Miscellania

- Some topics that don't quite fit…

Lecture objectives

Broader Considerations for real-time applications:

- Systems Questions:
  - Scaling & Stability
  - Mobility
  - Management
- Non-technical Questions
  - economic and user aspects
    - Pricing and Provisioning
  - implementation context:
    - **Active Networks**
    - **MPLS/"Circuits"**

Scaling and Stability

References

•Vern Paxson, End-to-end Routing Behavior in the Internet
ACM CCR, vol. 26, no. 4, pp. 25-38, Oct. 1996.
http://www.acm.org/sigcomm/ccr/archive/1996/conf/paxson.html

•Floyd, S., and Jacobson, V.,
The Synchronization of Periodic Routing Messages
IEEE/ACM ToN, V.2 N.2, p. 122-136, April 1994.
href="http://www.aciri.org/floyd/papers/sync_94.ps.Z

~

Scaling (or Complexity) - 1

- All mechanisms that we add to IP Have some cost - we would like ideally, this cost to be O(C) (Order constant) - I.e. if we add QoS, the cost in terms of messages, router and end system memory, router and end system CPU should just be a constant, ideally! In practice though…
- Its likely that some mechanisms will be O(n), where n is the number of…
- end systems or routers - or can we do better?
- Diff-serve versus Int-serve is based around this...

Scaling (or Complexity) - 2

- So per flow-queues are at least going to have a data structure in a router per active pair (tree) of sender/receiver(s)
- Whereas per class queues have some data structure per class although edge systems may have to do per source policing and/or shaping - which implies that overall, we may have O(ln(n))
- Need tostate overall architecture to see overall system costs!

Stability - 1

- Ideally, Traffic, whether user or management (e.g. signaling, routing updates etc) should be stable.
- Conditions for stability complex - basically need to do control theoretic analaysis
- Even if oscillatory, should converge or be bounded, not diverge….
- Reasons for instability or divergence:
  - Positive Feedback
  - Correlation/phase effects...

Stability - 2

- End-to-end congestion control systems are designed to be stable - damped feedback
- Routing systems are designed to be stable - randomized timers
- QoS systems (especially call admision and QoS routing) need to be stable too.
- Needs careful thought and smart engineering…
- e.g. don’t want to do alternate path routing and admission control on same timescales.

Mobility

Reference:

- Anup Kumar Talukdar, B. R. Badrinath and Arup Acharya, "Integratedservices packet networks with mobile hosts: architecture and performance",Wireless Networks, vol. 5, no. 2, 1999
- Jarkko Sevanto, Mika Liljeberg, and Kimmo Raatikainen, "Introducingquality-of-service and traffic classes into wireless mobile networks",Proceedings of first ACM international workshop on Wireless mobile multimedia, October 25-30, 1998, Dallas, TX USA

- Links…
- Patterns…
- Resources...

Mobile 1 - Wireless Links

- Wireless links can have variable characteristics, e.g. delay, throughput, loss
- Offering hard QoS is hard
- GPRS and other wireless links offer shared media
- May be able to coordinate QoS via shared media MAC layer management and handoff management (see ISSLL work in IETF) - requires cooperation
- Opposite of trend on fixed nets (e.g. shared media LANs moving to switched approaches!)

Mobile 2 - Patterns

- Mobile access patterns may be quite different from fixed ones
- Simply don't know yet, but may entail lots more state refresh (e.g. re-sending RSVP path/resv triggered by moves)
- Mobiel multicast with source or sink moving may be complex (involve re-building tree)

Mobile 3 - Resources

- Some QoS approaches are based on the netwrk running largely underloaded
- e.g. EF and AF may only work for IP telephony if it constitutes a small part of traffic
- This is not the case on many wireless links today.
- Need to look at hard QoS schemes - particularly for low latency (e.g. interactive voice/games) - even down to the level of limited frame/packet sizes - leads to interleave problems...

Management

All this needs managing by someone, at the very least the policies need configuration…..

Management-1

- User account management
- QoS auditing
- MIBs for queues, signalling protocols, etc
- risk analysis and trend prediction tools
- security (authentication and privacy aspects of payment for qos - see next)

Pricing and Provisioning

Reference: http://www.statslab.cam.ac.uk/~richard/PRICE

Pricing 1

- If you don’t charge for QoS, won’t everyone just ask for first-class?
- What are the users paying for?
- What are they prepared to pay?
- If you do charge, how to stop arbitrage (rich buy all the bandwidth and then re-sell at different price).

Pricing 2

- Typically, access fee can cover actual cost of infrastructure
- Bill is often just an *incentive scheme* (to stop users hogging capacity in a class)
- Parameters:
  - time of day and duration
  - distance (geographic, provider hops, AS-count?)
  - capacity
  - delay (iff possible) and jitter control
  - Loss (possibly)

Pricing 3

- Can price by effective capacity
- Do we want to vary price with network conditions? (optimal in theory but complex - too complex for user - in practice) - *congestion pricing*
- security associated with payment and policing necessary
- Predictable bills are often more important than cheapest fare (c.g. mobile phones).

Provisioning

- Users don't like being refused access (prefer degraded service, but…)
- Need to dimension network for the user satisfaction and revenue levels
- Base on traffic measured. Look at frequency of overload or call rejection for RSVP…
- IP telephony - can (if pricing and patterns match) base on Erlang models…traditional - may not apply - e.g. either or both of call and packet arrival independence may be wrong...

Implementation Novelties

Active Networks &
MPLS

Active Networks

Reference: D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, G. J.Minden, "A Survey of Active Network Research, IEEE Communications Mag.,Vol. 35, No. 1, pp 80-86. January 1997

- Active networks subject of large DARPA program, and quite a few european projects.
- Interpose processing of user data in network path by dynamically moving code there….radical idea based in strong distributed computation
- Originated in observation that it has become very hard in telephony and IP networks to deploy new services of any kind due to scale (and inflexibility) of the infrastructure.

Active Networks 2

- Weak model just puts code in place at application level points -either call handling (e.g. dynamic singlaing protocol code -*switchware, switchlets* IEEE programmable networks work) or at application level relays (e.g. non transparent caches)
- Strong model - re-programs switches on the fly possibly per packet - packet header is now code for VM in switch instead of data for fixed program in switch.

Active Networks 3

- Jury is out on AN
- Looks like at least some ideas will make it through to prime time though….
- Main problems
  - with strong AN is code performance, safety and liveness
  - with weak AN is management - could be very useful for generalized VPNs though...

MPLS

- Datagrams Meets Circuits
- Based on strong idea of “flow”

Performance

- Getting data from source to destination(s) as fast as possible
- Higher data rates required for:
  - large files …
  - multimedia data
  - real-time data (video)
- **Fast forwarding**
- Not the same as QoS provisioning, but closely linked

Forwarding vs. Routing

- Routers have to:
  - maintain routes
  - forward packets based on routing information
- Forwarding:
  - moving a packet from an input port to an output port
  - make a forwarding decision based on route information
  - get the packet to an output port (or output queue) fast
- Routing:
  - knowing how to get packets from source to destination

IP forwarding

- Packet arrives (input buffer?)
- Check destination address
- Look up candidate routing table entries:
  - destination address
  - routing entry
  - address mask
- Select entry:
  - longest prefix match selects next hop
- Queue packet to output port (buffer)

Flows

- A sequence of IP packets that are semantically related:
  - packet inter-arrival delay less than 60s
- Flows may be carrying QoS sensitive traffic
- Many thousands of flows could exist when you get to the backbone
- Detect flows and use label-based routing:
  - make forwarding decisions easier
  - make forwarding decisions faster

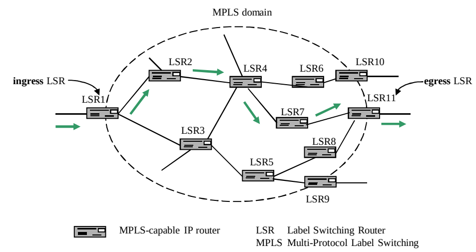
MPLS

- Multi-protocol label switching:
  - fast forwarding
  - IETF WG
- MPLS is an enabling technology:
  - helps scaling
  - increases performance
  - forwarding still distinct from routing
- Intended for use on NBMA networks:
  - e.g. ATM, frame-relay

MPLS architecture [1]

- IETF work in progress - requirements:
  - integrate with existing routing protocols
  - support unicast, multicast, QoS, source routing
- MPLS uses label-swapping
- Flows are labelled:
  - special shim header
  - can use existing labels in bearer technology (e.g. VCI)
- **LSR (Label Switching Router)**:
  - simple, fast link-level forwarding

MPLS architecture [2]



Label switching

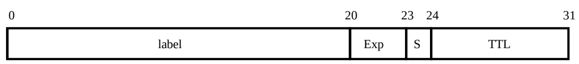
- Packet enters ingress router
  - lookup label: **Forwarding Equivalency Class (FEC)**
  - packet forwarded with label
- At next hop (next LSR):
  - label used in table lookup: **LIB** and **NHLFE**
  - new label assigned
  - packet forwarded with new label
- Saves on conventional look-up at layer 3
- Need label distribution mechanism

Labels [1]

- Label:
  - short
  - fixed-length
  - local significance
  - exact match for forwarding
- Forwarding equivalency class (FEC):
  - packets that share the same next hop share the same label (locally)
  - packets with the same FEC and same route: **streams**

Labels [2]: shim header

- Generic: can be used over any NBMA network
- Inserted between layer-2 and layer-3 header
- label: 20 bits
- Exp: 3 bits (use not yet fully defined - CoS)
- S: 1 bit stack flag (1 indicates last in stack)
- TTL: 8 bits



Label granularity

- IP prefix:
  - aggregation of several routes
- Egress router:
  - all IP destinations with common egress router for LSP
- Application flow:
  - per-flow, end-to-end
- Others possible:
  - e.g. host pairs, source tree (multicast)

Label distribution [1]

- Routing information used to distribute labels:
  - piggy-back label info on existing protocols?
- Performed by downstream nodes
- Each MPLS node:
  - receives outgoing label mapping from downstream peer
  - allocates/distributes incoming labels to upstream peers
- **Label Distribution Protocol (LDP):**
  - LDP peers (LDP adjacency)

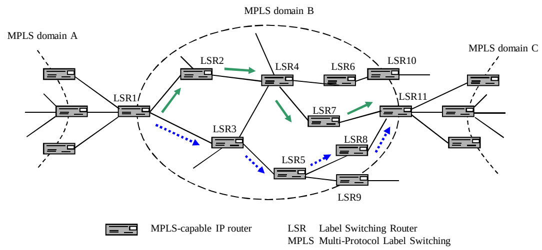
Label distribution [2]

- Distribution of label info from LSR only if:
  - egress LSR
  - LSR has an outgoing label
- **Downstream**: LSR allocates and distributes
- **Downstream-on-demand**: upstream LSR requests allocation from a downstream node
- Address prefix-based FEC/forwarding:
  - **independent** distribution: any node in LSP
  - **ordered** distribution: egress LSR

Label stacking [1]

- Two mechanisms:
  - equivalent to IP source routing
  - hierarchical routing
- Multiple labels are stacked by the ingress LSR
- LSRs along the route can pop the stack:
  - makes forwarding even faster

Label stacking [2]



MPLS-like implementations

- Control-based:
  - tag-switching: cisco
  - ARIS (Aggregated Routing and IP Switching): IBM
  - IP-Navigator (Ascend)
- Request-based: RSVP
- Traffic-based:
  - IP switching: Ipsilon
  - CSR (cell switch router): Toshiba
- Many others …

Other performance issues

- Router architectures
- Fast route-table lookup
- Fast packet-classification (QoS)
- Better address aggregation (e.g. CIDR, IPv6)
- Traffic engineering (differentiated services)
- Faster boxes or smarter software?

Summary

- Reference: Scott Shenker, "Fundamental design issues for the future Internet",IEEE J. Selected Areas Comm, 13 (1996), pp 1176-1188
- QoS isn’t that simple!
- Push something out of one part of the architecture, it will show up somewhere else
- e.g. if you remove statelessness by ading RSVP, you need to do congestion control of signaling
- e.g. if you remove adaption by adding connection admission (e.g. for TCP), users start adapting.