

# Management and Operations of Networks, Services, and Systems

## Monitoring and performance

Ricardo Morla

FEUP – GORS/M.EEC, GRS/M.EIC



# Monitoring

- Get a sense of how the network is performing
  - Make sure we're still offering a quality network
  - Essential for FCAPS - Fault, Accounting, Configuration, Performance, Security
  - See application requirements and QoS
- Short-term measurements
  - Identify faults, congestions, and attacks
- Longer-term measurements
  - Traffic engineering, e.g. reroute traffic or negotiate new agreements with peers
  - Upgrade link and device capacity
- Accounting
  - So you know how each client of the network is using the network



# Types of measurements

- Application and user-related measurements
  - e.g. web page loading time
- Device measurements
  - CPU, memory, disk, link usage
  - Temperature, fan speed, other hardware measurements
- Network measurements
  - Traffic data – packet traces, flow data
  - Latency
  - Troughput
  - Routing data



# Passive vs. Active network measurements

- **Passive**
  - Get a sense of the existing traffic in the network
  - Have devices report how much traffic is going through (e.g. SNMP, netflow)
  - Tap a link or copy traffic to monitoring port (port mirroring)
  - For measuring production traffic and its characteristics
- **Active**
  - Inject new, measurement packets in the network
  - Get a sense of how the network reacts to these packets
  - Including responses (e.g. ICMP req./reply for RTT)
  - For measuring the properties of the network (delay, jitter, topology, etc)

<https://dblp.org/db/conf/pam/index.html>



# #1 Packet traces

- Motion-picture-like recording of everything that goes through the network
  - What, when, where, who (?), why (?)
- Raw data – powerful but hard to use
- Difficult to manage
  - Capture limitations (copy data at 1, 10, etc Gbps scale)
  - Storage limitations (Gbps \* minutes, hours, days = ?)
- Difficult to use and process
  - Not in a table like format – would be easier
  - Can write processing rules to create tables – but only partial vision
  - AI and deep learning etc to process traces (raw or features)



# Packet trace usage example

- Classify botnet traffic
- First 400 IP payload bytes, TCP flows
- 20x20 pixels, 0-255
- Mostly good classification results

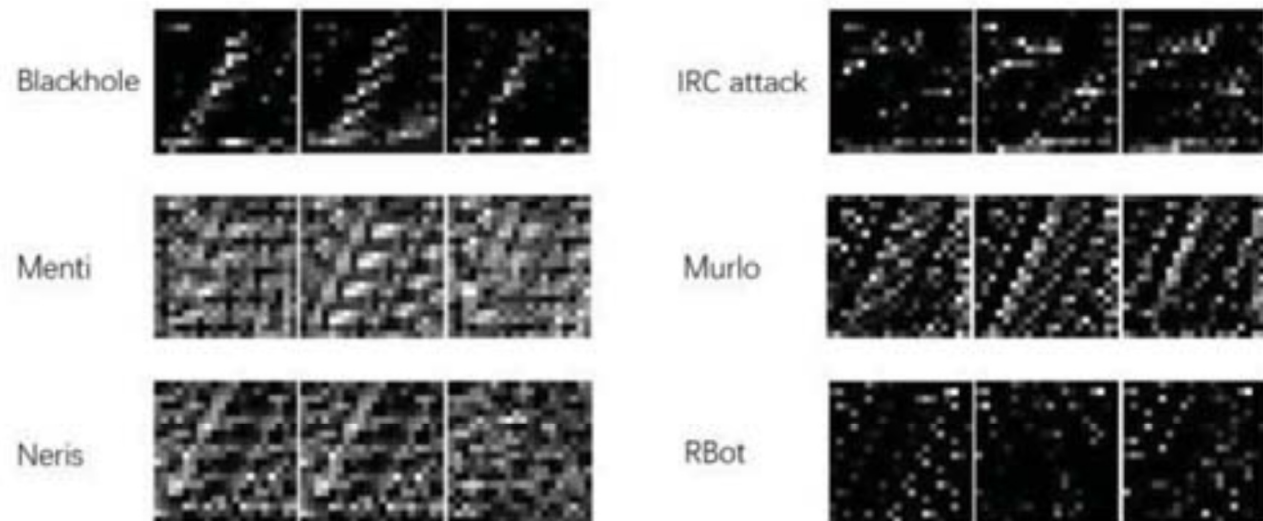


Fig. 8. Six samples of representation of the input data

Zhou, Z., Yao, L., Li, J., Hu, B., Wang, C., & Wang, Z. (2018). Classification of botnet families based on features self-learning under Network Traffic Censorship. *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, 1–7. <https://doi.org/10.1109/SSIC.2018.8556792>



## #2 Traffic counters (SNMP, etc)

- Routers keep track of how much traffic goes through each link
  - Packets, bytes
  - Periodically every n minutes
- Simple to use but limited in scope
  - Graph of link usage
  - Traffic matrix



# Traffic matrices

- Amount of data transmitted between every pair of nodes in the network
  - rows and columns are the nodes in the network
- Enterprise network, autonomous system:
  - points of presence (PoP)
  - links between PoPs – internal traffic
  - links between PoPs and external AS devices – external traffic
- Internal traffic matrix
  - traffic between the PoPs in the AS
- External traffic matrix
  - traffic between PoPs and external AS's





# #3 Flow measurements

- IP flows
  - Source, destination IP address and TCP/UDP ports (4 fields)
  - L3 header protocol (TCP, UDP, other)
  - Other info – ToS field, ??
- Keeps record of traffic for each flow
  - Packet, byte count on each direction
  - Duration, first/last packet timestamps, TCP flags
  - etc
- Tradeoff
  - simpler to use than packet traces
  - more information than counters
  - simply opening a web page can generate dozens of TCP flows



# Flow software

- Netflow
  - IETF IPFIX
  - Export directly from routers
  - Flow id, byte and packet counters, source/destination addresses and ports, duration, timestamps first and last packet
  - Router looks up entry in flow cache, updates counters or creates entry
  - Export flow records when idle (e.g. 15s)
  - Expire flows with TCP RST/FIN, timeout, or flow cache full
  - Sampling eases processing and memory, e.g. 1-N
- tstat
  - Open source for linux, process trace file or directly from eth0
  - 200+ flow features – IP, TCP, HTTP, TLS
  - Domain name from dns query, other fancy features
  - Also UDP and other protocols – see tshark, scapy.



# Active measurement tools

- Depends on what you want to measure
  - ping
  - traceroute
  - owamp (rfc4656), twamp (rfc5357)
  - Iperf (rfc6349, [iperf.fr](http://iperf.fr))
- Do it yourself
  - Raw sockets, scapy, TCP/UDP, application-level



# What to do with measurement data?

- Store for later query and processing
  - Send to ELK, other big data storage
  - Plot charts, do queries on past data, correlate between different data sources
  - Build historical dataset for learning AI models for different management tasks
- Use immediately once data is generated
  - Anomaly detection and diagnostic, security, traffic engineering, ...
  - Apply static rules, use pretrained AI model
  - Online learning, update AI model



# Performance and traffic engineering

- Apply TE when congestion arises
  - Admission control
  - Policing and shaping (non-conforming traffic, limit rates)
  - Queuing and scheduling policies
- On the device
  - Token bucket, fair queuing, etc
- On the network
  - ATM (dead), IP (intserv), label switching (diffserv, MPLS/IP)
- For protection
  - Pre-routed alternative paths,
  - don't wait until link failure to start routing algorithm
  - Start sending through alternative route when link failure detected
- For load balancing
  - Multipath TCP, DNS, reverse proxy



# Management and Operations of Networks, Services, and Systems

## Monitoring and performance

Ricardo Morla

FEUP – GORS/M.EEC, GRS/M.EIC

