TELE9752 Network Operations and Control

Week 10p: Performance

Outline

- Context
  - Other courses
  - References
  - FCAPS links
- Measuring performance
- Service Level Agreements (SLAs)
- Flow data and analysis
  - NetFlow
  - IPFIX

TELE4642: Network Performance

Aims: This course aims to develop an understanding of the tools and technologies for understanding and improving the performance of communication networks such as the Internet. It will introduce students to quantitative methods for loss and delay analysis in packet networks, using techniques from stochastic traffic modelling, Markov chains, and queueing theory. The quantitative methods will be applied to practical examples from communication protocol design, Internet switch architectures, Internet search algorithms, etc., and augmented with emerging qualitative techniques for providing quality of service in data networks.

Details:

- Runs in 1st Session
- http://www.handbook.unsw.edu.au/undergraduate/courses/2013/TELE4642.html
- http://subjects.ee.unsw.edu.au/tele4642/

References

- Comer
  - S 6.3 discusses SLAs
  - 7.9: Passive observation vs active probing
  - 7.10-20: capacity planning
  - Ch 11: Netflow
- Clemm
  - Ch 11: SLAs
  - Ch 8: Netflow/IPFIX

FCAP**S** links

Faults:

- SLAs include dependability with other performance metrics.
- Poor performance ≈ fault. -> performability <QG]

Configuration: Often configure to tune performance

Accounting: Record use for both accounting and measuring performance.

- Provider may vary tariff according to performance
- Vary tariff with time of day to reduce peak demand => help ensure performance

Security:

- Don’t tell competitors about performance!
- Usage data may be private => secure performance data

Outline

Measuring performance

- Dimensions
- Games
- Approaches: active vs passive
- Risks
- Hazards of packet measurement
- Presenting results

Measures of network performance

- Volume
- Throughput: Volume per unit time. (aka “bandwidth”)
  - Must define unit of time for averaging, e.g. 1b/s is smoother than 60b/min
- Delay: Time from sending to receiving (aka “latency”)
- Jitter = delay variation
  - e.g. caused by queueing delays rather than propagation delays (assuming fixed path)
- Loss rate: % info received / sent
- Dependability: e.g. MTTF, availability

Many dimensions to differentiate from competition & confuse customer.

Measurement games

Even when performance can be sort of measured and targets set, it is often all too easy to manipulate those targets. Say, for example, that your job is to process customers' complaints, and you're given a target that no customer should wait more than ten days for a reply. That means anybody who has been waiting seven or eight days becomes a priority, while you have nothing to gain by processing customers whose complaints have just arrived. If you aim at the target, your average response time might easily slow down. So then a new target arrives: Keep the average response time to a minimum. Responding to the incentive the new target gives you, you ignore any complaint that is difficult to resolve and send back quick letters when a response is easy. The average response improves but the customers with the most serious complaints never get a reply. Now a third target arrives: Hit *both* the previous targets. You can do that, sure, and present a handsome claim for overtime. So the fourth target restricts overtime. Now you send out a simple form letter: "Dear sir/madam, Thank you for your letter/email/fax/telephone call. I am afraid there is nothing we can do. Yours," etc.

- T. Harford: *The Logic of Life*, p. 92

How to measure performance

passively: Monitor service given to production traffic

- Realistic
- No extra traffic; won't interfere with real traffic

actively: Send probe messages & observe behaviour

- Probes can be discarded (unlike production traffic) => can observe extra aspects of network
  - e.g. ping & traceroute
- gives more control of load, e.g. ensure continuously present => good for monitoring availability
- artificial workload may not accurately match real workload (e.g. periodic vs bursty)
- adds load to network

Comer is quite biased about active vs passive: "The only way a manager can obtain a realistic assessment of end-to-end performance is to employ active probing" p. 82

Risks in testing

"On more than one occasion, I've dealt with a circuit with a perfect-looking interface that passed every standard ping test with flying colors and yet was useless. GIF files could not be moved across these circuits and Windows networking infrastructure protocols would not work over them. In each case, the problem was exactly the same: the circuit was provisioned incorrectly (with AMI encoding instead of the modern standard of B8ZS). This meant that the circuit could not pass large all-zero packets. The .gif image format uses a lot of large all-zero packets, as do the Windows network protocols."

M. Lucas: *Cisco Routers for the Desperate*, 2nd ed, section 5.3

Hazards of packet-level performance monitoring

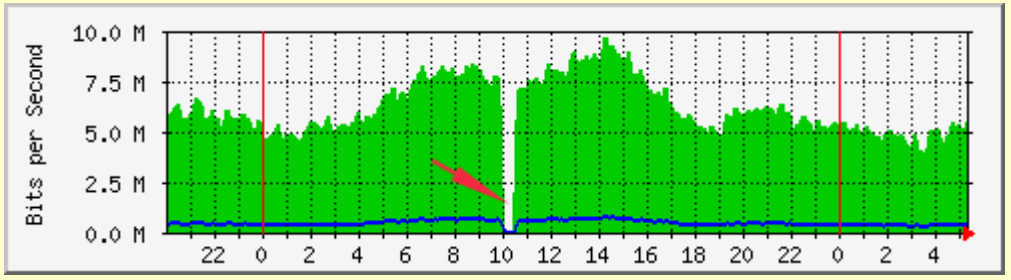
- Some aspects of performance only observable from **large flows** of packets, e.g.
  - caching of packet classification in router
  - traffic shaping

- Users care about *application* (e.g. HTTP) performance, which has complicated interactions with *packet* performance
  - e.g. TCP slows down in response to loss & other congestion indicators (e.g. ECN)
  - In fact they care about *perceived* performance of application => "Mean Opinion Scores" for voice/video

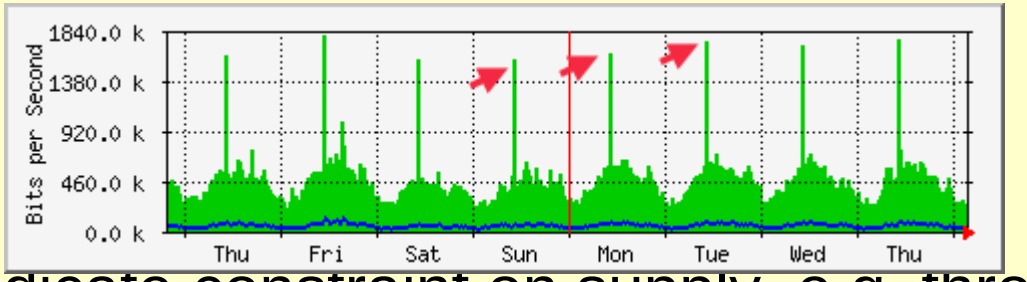
Presenting results of performance monitoring

Several tools exist for visualising network performance, e.g. Multi Router Traffic Grapher (MRTG) http://oss.oetiker.ch/mrtg/

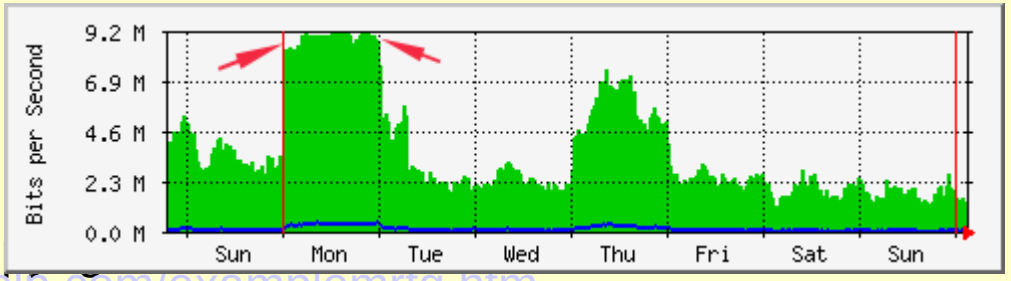
loss of service:



periodic spikes may indicate regular backups:



capped demand may indicate constraint on supply, e.g. throttling:



2EH Figures from http://www.mrtghelp.com/examplemrtg.htm

Outline

Service Level Agreements

"service level" = performance level +
services beyond network, e.g. support service

SLAs formalise the service that a provider will provide to a customer

- allows customer to compare service providers
- sets expectations for customer & limits obligations for provider
- forms a reference for comparison with delivered service

Sample SLAs:
AT&T Dedicated Internet Access
Amazon Simple Storage Service

2VZ See also: Why SLAs are redundant, but Zencoder offers one anyway

SLA components 1: Performance

- **Service performance**:
  - Captures network performance as discussed previously
  - *x*% of packets have delay less than *y* sec.
  - *z*% availability during normal hours
    - define normal hours (planned maintenance outside those hours)
    - define leadtime that constitutes sufficient warning from provider about planned outages
- **Responsiveness of support** from provider, e.g.
  - for events that affect service to *u* users, ISP will start acting within *x* seconds, resolve within *y* seconds, and give updates every *z* seconds.
    - lower delay for events that affect more users
- **Verification** of the above, e.g. by neutral 3rd party

SLA components 2: Other

- **Acceptable Use Policy (AUP)**:
  - Volume/rate of traffic that is permitted/expected
    - Offered load will affect provider's ability to serve this traffic & that from others.
    - e.g. phone: Busy hour call attempts (BHCA), call length
  - Purpose of traffic, e.g. business use, transit, copyright infringement
- **Contacts** for parties to the agreement
  - e.g. for receiving outage warnings, fault reports, requests for change
- **Penalties** for violating/breaking agreement
  - e.g. reduced fees, or compensation

Outline

- Flow data and analysis
  - NetFlow
  - IPFIX

Why consider flows?

- Have already covered RMON
- “Because it evaluates individual packets, a packet analyzer or RMON MIB can only provide aggregate statistics.” - Comer p. 164
  - Confusion: focus on *individual* restricts output to *aggregate*?!
  - Wireshark can definitely analyse high-layer units, e.g. TCP streams & HTTP request/response pairs
  - RMON traffic matrices may be unable to distinguish separate “conversations”, e.g. closing & then re-opening TCP socket.
  - RMON wasn't invented by Cisco
- Analyzers & RMON observe all packets => high processing & storage overhead for fast links.
  - sFlow [http://www.sflow.org/, RFC 3176] introduced **sampling**, e.g. every n^{th} packet, or randomly
  - Cisco provides Sampled NetFlow, e.g. Random Sampled Netflow or Deterministic NetFlow

Flow Information

Flows are similar to connections, but also applicable to non-TCP traffic.
Flows are identified by:

- **Socket**:
  - Addresses of source and destination
    - Flows are unidirectional => replies in separate flow
  - Protocol (e.g. UDP or TCP)
  - Ports on hosts
- **Interface** on observing device
  - augments addresses (which may be private) to create globally unique host identifier
- **Type Of Service** byte in IP header: affects router forwarding
- **Time** (optional), e.g. close flow after:
  - inactivity for some time (default = 15 sec)
  - active for long time (default = 30 min)
    - s.t. collector becomes aware of long flows without delay
  - indicators of higher-layer session closing (e.g. TCP FIN)

Uses for flows

- **Network management**: “Accounting” for traffic
- **Router operation**:
  - **Packet classification** problem: Given multiple packet header fields, how to quickly determine what to do with a packet (e.g. which router interface to forward it onto)?
  - TELE9751 describes high-speed implementations
  - Some involve **caching**: recent results stored in fast memory, comprehensive results stored in slower memory.
  - **NetFlow** may have originally been used to report on occupancy of cache. [Comer S 11.13] or perhaps with IPv6 flow label field to expedite classification.
    - Bonus mark on offer if you can definitively answer this!

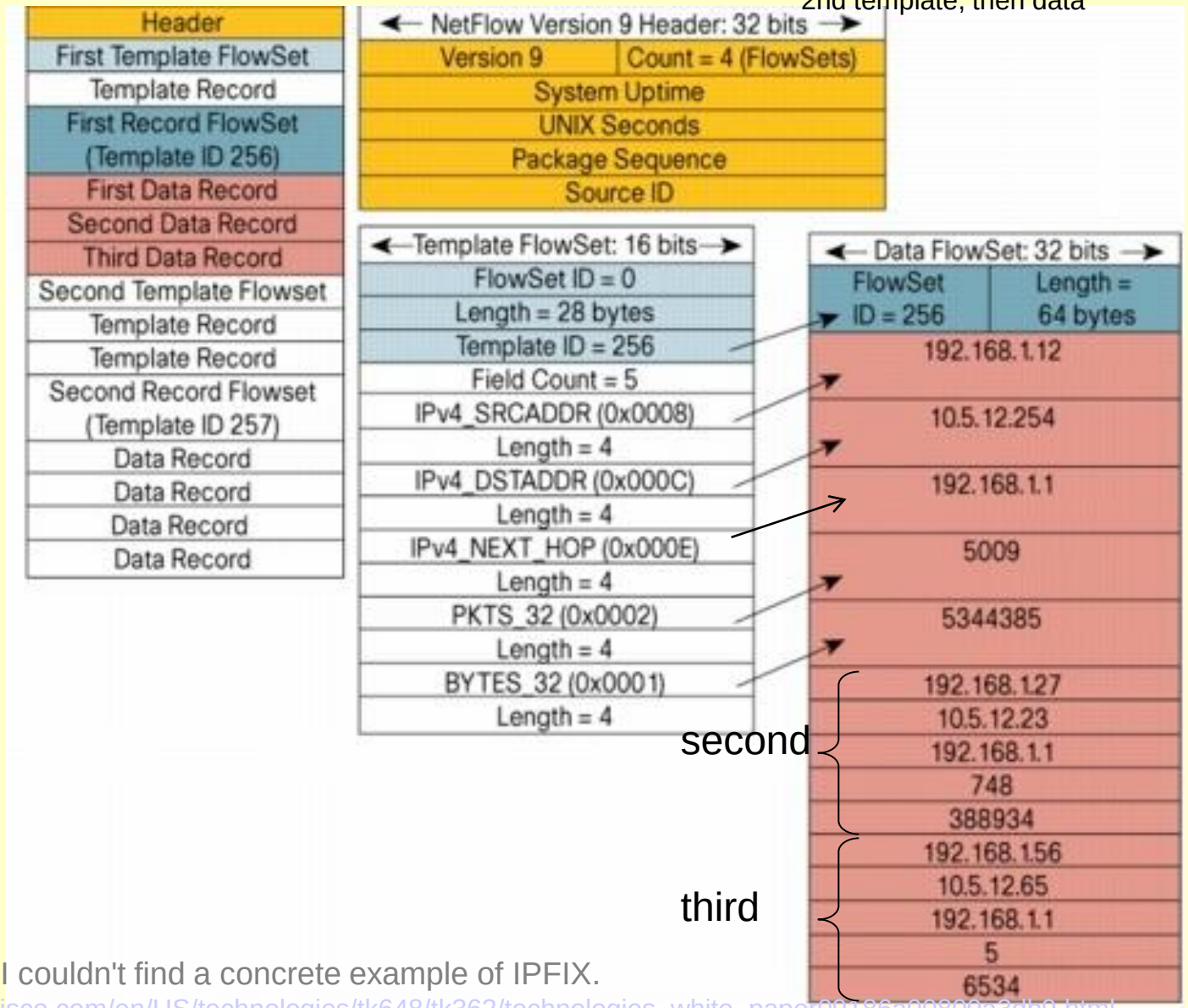
Flow info export protocols

- "Flow records" contain
  - keys that identify the flow (e.g. socket parameters)
  - statistics about flow (e.g. packet count, start/stop time)
- Avoid overhead of SNMP/syslog (due to ASN.1/BER generality) for common flow reports by using a new "flow info export protocol"
- **Cisco: NetFlow** (format documented in RFC 3954). NetFlow v9 forms the basis for IPFIX
  - NetFlow 9 & IPFIX provide for "templates" that define what statistics are collected in a flow record. Template is transmitted with flow records.
- **Internet Protocol Flow Information eXport (IPFIX)**
  - RFCs 5101-3, Jan. 2008
  - UDP/TCP/SCTP ports 4739 & 4740 (secured)

Netflow packet example

reporting on 3 flows

FlowSet = collection of records (of same type?)
Here: 4 = 1 template, then data, then
2nd template, then data



Despite trying to be vendor-neutral, I couldn't find a concrete example of IPFIX.
This Netflow example is from http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html

2XC

The end

- Week 11: No lecture
- Week 12: Student presentations