

# COMP90007 Internet Technology

Week5

Yiran (Scott) Ruan

Email: [yrrua@unimelb.edu.au](mailto:yrrua@unimelb.edu.au)

Web Page: <https://yiranruan.github.io>

# Device

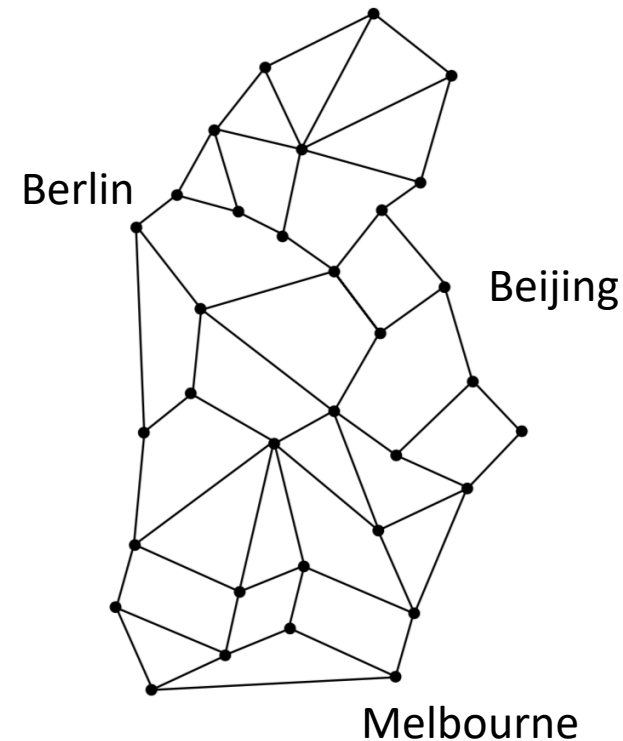
- Of course you are welcome to use your laptop
- We are provided you two servers
  - ssh username@digitalis.eng.unimelb.edu.au
  - ssh username@digitalis2.eng.unimelb.edu.au

# Notice

- Please read your Project documentation carefully for more details about your experiments. This slide is just for a better preparation and understanding.

# Measuring the hop count

- When we send a packet from our source host to the target, there are lots of intermediate hosts in the route.

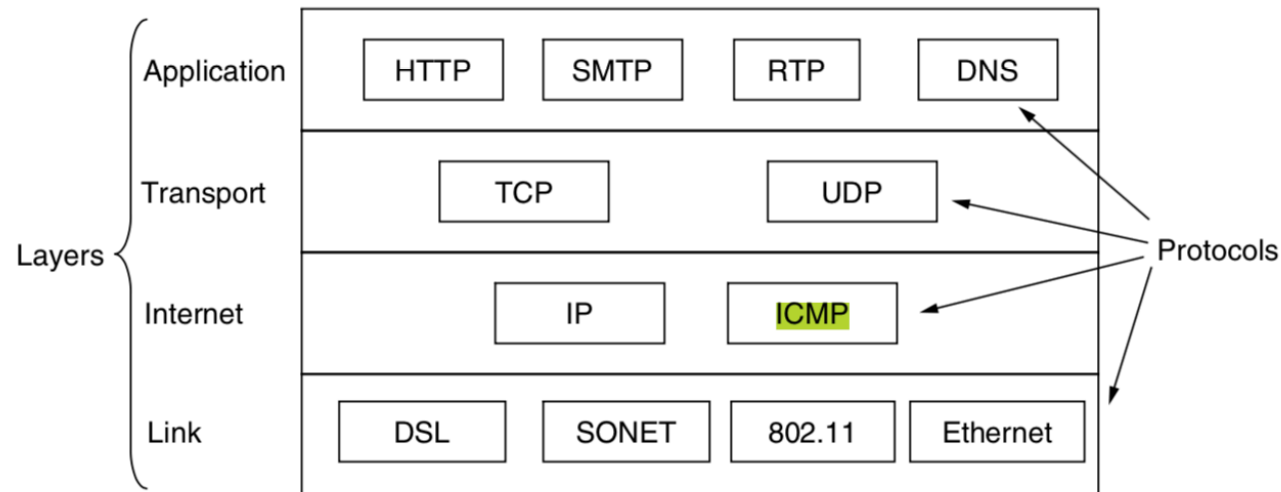


# traceroute/tracert

- To count the hops number between source and target
  - Mac: `traceroute -nw1 cis.unimelb.edu.au`
  - Linux: `traceroute -nw 1 cis.unimelb.edu.au`
  - Win: `tracert -dw1 cis.unimelb.edu.au`

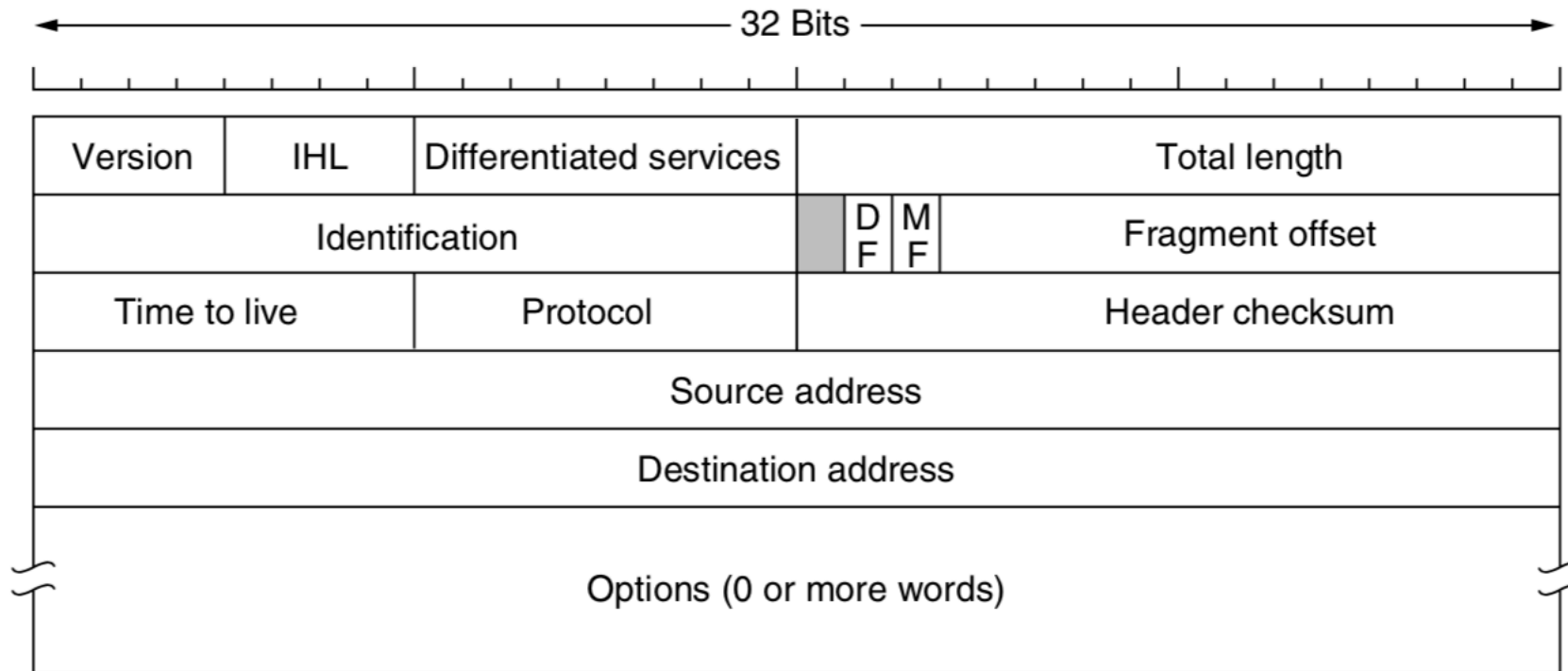
# Traceroute

- Command statement: traceroute hostname
- Using the TTL field in the header of IP (Internet Protocol) and ICMP (Internet Control Message Protocol) message carried in IP packet.



**Figure 1-22.** The TCP/IP model with some protocols we will study.

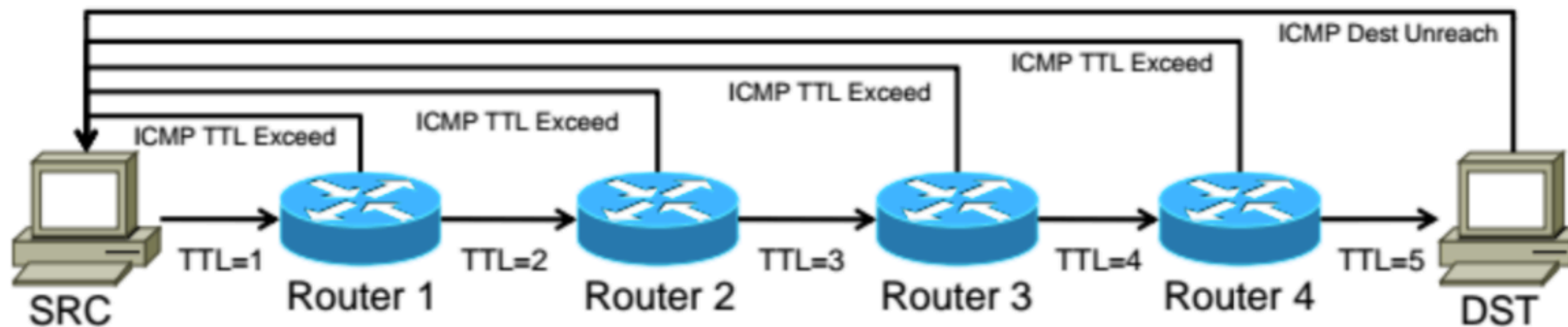
# IPv4 header



**Figure 5-46.** The IPv4 (Internet Protocol) header.

# TTL field

- The *Ttl* (*Time to live*) field is a counter used to limit packet lifetimes. It was originally supposed to count time in seconds, allowing a maximum lifetime of 255 sec. It must be decremented on each hop and is supposed to be decremented multiple times when a packet is queued for a long time in a router. In practice, it just counts hops. When it **hits zero**, the packet is discarded and a warning packet is sent back to the source host.





# ICMP

- ICMP could be used to test the Internet. About a dozen types of ICMP messages are defined. Each ICMP message type is carried encapsulated in an IP packet.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

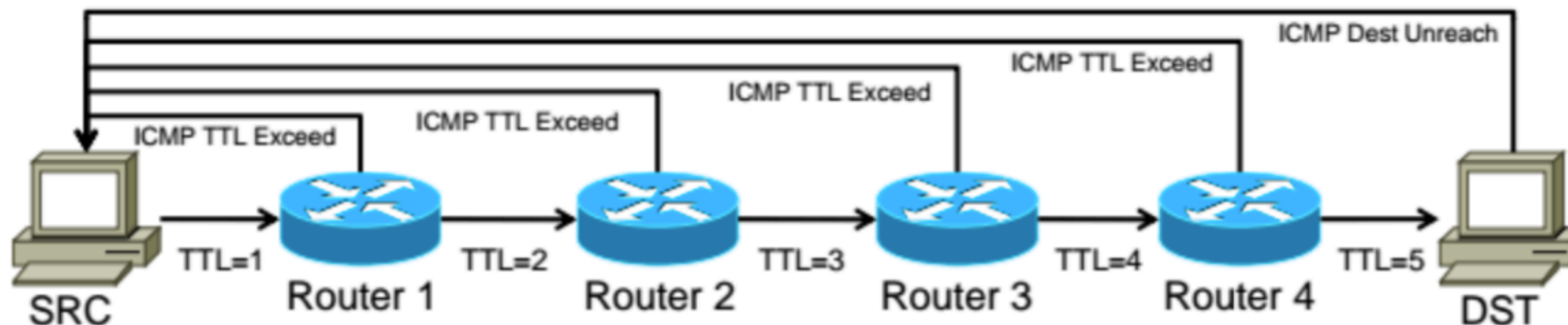
**Figure 5-60.** The principal ICMP message types.

# ICMP – TIME EXCEEDED

- The ***TIME EXCEEDED*** message is sent when a packet is dropped because its *TtL (Time to live)* counter has reached zero. This event is a symptom that packets are looping, or that the counter values are being set too low.

# traceroute

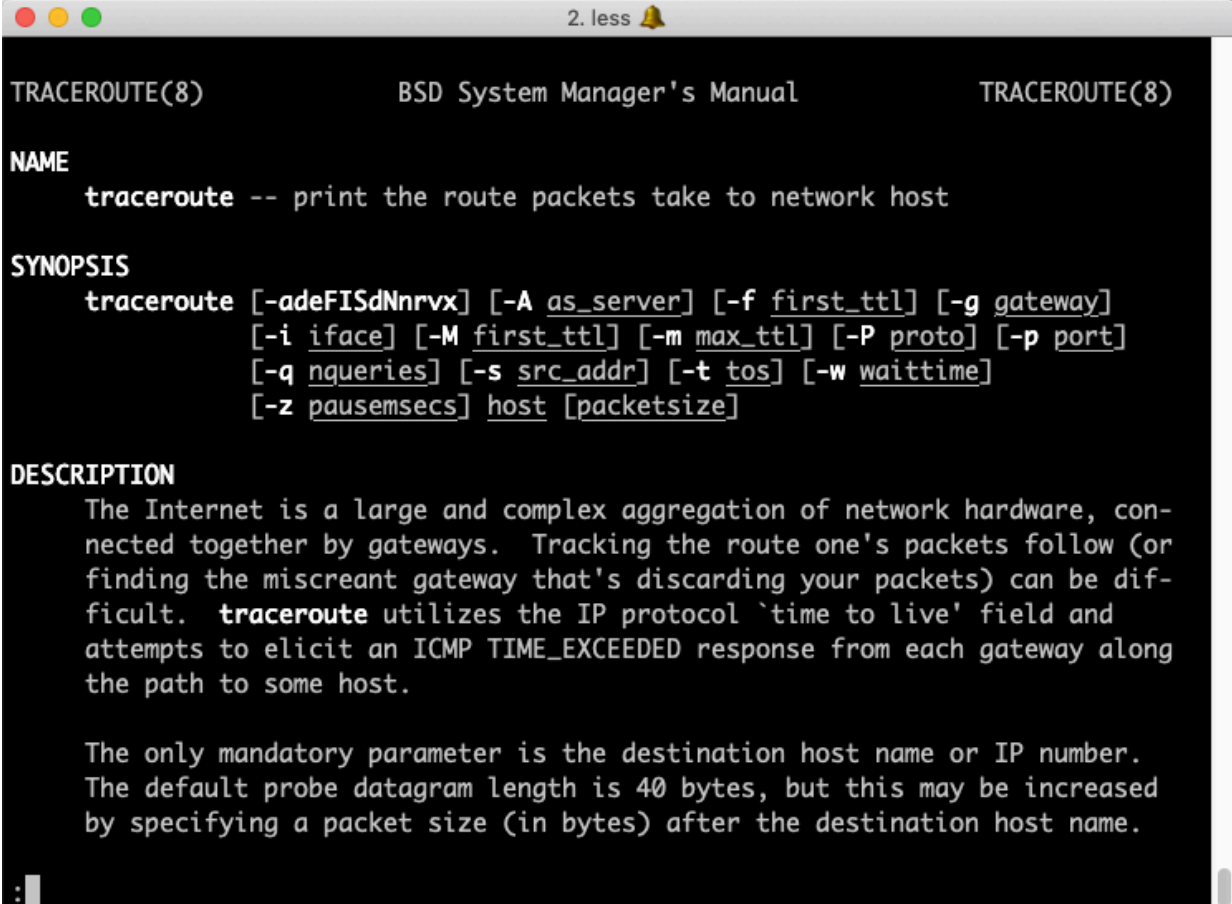
- 1. Send three IP **datagrams** (40 bytes, include source address, target address and sending time label) with TTL = 1
- 2. router: decrease TTL by 1.
  - If TTL = 0, drop this datagram and send back a “ICMP time exceeded” message including source host IP address, the content of the datagram and IP address of the router.
- 3. Source host increase TTL by 1 and send it again.



# Details of traceroute – man traceroute

The help documentation for the ***traceroute*** utility can be accessed by running

- **man traceroute**
- **tracert /?**



```
TRACEROUTE(8) BSD System Manager's Manual TRACEROUTE(8)

NAME
  traceroute -- print the route packets take to network host

SYNOPSIS
  traceroute [-adeFISdNnrvx] [-A as_server] [-f first_ttl] [-g gateway]
             [-i iface] [-M first_ttl] [-m max_ttl] [-P proto] [-p port]
             [-q nqueries] [-s src_addr] [-t tos] [-w waittime]
             [-z pausesecs] host [packetsize]

DESCRIPTION
  The Internet is a large and complex aggregation of network hardware, con-
  nected together by gateways. Tracking the route one's packets follow (or
  finding the miscreant gateway that's discarding your packets) can be dif-
  ficult. traceroute utilizes the IP protocol 'time to live' field and
  attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along
  the path to some host.

  The only mandatory parameter is the destination host name or IP number.
  The default probe datagram length is 40 bytes, but this may be increased
  by specifying a packet size (in bytes) after the destination host name.
```

# traceroute

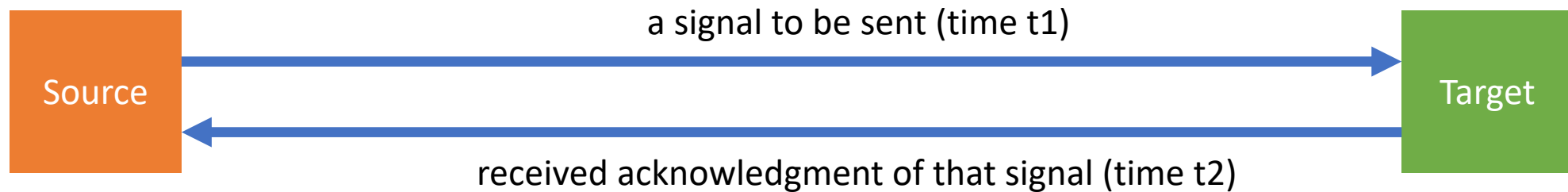
- Each record is one hop
- For each line, there are three time records.
- Traceroute -q 4 [www.google.com](http://www.google.com)
- “\*\*\*” time out hops still count
  - Firewall blocks the ICMP messages
- Latency:
  - Block by some gateway
  - DNS – cannot resolve Host name and Domain name (-n)

```
2. ssh
-bash-4.2$ traceroute -nw 1 www.google.com.au
traceroute to www.google.com.au (172.217.25.131), 30 hops max, 60 byte packets
 1  128.250.106.2  0.566 ms  0.664 ms  0.813 ms
 2  172.18.86.73   0.417 ms  0.531 ms  0.575 ms
 3  172.18.65.17   0.487 ms  0.473 ms  0.516 ms
 4  172.19.1.181   48.734 ms 48.767 ms 48.815 ms
 5  172.19.1.182   0.448 ms  0.487 ms  0.468 ms
 6  172.18.66.254   0.449 ms  *  0.415 ms
 7  138.44.64.62   1.790 ms  1.771 ms  1.813 ms
 8  113.197.15.174  1.588 ms  1.606 ms  1.587 ms
 9  202.158.210.41 13.521 ms 13.526 ms 13.567 ms
10  * * *
11  209.85.250.138 15.302 ms 209.85.247.126 15.765 ms 15.755 ms
12  108.170.247.75 17.008 ms 74.125.37.155 13.173 ms 108.170.247.42 13.920 ms
13  108.170.247.81 12.961 ms 108.170.247.49 14.618 ms 108.170.247.81 13.015 m
s
14  74.125.37.155 13.536 ms 13.535 ms 172.217.25.131 13.279 ms
-bash-4.2$
```

- Connect to VPN
- AnyConnect
- [remote.unimelb.edu.au/student](https://remote.unimelb.edu.au/student)

# Delay

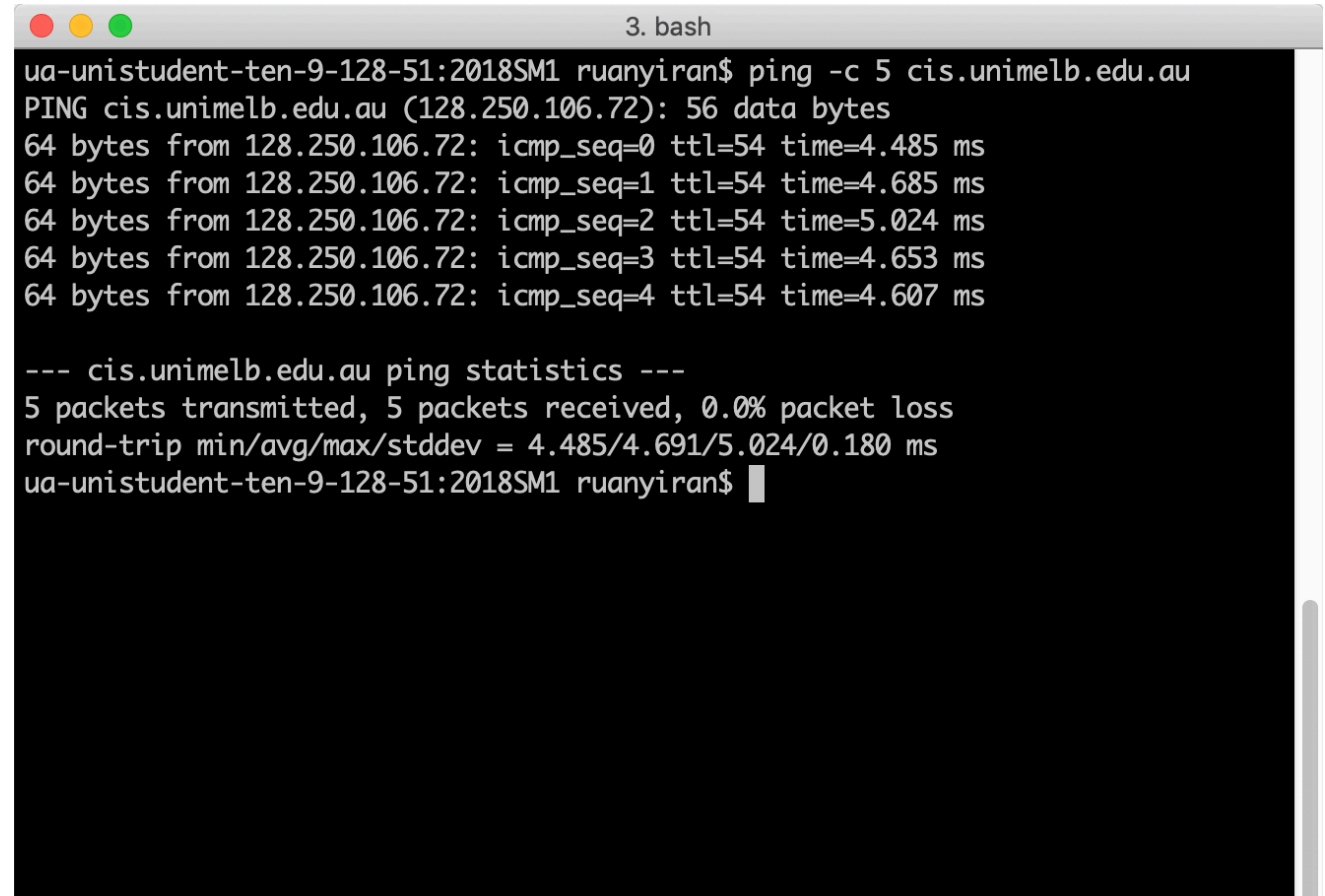
- Round-trip Time (RTT)
  - The total time of one source host to send a packet to the target and wait for its acknowledgement to be received.



$$RTT = t1 + t2$$

# Ping -> Delay

- We will use ***ping*** utility, to measure the round-trip delay of packets. The ping utility should be pre-installed on all major operating systems.
- Win: ping cis.unimelb.edu.au
- Mac: ping -c 5 cis.unimelb.edu.au
  - mac users need to set a specific number for testing packets

A terminal window titled "3. bash" with a dark background and light text. It shows the execution of the command "ping -c 5 cis.unimelb.edu.au". The output displays five individual ping results with their respective times and TTL values, followed by a summary of the statistics.

```
3. bash
ua-unistudent-ten-9-128-51:2018SM1 ruanyiran$ ping -c 5 cis.unimelb.edu.au
PING cis.unimelb.edu.au (128.250.106.72): 56 data bytes
64 bytes from 128.250.106.72: icmp_seq=0 ttl=54 time=4.485 ms
64 bytes from 128.250.106.72: icmp_seq=1 ttl=54 time=4.685 ms
64 bytes from 128.250.106.72: icmp_seq=2 ttl=54 time=5.024 ms
64 bytes from 128.250.106.72: icmp_seq=3 ttl=54 time=4.653 ms
64 bytes from 128.250.106.72: icmp_seq=4 ttl=54 time=4.607 ms

--- cis.unimelb.edu.au ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 4.485/4.691/5.024/0.180 ms
ua-unistudent-ten-9-128-51:2018SM1 ruanyiran$
```



# ICMP – ECHO & ECHO REPLY

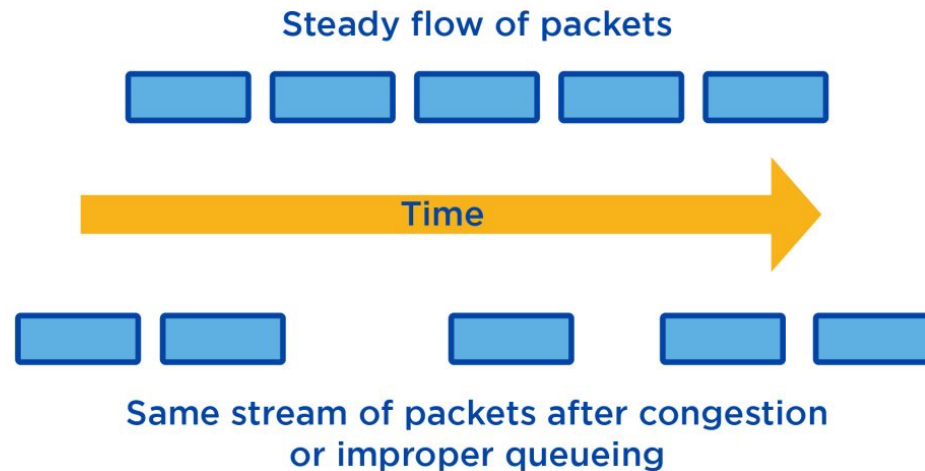
- Using ICMP
  - The **ECHO** and **ECHO REPLY** messages are sent by hosts to see if a given destination is reachable and currently alive. Upon receiving the **ECHO** message, the destination is expected to send back an **ECHO REPLY** message. These messages are used in the **ping** utility that checks if a host is up and on the Internet.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

**Figure 5-60.** The principal ICMP message types.

# Jitter

- Jitter is defined as a variation in the delay of received packets.
- The standard deviation of the round-trip delay time will be taken as the value for jitter for this project.



(<https://www.nextiva.com/blog/network-jitter.html>)

# Jitter

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

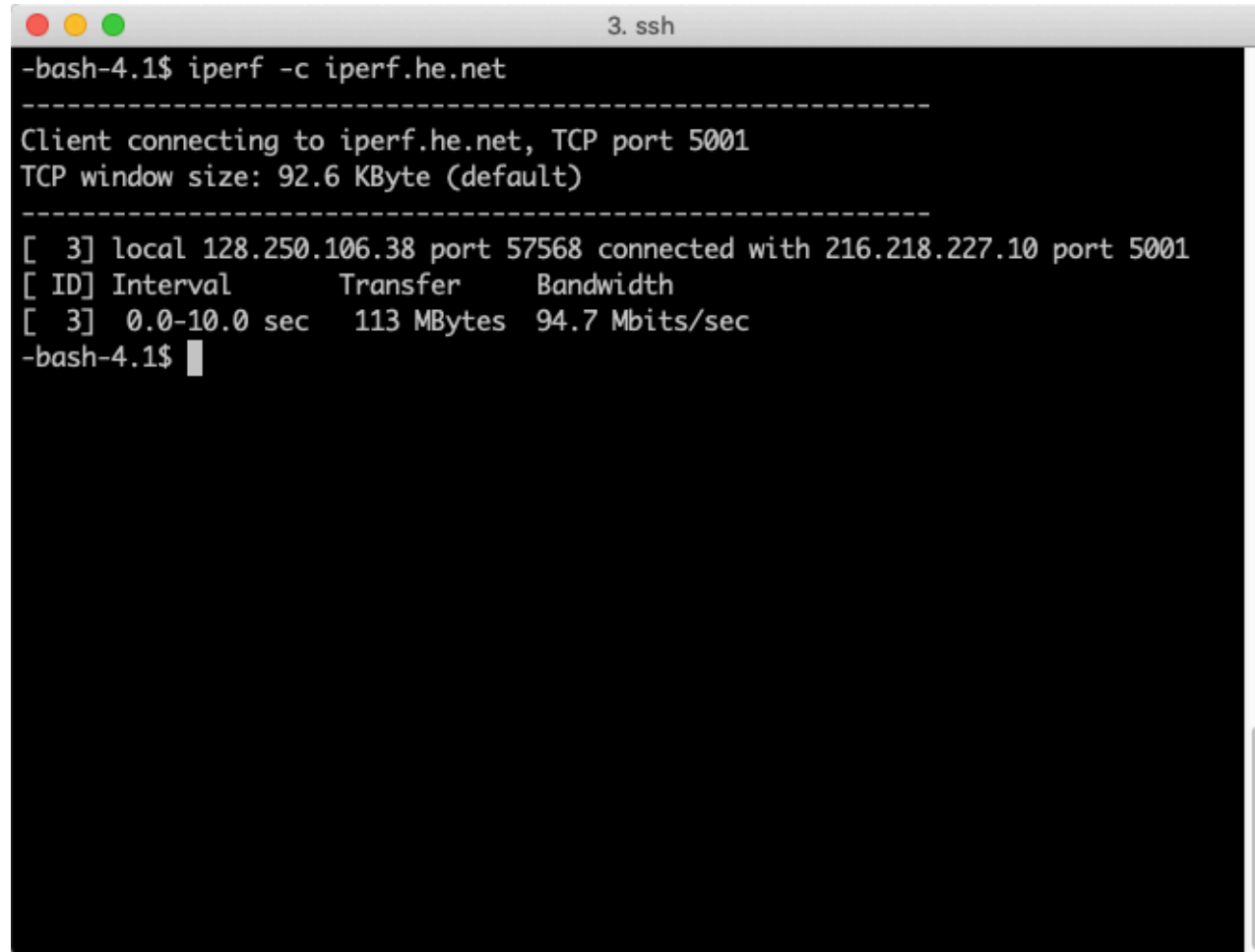
- where the  $\bar{x}$  would be the mean of the set of data

# iperf -> Bandwidth

- iperf: perform bandwidth measurements
- Download: <https://iperf.fr>
- Two modes:
  - Server mode (iperf3 -s): host a server which will listen to incoming requests from a client.
  - Client mode (e.g. iperf3 -c iperf.eenet.ee): connect to the server, and packets will be exchanged and timed between the two hosts to calculate the bandwidth.
- In this project, we will be running ***iperf*** in client mode.

# iperf -c iperf.he.net

- Example

A terminal window titled "3. ssh" with standard macOS window controls (red, yellow, green buttons). The terminal shows the execution of the command "iperf -c iperf.he.net". The output indicates a successful connection to iperf.he.net on TCP port 5001, with a TCP window size of 92.6 KByte. It then shows a connection from a local IP of 128.250.106.38 on port 57568 to the remote IP of 216.218.227.10 on port 5001. A table of performance metrics follows, showing a transfer of 113 MBytes and a bandwidth of 94.7 Mbits/sec over a 10-second interval.

```
-bash-4.1$ iperf -c iperf.he.net
-----
Client connecting to iperf.he.net, TCP port 5001
TCP window size: 92.6 KByte (default)
-----
[ 3] local 128.250.106.38 port 57568 connected with 216.218.227.10 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  113 MBytes 94.7 Mbits/sec
-bash-4.1$
```

```
iperf3 -c speedtest.serverius.net -p 5002
```

### Note:

For speedtest.serverius.net in Table 1, we may need to use the port 5002.

Also, some iperf servers respond to iperf2, rest to iperf3 so you might want to use both iperf2 and iperf3 to verify if the server is responsive.

```

3. ssh
-bash-4.2$ iperf3 -c speedtest.serverius.net -p 5002
Connecting to host speedtest.serverius.net, port 5002
[ 4] local 128.250.106.77 port 40346 connected to 178.21.16.76 port 5002
[ ID] Interval            Transfer      Bandwidth      Retr  Cwnd
[ 4]  0.00-1.00    sec    242 KBytes   1.98 Mbits/sec    0   56.6 KBytes
[ 4]  1.00-2.00    sec   1.57 MBytes  13.1 Mbits/sec    0   334 KBytes
[ 4]  2.00-3.00    sec   3.05 MBytes  25.6 Mbits/sec    5   993 KBytes
[ 4]  3.00-4.00    sec   3.75 MBytes  31.5 Mbits/sec   12   3.59 MBytes
[ 4]  4.00-5.00    sec   7.50 MBytes  62.9 Mbits/sec   13   6.03 MBytes
[ 4]  5.00-6.00    sec  11.2 MBytes  94.4 Mbits/sec    0   6.03 MBytes
[ 4]  6.00-7.00    sec   8.75 MBytes  73.4 Mbits/sec    0   6.03 MBytes
[ 4]  7.00-8.00    sec  12.5 MBytes  105 Mbits/sec    0   6.03 MBytes
[ 4]  8.00-9.00    sec   8.75 MBytes  73.4 Mbits/sec    0   6.03 MBytes
[ 4]  9.00-10.00   sec  11.2 MBytes  94.4 Mbits/sec    0   6.03 MBytes
- - - - -
[ ID] Interval            Transfer      Bandwidth      Retr
[ 4]  0.00-10.00   sec   68.6 MBytes  57.5 Mbits/sec   30
[ 4]  0.00-10.00   sec   68.3 MBytes  57.3 Mbits/sec
sender
receiver



iperf Done.
-bash-4.2$

```

# iperf3

- sender - is iperf client, Upload speed from iperf client to iperf server is measured
- receiver - is iperf server, Download speed on iperf server from iperf client is measured

# Iperf3 – sender or receiver



1 day ago

**using iperf3 - which bandwidth result to use**

Hi

Given that some of the servers only respond to iperf3, but iperf3 pops two bandwidth result end with 'sender' and 'receiver' tag.

However, iperf2 only pops 1 bandwidth result.










So my question is which bandwidth result from iperf3 should we use.

Regards

Derossi

Reply

Thanks for this student's question. Choosing bandwidth of sender would be better. Rahul has given the explanation.





# Servers

- If you want to use server
- NOTICE:
  - `ssh username@digitalis.eng.unimelb.edu.au -> iperf -c xxx`
  - `ssh username@digitalis2.eng.unimelb.edu.au -> iperf3 -c xxx`