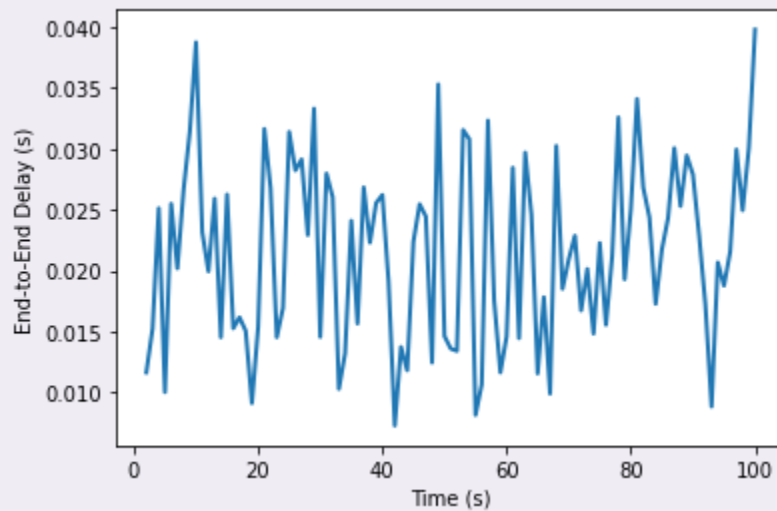


```
$ python e2edelay.py 55mbps/wireless.tr 2 8
Total Delay from _2_ to _8_ = 4792.209511220734s
Number of packets transferred from _2_ and _8_ = 45580
Average End-to-End Delay = 0.10513842718781778s
```



iii. Bandwidth = 155Mbps

```
$ python e2edelay.py 155mbps/wireless.tr 0 7
Total Delay from _0_ to _7_ = 997.4827967223022s
Number of packets transferred from _0_ and _7_ = 46250
Average End-to-End Delay = 0.021567195604806532s
```



```
$ python e2edelay.py 155mbps/wireless.tr 2 8  
Total Delay from _2_ to _8_ = 4751.912642871615s  
Number of packets transferred from _2_ and _8_ = 46402  
Average End-to-End Delay = 0.10240749629049643s
```

