COMN - Computer Communications and Networks

Coursework Introduction: Traffic Analysis

Michio Honda

School of Informatics

University of Edinburgh

Text version is also available at
https://git.ecdf.ed.ac.uk/mhonda/comn23cw/blob/main/traffic_analysis/introduction.md

What is traffic analysis

- Essential task in operational networks
  - anomaly detection
  - usage understanding
  - infrastructure planning
- What traffic have we seen? from/to where?

What packet trace looks like

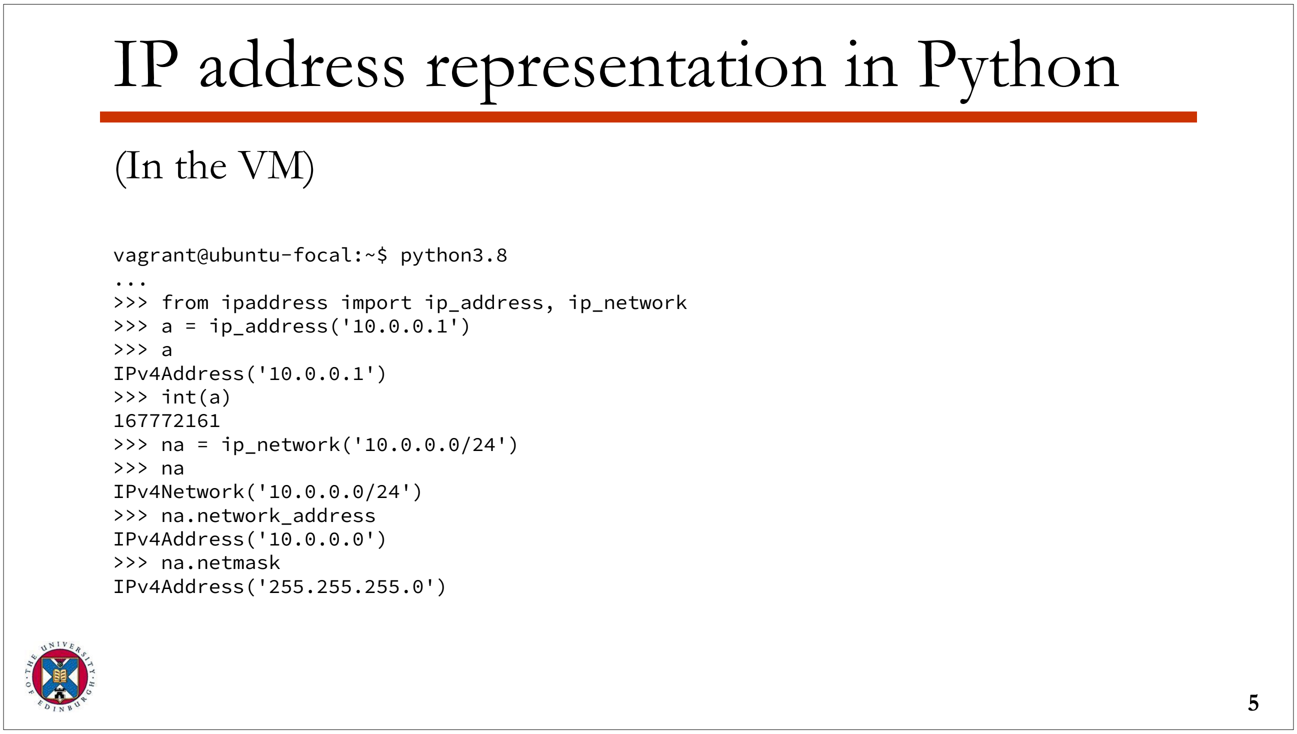
```
05:11:08.999770 IP 133.34.250.140.1194 > 24.26.253.176.39264: UDP, length 26
05:11:08.999770 IP 151.133.237.33.443 > 163.210.142.7.53299: Flags [.], seq 10220:11680,
ack 1, win 83, length 1460
05:11:08.999772 IP 202.132.100.13.50026 > 52.141.212.244.443: Flags [.], ack 2242675521,
win 1029, length 0
05:11:08.999803 IP 203.62.165.212 > 202.119.233.22: ICMP echo request, id 47487, seq
5353, length 12
```

- Typically stored in **pcap** format

Tips: you can always see the packets departing/arriving at your computer using the `tcpdump` command, and save/read them to/from the pcap file with the `-w/-r FILENAME` option.

IP address

- IPv4 address: a 32 bit integer
  - 10.0.0.1 represents 00001010 00000000 00000000 00000001
- An IP address divides into the network address (higher bits) and host address (lower bits) parts
  - The network address owner assigns the host addresses
  - Netmask: How long the network address is
    - 255.255.255.0 means the most significant 24 bits
  - Network address: the address of the network (e.g., 10.0.0.0/24)
    - used to represent a set of hosts or networks
    - e.g., routing table entry - forward packets destined to 129.0.0.0/8 to the switch port 3

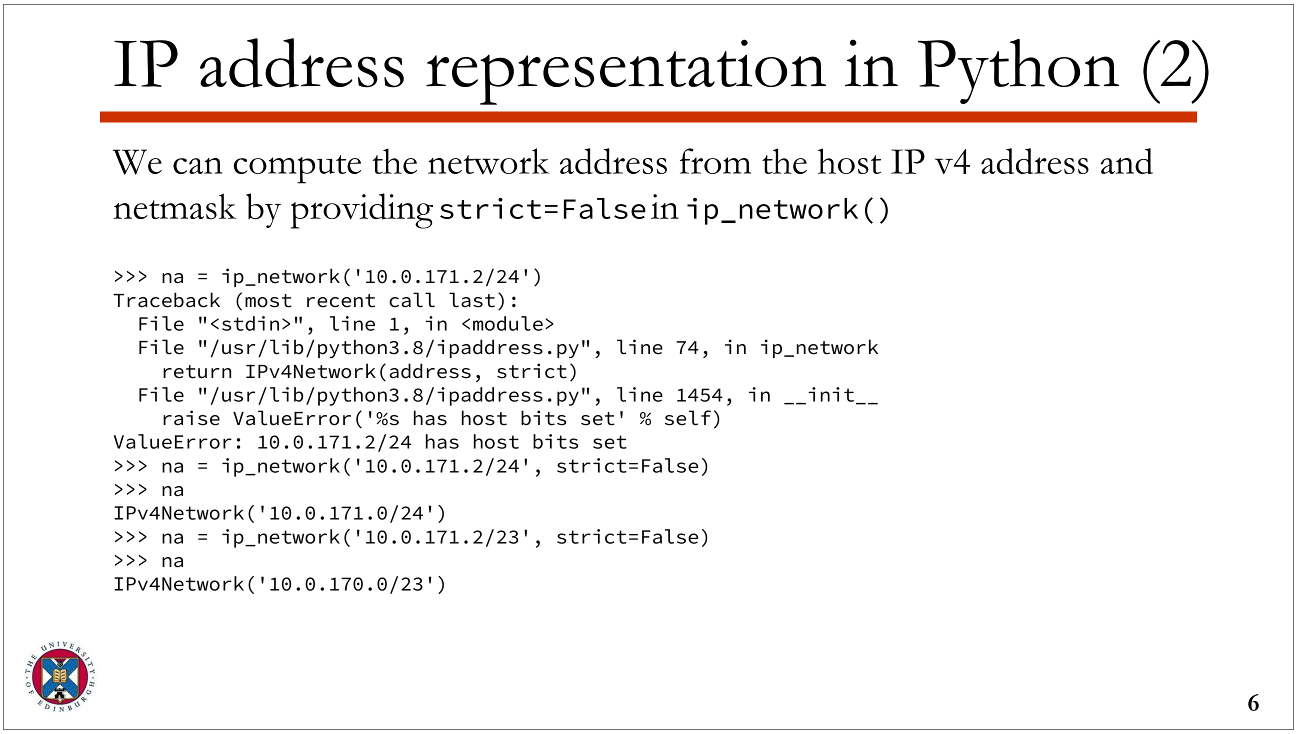


IP address representation in Python

(In the VM)

```
vagrant@ubuntu-focal:~$ python3.8
...
>>> from ipaddress import ip_address, ip_network
>>> a = ip_address('10.0.0.1')
>>> a
IPv4Address('10.0.0.1')
>>> int(a)
167772161
>>> na = ip_network('10.0.0.0/24')
>>> na
IPv4Network('10.0.0.0/24')
>>> na.network_address
IPv4Address('10.0.0.0')
>>> na.netmask
IPv4Address('255.255.255.0')
```

5

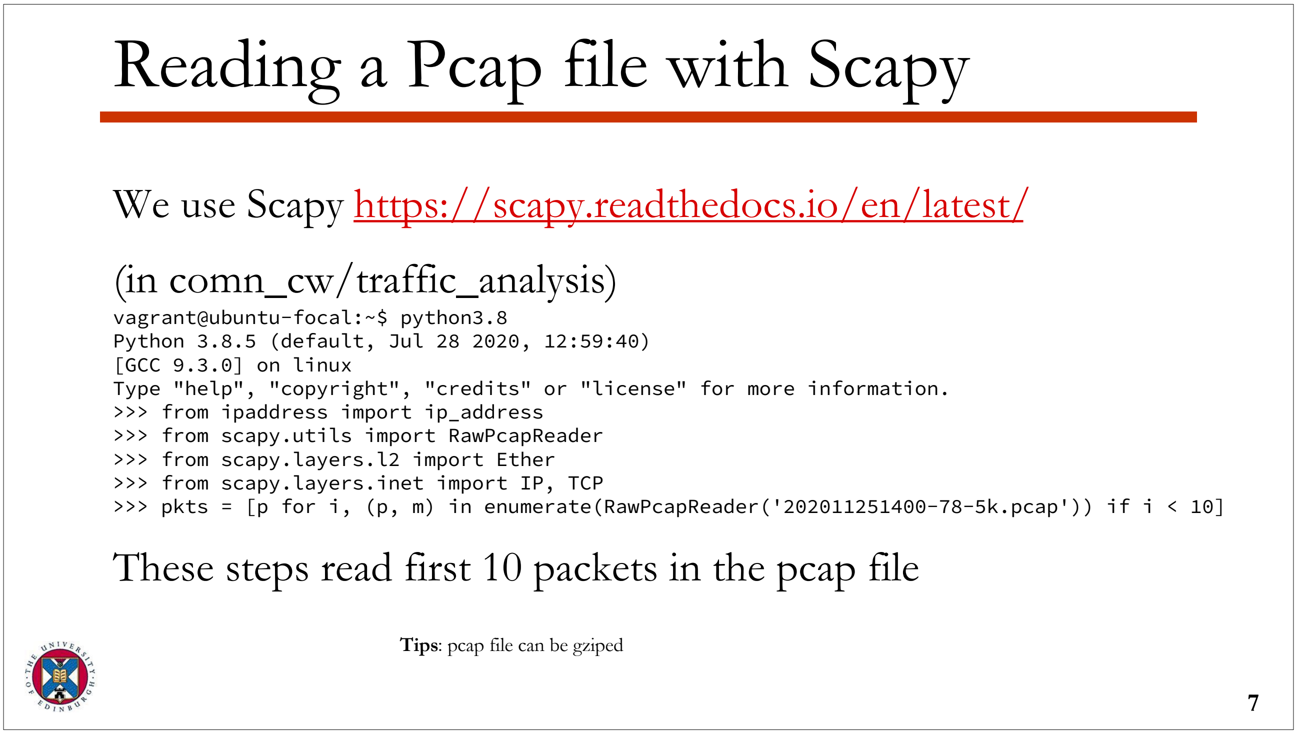


IP address representation in Python (2)

We can compute the network address from the host IP v4 address and netmask by providing `strict=False`in `ip_network()`

```
>>> na = ip_network('10.0.171.2/24')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/python3.8/ipaddress.py", line 74, in ip_network
    return IPv4Network(address, strict)
  File "/usr/lib/python3.8/ipaddress.py", line 1454, in __init__
    raise ValueError('%s has host bits set' % self)
ValueError: 10.0.171.2/24 has host bits set
>>> na = ip_network('10.0.171.2/24', strict=False)
>>> na
IPv4Network('10.0.171.0/24')
>>> na = ip_network('10.0.171.2/23', strict=False)
>>> na
IPv4Network('10.0.170.0/23')
```

6



Reading a Pcap file with Scapy

We use Scapy https://scapy.readthedocs.io/en/latest/

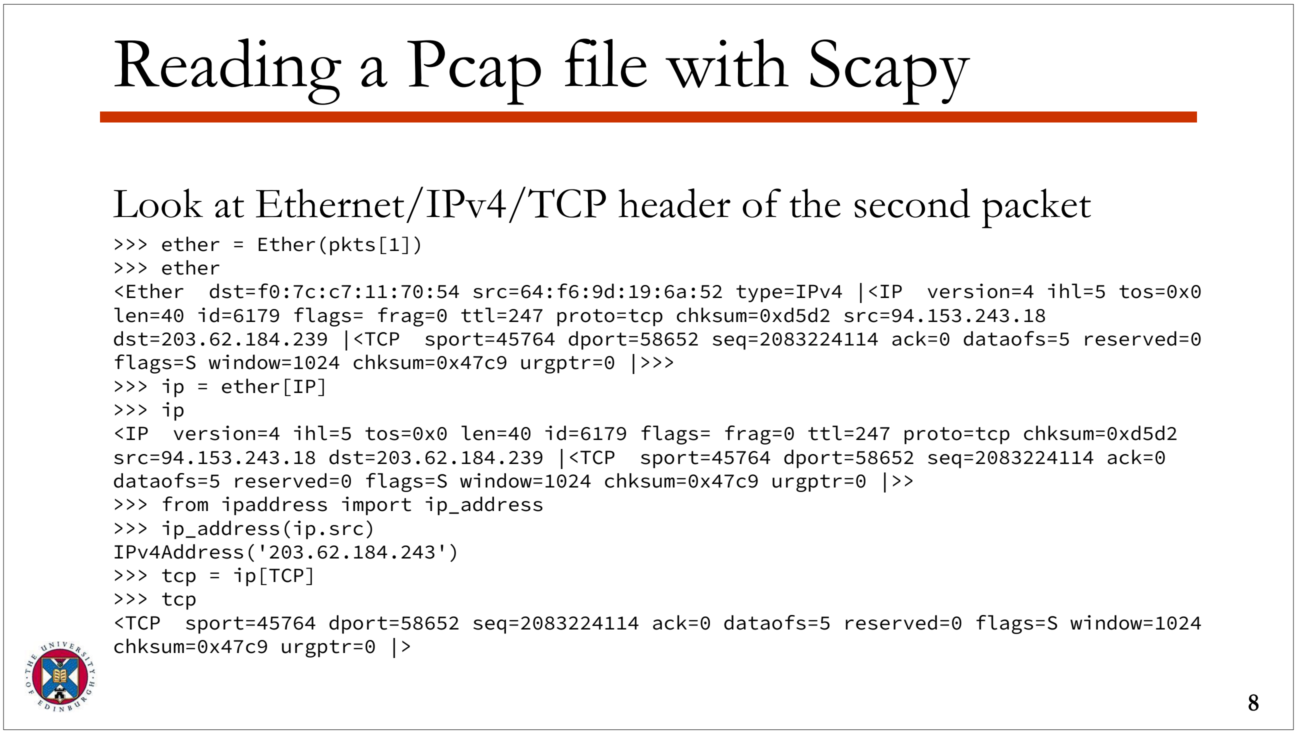
(in comn_cw/traffic_analysis)

```
vagrant@ubuntu-focal:~$ python3.8
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from ipaddress import ip_address
>>> from scapy.utils import RawPcapReader
>>> from scapy.layers.l2 import Ether
>>> from scapy.layers.inet import IP, TCP
>>> pkts = [p for i, (p, m) in enumerate(RawPcapReader('202011251400-78-5k.pcap')) if i < 10]
```

These steps read first 10 packets in the pcap file

Tips: pcap file can be gziped

7



Reading a Pcap file with Scapy

Look at Ethernet/IPv4/TCP header of the second packet

```
>>> ether = Ether(pkts[1])
>>> ether
<Ether  dst=f0:7c:c7:11:70:54 src=64:f6:9d:19:6a:52 type=IPv4 |<IP  version=4 ihl=5 tos=0x0
len=40 id=6179 flags= frag=0 ttl=247 proto=tcp chksum=0xd5d2 src=94.153.243.18
dst=203.62.184.239 |<TCP  sport=45764 dport=58652 seq=2083224114 ack=0 dataofs=5 reserved=0
flags=S window=1024 chksum=0x47c9 urgptr=0 |>>>
>>> ip = ether[IP]
>>> ip
<IP  version=4 ihl=5 tos=0x0 len=40 id=6179 flags= frag=0 ttl=247 proto=tcp chksum=0xd5d2
src=94.153.243.18 dst=203.62.184.239 |<TCP  sport=45764 dport=58652 seq=2083224114 ack=0
dataofs=5 reserved=0 flags=S window=1024 chksum=0x47c9 urgptr=0 |>>
>>> from ipaddress import ip_address
>>> ip_address(ip.src)
IPv4Address('203.62.184.243')
>>> tcp = ip[TCP]
>>> tcp
<TCP  sport=45764 dport=58652 seq=2083224114 ack=0 dataofs=5 reserved=0 flags=S window=1024
chksum=0x47c9 urgptr=0 |>
```

8

Summary

- We learned how to read a pcap file in Python
- Traffic analysis helps us understand:
  - New applications: P2P [1], YouTube [2]
  - New human behavior: SmartPhone [3], COVID-19 [4]
  - Internet status: Security [5], Extensibility [6]
- Many interesting papers at ACM IMC

[1] http://conferences.sigcomm.org/imc/2004/papers/p121-karagiannis.pdf
[2] http://conferences.sigcomm.org/imc/2007/papers/imc78.pdf
[3] http://conferences.sigcomm.org/imc/2010/papers/p281.pdf
[4] https://dl.acm.org/doi/pdf/10.1145/3419394.3423621
[5] http://conferences.sigcomm.org/imc/2014/papers/p475.pdf
[6] https://conferences.sigcomm.org/imc/2011/docs/p181.pdf