

About Cellsim

1. How cellsim works

The cellsim models two interfaces, `client_side` and `internet_side` as objects of class `PacketSocket`, which bind the specific interface, so we can use system calls to receive data and send data between two sides.

The cellsim holds the packets and then transmit them later with help of class `DelayQueue`, which has two objects, `uplink`(from `client_side` to `internet_side`) and `downlink`(from `internet_side` to `client_side`), to initiate an object of such class, it loads in the trace file and create a vector(mutable array) called `_schedule` of relative timestamps of the trace. When the cellsim starts, it updates `_schedule` with current timestamp(add with each entry). When a packet arrives, say from client side, the file descriptor of `client_side` receive some data and write them in `uplink`(push them in queue `_delay`), then `uplink` would check with function `tick()`, that if the current timestamp is less than or equal to the entry in `_schedule` and also there's available packets in `_delay`, then it send the content to the other side.

Uplink and downlink works as a temporary transfer station.

Packets flow:

```
client_side —> uplink(delayed according to trace) —> internet_side
internet_side —> downlink(delayed according to trace) —> client_side
```

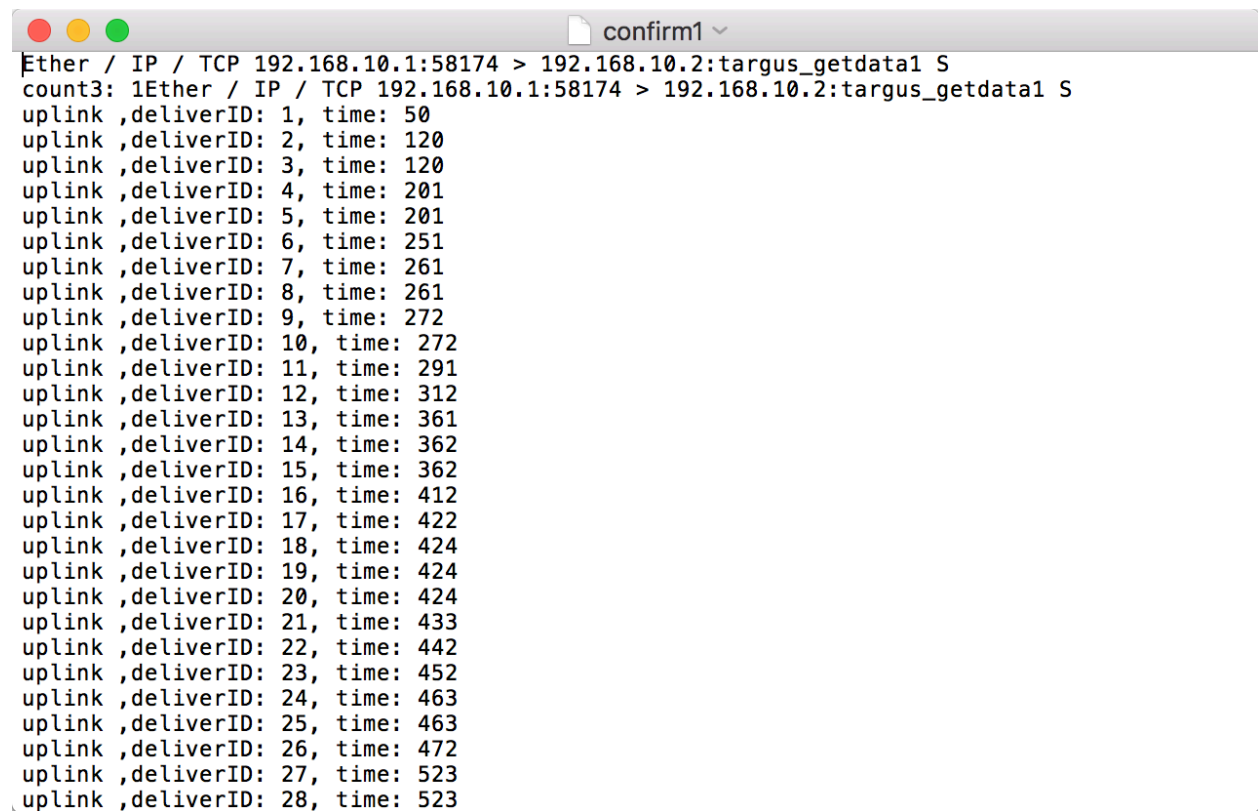
2. Idea about test correspondence between result and trace

Basically, we can know at which time the packet was transferred to the other side, since the cellsim uses these timestamp to compare with `_schedule`, so we can

output these timestamps and compare them at each experiment and also with the trace file.

3. Results and analysis

I ran some test based on section 2. To make it more clear, I only output the uplink results. Here are three consecutive run result:

A screenshot of a terminal window with a title bar containing three colored circles (red, yellow, green) and the text 'confirm1'. The terminal displays the output of a network test. The first line shows the command: 'Ether / IP / TCP 192.168.10.1:58174 > 192.168.10.2:targus_getdata1 S'. The second line shows the result: 'count3: 1Ether / IP / TCP 192.168.10.1:58174 > 192.168.10.2:targus_getdata1 S'. The following lines list 28 uplink results, each with a deliverID and a time value. The results are as follows:

```
Ether / IP / TCP 192.168.10.1:58174 > 192.168.10.2:targus_getdata1 S
count3: 1Ether / IP / TCP 192.168.10.1:58174 > 192.168.10.2:targus_getdata1 S
uplink ,deliverID: 1, time: 50
uplink ,deliverID: 2, time: 120
uplink ,deliverID: 3, time: 120
uplink ,deliverID: 4, time: 201
uplink ,deliverID: 5, time: 201
uplink ,deliverID: 6, time: 251
uplink ,deliverID: 7, time: 261
uplink ,deliverID: 8, time: 261
uplink ,deliverID: 9, time: 272
uplink ,deliverID: 10, time: 272
uplink ,deliverID: 11, time: 291
uplink ,deliverID: 12, time: 312
uplink ,deliverID: 13, time: 361
uplink ,deliverID: 14, time: 362
uplink ,deliverID: 15, time: 362
uplink ,deliverID: 16, time: 412
uplink ,deliverID: 17, time: 422
uplink ,deliverID: 18, time: 424
uplink ,deliverID: 19, time: 424
uplink ,deliverID: 20, time: 424
uplink ,deliverID: 21, time: 433
uplink ,deliverID: 22, time: 442
uplink ,deliverID: 23, time: 452
uplink ,deliverID: 24, time: 463
uplink ,deliverID: 25, time: 463
uplink ,deliverID: 26, time: 472
uplink ,deliverID: 27, time: 523
uplink ,deliverID: 28, time: 523
```

confirm2

```
Ether / IP / TCP 192.168.10.1:58177 > 192.168.10.2:targus_getdata1 SEC
count3: 1Ether / IP / TCP 192.168.10.1:58177 > 192.168.10.2:targus_getdata1 SEC
uplink ,deliverID: 1, time: 50
uplink ,deliverID: 2, time: 120
uplink ,deliverID: 3, time: 120
uplink ,deliverID: 4, time: 201
uplink ,deliverID: 5, time: 201
uplink ,deliverID: 6, time: 251
uplink ,deliverID: 7, time: 261
uplink ,deliverID: 8, time: 261
uplink ,deliverID: 9, time: 273
uplink ,deliverID: 10, time: 273
uplink ,deliverID: 11, time: 291
uplink ,deliverID: 12, time: 312
uplink ,deliverID: 13, time: 361
uplink ,deliverID: 14, time: 362
uplink ,deliverID: 15, time: 362
uplink ,deliverID: 16, time: 414
uplink ,deliverID: 17, time: 423
uplink ,deliverID: 18, time: 423
uplink ,deliverID: 19, time: 423
uplink ,deliverID: 20, time: 423
uplink ,deliverID: 21, time: 432
uplink ,deliverID: 22, time: 443
uplink ,deliverID: 23, time: 452
uplink ,deliverID: 24, time: 464
uplink ,deliverID: 25, time: 464
uplink ,deliverID: 26, time: 472
uplink ,deliverID: 27, time: 523
uplink ,deliverID: 28, time: 523
```

confirm3

```
Ether / IP / TCP 192.168.10.1:58179 > 192.168.10.2:targus_getdata1 SEC
count3: 1Ether / IP / TCP 192.168.10.1:58179 > 192.168.10.2:targus_getdata1 SEC
uplink ,deliverID: 1, time: 50
uplink ,deliverID: 2, time: 120
uplink ,deliverID: 3, time: 120
uplink ,deliverID: 4, time: 201
uplink ,deliverID: 5, time: 201
uplink ,deliverID: 6, time: 251
uplink ,deliverID: 7, time: 261
uplink ,deliverID: 8, time: 261
uplink ,deliverID: 9, time: 272
uplink ,deliverID: 10, time: 272
uplink ,deliverID: 11, time: 291
uplink ,deliverID: 12, time: 312
uplink ,deliverID: 13, time: 361
uplink ,deliverID: 14, time: 362
uplink ,deliverID: 15, time: 362
uplink ,deliverID: 16, time: 412
uplink ,deliverID: 17, time: 422
uplink ,deliverID: 18, time: 423
uplink ,deliverID: 19, time: 423
uplink ,deliverID: 20, time: 423
uplink ,deliverID: 21, time: 432
uplink ,deliverID: 22, time: 442
uplink ,deliverID: 23, time: 452
uplink ,deliverID: 24, time: 464
uplink ,deliverID: 25, time: 464
uplink ,deliverID: 26, time: 472
uplink ,deliverID: 27, time: 523
uplink ,deliverID: 28, time: 523
```

We can see, although with some little variance(at most 2 milliseconds) the output time for each packet is quite certain. Then after two days, I tested again.

```
confirm_again ~
Ether / IP / TCP 192.168.10.1:62610 > 192.168.10.2:targus_getdata1 S
count3: 1Ether / IP / TCP 192.168.10.1:62610 > 192.168.10.2:targus_getdata1 S
uplink ,deliverID: 1, time: 52
uplink ,deliverID: 2, time: 120
uplink ,deliverID: 3, time: 120
uplink ,deliverID: 4, time: 201
uplink ,deliverID: 5, time: 201
uplink ,deliverID: 6, time: 251
uplink ,deliverID: 7, time: 261
uplink ,deliverID: 8, time: 261
uplink ,deliverID: 9, time: 273
uplink ,deliverID: 10, time: 273
uplink ,deliverID: 11, time: 291
uplink ,deliverID: 12, time: 313
uplink ,deliverID: 13, time: 361
uplink ,deliverID: 14, time: 362
uplink ,deliverID: 15, time: 362
uplink ,deliverID: 16, time: 412
uplink ,deliverID: 17, time: 422
uplink ,deliverID: 18, time: 423
uplink ,deliverID: 19, time: 423
uplink ,deliverID: 20, time: 423
uplink ,deliverID: 21, time: 432
uplink ,deliverID: 22, time: 442
uplink ,deliverID: 23, time: 452
uplink ,deliverID: 24, time: 464
uplink ,deliverID: 25, time: 464
uplink ,deliverID: 26, time: 472
uplink ,deliverID: 27, time: 523
uplink ,deliverID: 28, time: 523
```

The result still matches. However, there is a problem, if we pause for some seconds before iperf, like 30 seconds. Then the result would be different.

```
Ether / IP / UDP 192.168.10.1:db_lsp_disc > 192.168.10.255:db_lsp_disc / Raw
Ether / IPv6 / ICMPv6ND_RS / ICMPv6 Neighbor Discovery Option - Source Link-Layer Address
64:31:50:43:08:31
Ether / IP / UDP / DNS Qry "_airport._tcp.local."
Ether / IPv6 / UDP / DNS Qry "_airport._tcp.local."
Ether / IPv6 / UDP / DNS Qry "_nfs._tcp.local."
Ether / IP / UDP 0.0.0.0:bootpc > 255.255.255.255:bootps / BOOTP / DHCP
Ether / IP / UDP 192.168.10.1:58393 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.10.1:58393 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.10.1:58393 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.10.1:58393 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 0.0.0.0:bootpc > 255.255.255.255:bootps / BOOTP / DHCP
Ether / IP / TCP 192.168.10.1:58208 > 192.168.10.2:targus_getdata1 S
count3: 40Ether / IP / TCP 192.168.10.1:58208 > 192.168.10.2:targus_getdata1 S
uplink ,deliverID: 1, time: 50
uplink ,deliverID: 2, time: 50
uplink ,deliverID: 3, time: 50
uplink ,deliverID: 4, time: 50
uplink ,deliverID: 5, time: 50
uplink ,deliverID: 6, time: 50
uplink ,deliverID: 7, time: 50
uplink ,deliverID: 8, time: 50
uplink ,deliverID: 9, time: 50
uplink ,deliverID: 10, time: 121
uplink ,deliverID: 11, time: 121
uplink ,deliverID: 12, time: 201
uplink ,deliverID: 13, time: 201
uplink ,deliverID: 14, time: 251
uplink ,deliverID: 15, time: 261
```

The first schedule was sent with many packets but later on the result still matches pretty well with others. That could be a result of many packets sending to the interface that does not trigger the cellsim also be processed by cellsim after its start. I'm considering something like a flush for interfaces to fix it?