IP Network Management

G54ACC – IP and Up

Lecture 7

Recap

- IP provides best-effort packet delivery
- TCP and UDP provide transport over that
  - TCP reliable ordered byte-stream, UDP not
- Several adjunct control protocols in use
  - ICMP, routing, &c.

Contents

- Introduction
- Abstractions
- IP network components
- IP network management protocols
- Pulling it all together
- An alternative approach

Contents

- Introduction
  - What’s it all about then?
- Abstractions
- IP network components
- IP network management protocols
- Pulling it all together
- An alternative approach

What is *network management*?

- One point-of-view: a large field full of acronyms
  - EMS, TMN, NE, CMIP, CMISE, OSS, AN.1, TL1, EML, FCAPS, ITU, ...
  - (Don’t ask me what all of those mean, I don’t care!)
- From question.com:
  - In 1989, a random of the journalistic persuasion asked hacker Paul Boutin “What do you think will be the biggest problem in computing in the 90s?” Paul’s straight-faced response: “There are only 17,000 three-letter acronyms.”
- We will ignore most of them ☺

What is *network management*?

- Computer networks are considered to have three operating timescales
  - Data: packet forwarding [μs, ms]
  - Control: flows/connections [secs, mins]
  - Management: aggregates, networks [hours, days]
- …so we’re concerned with “the network” rather than particular devices or protocols
- Standardization is key!

Contents

- Introduction
- Abstractions
  - ISO FCAPS, TMN EMS, ATM
- IP network components
- IP network management protocols
- Pulling it all together
- An alternative approach

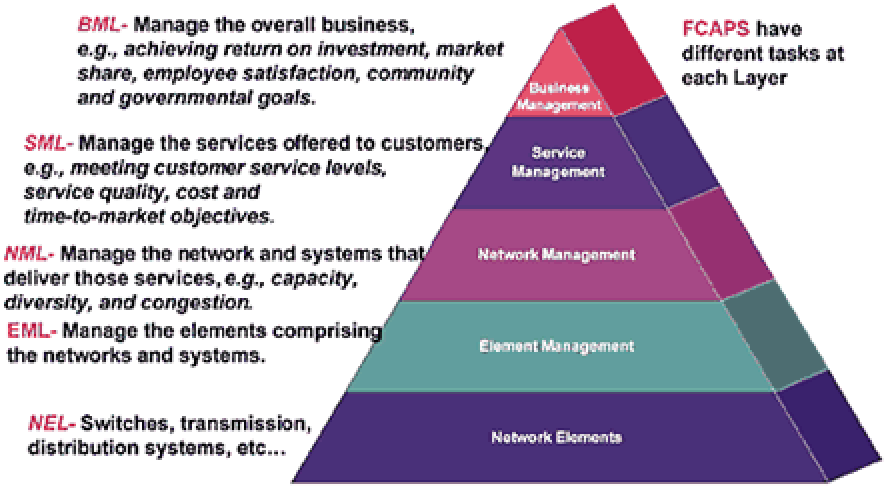
ISO FCAPS: *Functional* Separation

- **F**ault
  - Recognize, isolate, correct, log faults
- **C**onfiguration
  - Collect, store, track configurations
- **A**ccounting
  - Collect statistics, bill users, enforce quotas
- **P**erformance
  - Monitor trends, set thresholds, trigger alarms
- **S**ecurity
  - Identify, secure, manage risks

TMN EMS: *Administrative* Separation

- Telecommunications Management Network
- Element Management System

- “...simple ***but*** elegant...” (!)
  - (my emphasis)
  - (often the two go together)



BML- Manage the overall business, *e.g., achieving return on investment, market share, employee satisfaction, community and governmental goals.*

SML- Manage the services offered to customers, *e.g., meeting customer service levels, service quality, cost and time-to-market objectives.*

NML- Manage the network and systems that deliver those services, *e.g., capacity, diversity, and congestion.*

EML- Manage the elements comprising the networks and systems.

NEL- Switches, transmission, distribution systems, etc...

FCAPS have different tasks at each Layer

- NEL: network elements (switches, transmission systems)
- EML: element management (devices, links)
- NML: network management (capacity, congestion)
- SML: service management (SLAs, time-to-market)
- BML: business management (RoI, market share, blah)

Network Management

- Models of general communication networks
  - Tend to be quite abstract and *exceedingly* tedious!
  - Many practitioners still seem excited about OO programming, WIMP interfaces, etc
  - …probably because implementation is hard due to so many excessively long and complex standards!
- Basic “need-to-know” requirements are
  1. What should be happening? [c]
  2. What is happening? [f, p, a]
  3. What shouldn’t be happening? [f, s]
  4. What will be happening? [p, a]

Network Management

- We’ll concentrate on IP networks
  - Still acronym city: ICMP, SNMP, MIB, RFC ☺
  - Sample size: 10^2 routers, 10^5 hosts
- We’ll concentrate on the network core
  - Routers, not hosts
- We’ll ignore *“service management”*
  - DNS, AD, file stores, etc

Contents

- Introduction
- Abstractions
- IP network components
  - IP, networks, routers
- IP network management protocols
- Pulling it all together
- An alternative approach

IP Primer: You Know This...

- Destination-routed packets – no connections
  - Time-to-live field: allow removal of looping packets
- Routers forward packets based on routeing tables
  - Tables populated by routeing protocols
- Routers and protocols operate independently
  - …although protocols aim to build consistent state
- RFCs ~= standards
  - Often much looser semantics than e.g. ISO, ITU standards
  - Compare for example OSPF [RFC2327] and IS-IS [RFC1142, RFC1195], two link-state routeing protocols

So, how do you build an IP network?

1. Buy (lease) routers

   $1m? $2m? for a new, populated, backbone router!

2. Buy (lease) fibre

   Wayleaves = $$$
   Be a landowner!

3. Connect them all together

   Correctly.
   For now.

4. Configure routers

   Mwuhahaha.

5. Configure end-systems

   Someone else's can of worms.

Multiple router flavours

- Core
  - OC-12 (622Mbps) and up (to OC-768 ~= 40Gbps)
  - Big, fat, fast, expensive
  - E.g. Cisco HFR, Juniper T-640
    - HFR: 1.2Tbps each, interconnect up to 72 giving 92Tbps, start at $450k
- Transit/Peering-facing
  - OC-3 and up, good GigE density
  - ACLs, full-on BGP, uRPF, accounting
- Customer-facing
  - FR/ATM/…
  - Feature set as above, plus fancy queues, etc
- Broadband aggregator
  - High scalability: sessions, ports, reconnections
  - Feature set as above
- Customer-premises (CPE)
  - 100Mbps, maybe
  - NAT, DHCP, firewall, wireless, VoIP, …
  - Low cost, low-end, perhaps just software on a PC

Multiple router flavours

- Co
  -
  -
  -

50k

- Tr
  -
  -
- Cu
  -
  -
- Br
  -
  -
- Cu
  -
  -
  -

Cisco CRS-1
Multi-shelf system

Network Design

- Whose network?
  - ISPs, IXs, enterprise, campus
  - POPs, DCs
- Many designs: flat, hierarchical, hybrids, multiple scales
- Many constraints
  - Business
    - Backwards compatibility. Who to connect. Peering.
  - Technology
    - Power – directly (24x7 operation) and indirectly (cooling)
    - Port density *vs.* raw bandwidth
    - Software reliability
    - Hardware/software capability
      - Addressing schemes for scalability, summarization
      - Can’t run feature X with feature Y on vendor C in network size N
  - Connectivity/resiliency
    - “All core routers connect to at least 2 other core routers”
    - “All edge routers connect to at least 2 core routers”

Router Configuration

- Initialization
  - Name the router, setup boot options, setup authentication options
- Configure interfaces
  - Loopback, ethernet, fibre, ATM
  - Subnet/mask, filters, static routes
  - Shutdown (or not), queueing options, full/half duplex
- Configure routeing protocols (OSPF, BGP, IS-IS, …)
    - Process number, addresses to accept routes from, networks to advertise
- Access lists, filters, ...
  - Numeric id, permit/deny, subnet/mask, protocol, port
- Route-maps, matching *routes* rather than data traffic
- Other configuration aspects: traps, syslog, etc
- (Oh, and switch configuration is about as painful)

Router configuration fragments

```
hostname FOOBAR
!
boot system flash slot0:a-boot-image.bin
boot system flash bootflash:
logging buffered
logging console
aaa new-model
aaa authenticati
authentication l
aaa authenticati
aaa authorizatic
ip tftp source-i
no ip domain-loc
ip name-server 1
!
ip multicast-rou
ip dvmrp route-l
ip cef distribut
```

```
interface Loopback0
 description router-1.network.corp.com
 ip address 10.65.21.43 255.255.255.255
!
interface FastEthernet0/0/0
 descri
 ip add
 ip acce
 ip hel
 ip pim
 ip cgm
 ip dvm
 full-d
```

```
router ospf 2
 log-adjacency-changes
 passive-interface FastEthernet0/0/0
 passive-interface FastEthernet0/1/0
 passive-interface FastEthernet1/0/0
 passive-interface FastEthernet1/1/0
 passive-interface FastEthernet2/0/0
 passive-interface FastEthernet2/1/0
 passive-interface FastEthernet3/0/0
```

```
access-list 24 remark Mcast ACL
access-list 24 permit 239.255.255.254
access-list 24 permit 224.0.1.111
access-list 24 permit 239.192.0.0 0.3.255.255
access-list 24 permit 232.192.0.0 0.3.255.255
access-list 24 permit 224.0.0.0 0.0.0.255
                                                         00.0000.0000 0xD1 2 eq 0x42
                                                         ff.ffff.ffff
```

```
tftp-server slot1:some-other-image.bin
tacacs-server host 10.65.0.2
tacacs-server key xxxxxxxx
rmon event 1 trap Trap1 description "CPU Utilization>75%" owner config
rmon event 2 trap Trap2 description "CPU Utilization>95%" owner config
```

Router Configuration

- Lots of large, fragile text files
  - 00s/000s routers, 00s/000s lines per config
  - Errors are hard to find and have non-obvious results
  - Router configuration also editable on-line
  - Order matters!

- How to keep track of them all?
  - Naming schemes, directory trees, CVS, ssh upload and atomic commit to router
  - Perhaps even a proper database

This counts
as advanced!

- State of the art is pretty basic
  - Few tools to check consistency, design goals
  - Generally generate configurations from templates and have human-intensive process to control access to running configs
- Topic of current research [Feamster et al]

Contents

- Introduction
- Abstractions
- IP network components
- IP network management protocols
  - ICMP, SNMP, NetFlow
- Pulling it all together
- An alternative approach

ICMP

- Internet Control Message Protocol [RFC792]
  - IP protocol #1
  - In-band “control”
- Variety of message types
  - echo/echo reply
    [PING (packet internet groper)]
  - time exceeded [TRACEROUTE]
  - destination unreachable, redirect
  - source quench

Ping (Packet INternet Groper)

- Test for liveness
  - …also used to measure (round-trip) latency
- Send ICMP echo
- Valid IP host [RFC1122, RFC1123] *must* reply with ICMP echo response
- Subnet PING?
  - Useful but often not available/deprecated
  - "ACK" implosion could be a problem
  - RFCs ~= standards

Traceroute

- Which route do my packets take to their destination?
  - Send UDP packets with increasing time-to-live values
  - Compliant IP host must respond with ICMP “time exceeded”
  - Triggers each host along path to so respond
- Not quite that simple
  - One router, many IP addresses: which source address?
  - Router control processor, inbound or outbound interface
  - Asymmetric routes (return path != outbound path)
  - Routes change
- Do we want full-mesh host-host routes anyway?!
  - Size of data set, amount of probe traffic
  - This is topology, what about load on links?

SNMP

- Protocol to manage information tables at devices
- Provides get, set, trap, notify operations
  - get, set: read, write values
  - trap: signal a condition (e.g. threshold exceeded)
  - notify: reliable trap
- Complexity mostly in the MIB design
  - Some standard tables, but many vendor specific
  - Non-critical, so often tables populated incorrectly
  - Many tens of MIBs (thousands of lines) per device
  - Different versions, different data, different semantics
- Yet another configuration tracking problem
  - Inter-relationships between MIBs

IPFIX

- IETF working group
  - Export of flow based data out of IP network devices
  - Developing suitable protocol from Cisco NetFlow™ v9
  - [RFC3954, RFC3955]
- Statistics reporting
  - Setup template
  - Send data records matching template
- Many variables
  - Packet/flow counters, rule matches, quite flexible

Contents

- Introduction
- Abstractions
- IP network components
- IP network management protocols
- Pulling it all together
  - Network mapping, statistics gathering, control
- An alternative approach

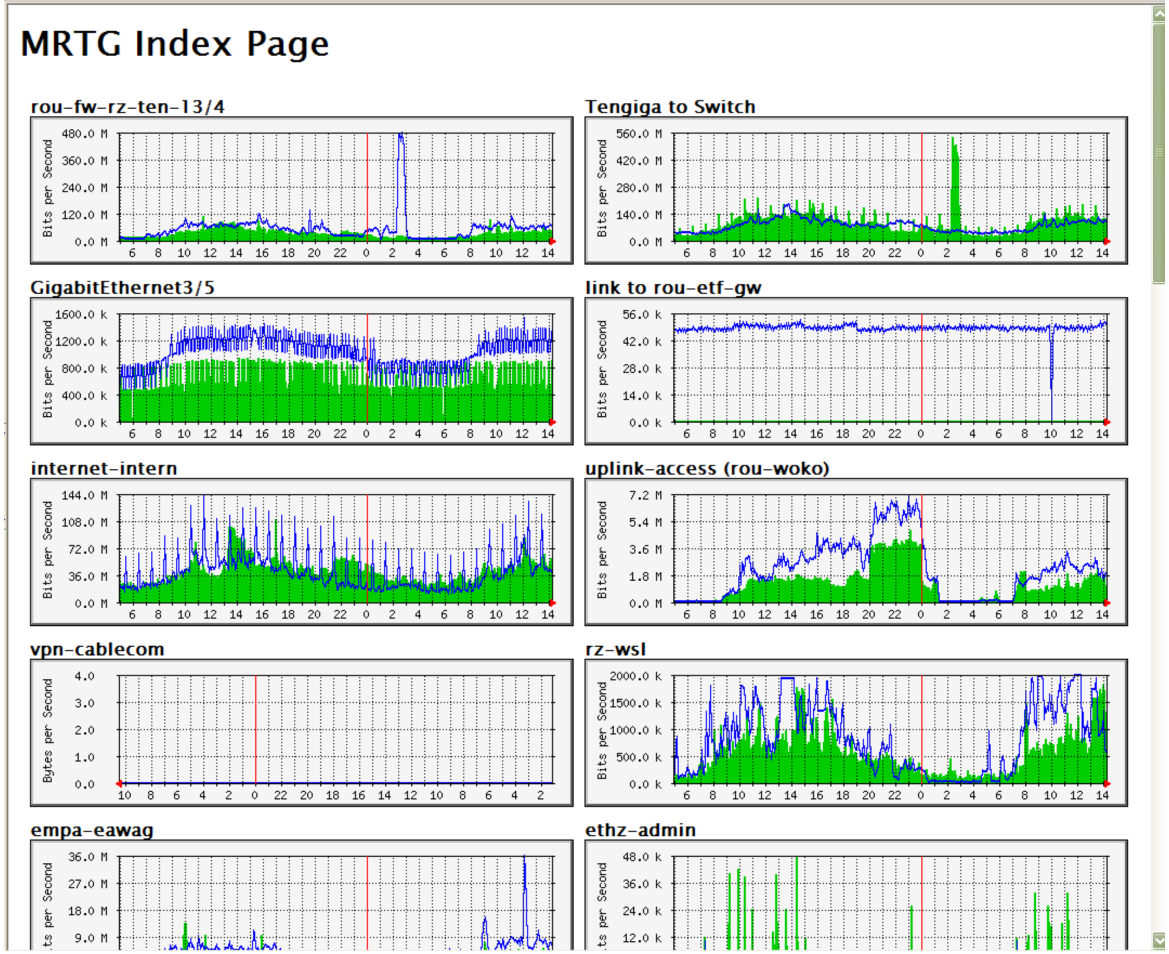
An Hypothetical NMS

- GUI around ICMP (ping, traceroute), SNMP, etc
  - Recursive host discovery
  - Broadcast ping, ARP, default gateway: start somewhere
  - Recursively SNMP query for known hosts/connected networks
  - Ping known hosts to test liveness
  - Iterate
- Display topology: allow “drill-down” to particular devices
- Configure and monitor known devices
  - Trap, Netflow™, syslog message destinations
  - Counter thresholds, CPU utilization threshold, fault reporting
  - Particular faults or fault patterns
- Interface statistics and graphs (MRTG)

NOC, NOC. Calling AT&T…

What are they all looking at?

http://www.stat.ee.ethz.ch/mrtg/



MRTG Index Page

rou-fw-rz-ten-13/4

Tengiga to Switch

GigabitEthernet3/5

link to rou-etf-gw

internet-intern

uplink-access (rou-woko)

vpn-cablecom

rz-wsl

empa-eawag

ethz-admin

An Hypothetical NMS

- All very straightforward? No, not really
  - A lot of software engineering: corner cases, traceroute interpretation, NATs, etc
- Correctness
  - MIBs may contain rubbish
  - Can only view inside your network anyway
  - Tunnelled, encrypted protocols becoming prevalent
- Efficiency
  - Rate pacing discovery traffic: ping implosion/explosion
  - SNMP overloading router CPUs
- Using NMSs also not straightforward
  - How to setup “correct” thresholds?
  - How to decide when something “bad” has happened?
  - How to present (or even interpret) reams and reams of data?

Summary

- Network management is the problem of designing, building, configuring and maintaining the network
- There are many older standards and processes around this
- IP has very little inherent support for it so numerous protocols have been developed and deployed