Miscellania

DigiComm II

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"**

DigiComm II

The Internet is a complex machine - probably now the most complex machine that human's have built to date, now that it has outgrown the telephone network in capacity and complexity. A number of things are impacted when we try to make the Internet service at the network layer more rich. Firstly, we have to consider the impact on scaling of any such a refinement. The Internet scales well - this is evidenced by the fact that it is growing at roughly 100% per annum in terms of numbers of end systems, numbers of end to end services, and raw backbone capacity to accommodate all of this. This ability to scale is all due to the decentralized nature of the control systems for the network (naming, addressing, routing, caching), and the lack of more than very simple peering rules between user and provider, and between access and core, and core and core network service provider.

Adding QoS will affect this scaling. Similarly, the Internet is relatively stable - the protocols in use have been subject to some refinements over the past 15 years which have included factoring in control theoretic results to improve stability.

The facilities used to access the Internet are evolving even more rapidly right now than the services, and mobile access is one area that most pundits expect to see massive growth over the next 5 years - to date there are some 250M Internet users, and some 250M mobile phone users -the demographics of these groups are quite similar - there's certainly a massive overlap - wireless link technology brings interesting problems for basic IP performance (well, for TCP and RTP anyhow) without having to start worrying about QoS.

Payment is always a thorny question - the Internet appears to make economic sense, although the models for funding the infrastructure continually evolve. Service discrimination will require more complex payment options.

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

~

The first part of the net subject to control to avoid instability was the end-to-end protocol area. TCP uses a congestion avoidance and control strategy that is now part of general requirements from many ISPs for baseline source behaviour in the best effort arena. Other protocols (e.g. reliable multicast transport, and even some rate adaptive audio and video sources, e.g. from Real Networks) now follow this design goal. Given careful implementation, this leads to a network sharing model that approximates to weighted fair share on a single ISP service.

The routing systems are also dynamic, and as such could under high event loads potentially become unstable - this would be exactly when least wanted (e.g. when there are correlated outages) - work has been done to try to understand stabilising this part of the network control system too..

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...

Many Internet papers and articles use the term “scalable” when describing a protocol or a part of the architecture.

What they actually mean is that this component makes sense economically in terms of its computational complexity with regard to related inter-dependant components. For example, the capacity of the backbone scales as linear or better in the number of end systems, and the routing table in the Internet as it currently operates scales with the number of end systems, n, as better than ln(n) - this means that as we add end systems and gain revenue, we can afford to deploy more capacity, and the routers’ memory in the core do not need continually upgrading at more than this rate. Address aggregation means that routing updates do not grow rapidly either.

If we add new services in the IP layer, how will they affect this scaling?

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!

DigiComm II

Current routing tables reflect the address hierarchy and to some extent nowadays due to address allocation policies , topological hierarchy (at least in some providers).

Flow based forwarding may require state in the routers for packet scheduling that grows beyond this rate depending on the approach.

Integrated services can be naively implemented with per source/destination state - this would mean for example on the UK-US academic link (2 OC3 links, operating at 300Mbps), some 4000 entries at the busy hour (assuming that the number of flows requiring QoS was the same as the number of TCP Flows currently seen there). This is not necessarily difficult to implement, but would require quite fast memor for per packet lookup, and would be required on all routers on any Int-serv capable path.

Diff-serve might more easily be implemented with priority queues (at least for low utilisation of higher priority levels) in the core routers, although edge devices would need more per-flow specific state - however, edge devices would also handle far less flows- so while overall across an entire ISP, the state might grow, the additional memory per router is potentially a lot less.

Of course, if accounting is needed per flow (for billing, or for trend analysis and provisioning), then we may be back to the same position as int-serve.

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...

DigiComm II

Oscillation is not necessarily a bad thing (maybe ought to see if John Harrison said anything about this:-)

However, in networks, traffic that oscillates (e.g. alternates routers, alternates between low and high rate) causes knock-on effects - for example to accommodate a delay, we would need to provision for the higher rate (on multiple links). Oscillations tend to synchronise - this may lead to divergence of load, in which case we will get a service failure.

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.

DigiComm II

So schemes are put in place to damp out oscillations

avoid positive feedback
damp delayed feedback
randomize timers to remove phase locking
etc etc

As the system gets larger, higher level synchronisation effects become possible - e.g. web page announcement on bulletin board or TV or radio program causes a number of users to direct browsers towards the same server in a loosely synchronised fashion familiar in telephone networks as the flash-call or mothers day (or 11.59:31/12/1999) problem.

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...

DigiComm II

Two common wireless networks:

wavelan - wireless ethernet

GSM/GPRS - mobile telephones

Both offer IP. Neither offers IP QoS yet. Problem is that they are shared media - the appropriate solutions are link level combined with IP level.

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!)

DigiComm II

IETF has working group called

Integrated Services over Specific Link Layers

working on how to map QoS from IP level services onto link layers.

In the point to point case it is fairly simple and works well at the pure IP level provided there is no link layer multiplexing of other services in an unknown way. On shared media links, it is less simple, and depends on adding facilities to the MAC layer. In the IEEE 802 committee there are working groups doing this for int-serve like QoS (q) and diff-serve type Class of Service (p).

For wavelan the appropriate solution may be the Subnet Bandwidth Manager

For GSM, the IP level sees a fixed telephony link so there is no special problem (apart from those associated with low speed links) but for GPRS the problem is complex, and handover makes life even harder

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)

DigiComm II

Building predictable QoS is not just a matter of implementing the signalling, call admission and scheduling - the user doesn't want to be told "no" too often! So we need to provision the network - this relies on some idea of the traffic pattern and its dynamics. Unfortunately, we have practically no idea of what this will look like for mobile, wireless IP ! However, since a large amount of wireless access will be confined to access networks (whether wireless ethernet or wireless telephony) it is likely that initially, we will not have to deal with the wireless end-to-end QoS problem in the Internet.

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...

DigiComm II

Wireless links are generally slower than their fixed counterparts due to physics- in the extreme this means that there is no choice - capacity is a genuine scarce resource.

The first place that this impacts is on packet headers and packet sizes and their effect on delay - at 10Kbps, a 40 byte header (TCP/IP or RTP/UDP/IP) is a serious liability. Luckily, this problem has been tackled, at least partly, already on low bandwidth fixed network links of the past, by applying hop-by-hop header compression. Typically, this reduces the header size by just over 1 order of magnitude, down to around 3 octets, at the cost of some state in the routers at each end of a wireless hop. The state is soft (I.e. if the route changes or the state is otherwise lost, it is recovered at some delay cost, on the next packet exchange.

Some approximations to QoS work by assuming that a link runs very underloaded, clearly these will not work well in many wireless domains.

Management

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

DigiComm II

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)

DigiComm II

Each new QoS component needs a management information base. This adds cost to the network management systems.The inter-relationship of some management information is also stressed by adding QoS - for example, one would like to relate usage to topology, and call rejections to network faults and errors.

Pricing and Provisioning

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

DigiComm II

It is often said that the Internet needs pricing. This is clearly a rather naïve statement - the Internet is priced - typcially users pay commercial ISPs for access per month at the rate appropriate to their access link. The traditional model then has been a flat lease fee for the right to access anywhere in the Internet, only limited at the ISP ingress by the speed of ones access line. Of course, traffic conditions and server performance would also cause ones “mileage to vary”.

Recently (particularly in the UK with massive de-regulation of the telecommunications industry), a number of ISPs have emerged who have been referred to as “free”. This is again rather a misnomer - there are two main ways that such ISPs have raised revenue:

1. For dial up users, some ISPs get a fraction of the telephone call charge - this is part of normal inter-LATA arrangements n some countries, particularly where local calls are not free, this is a very effective partnership between ISP and local telephony company since the access phone company gets more calls for IP access (and therefore more revenue) and the ISP gets to avoid having to build a billing system. It does usually require call-line-identification to match the $$$ to the account.

2. The server side pays - typically gaining revenue from advertising - this model mimics commercial television and suits highly commercial content providers.

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).

DigiComm II

Costs:

Capacity is non linear priced in line rate, distance, and time!

Routers are non-linear priced in CPUI, Memory, Line Cards, performance (over time too)

Access network has been slowest to change (either academic LAN + router based, or domestic/small office, dial-up based) - this has been stabilised in price by the near monopolies -its now changing due to use of other technologies like fast wireless, DSL replacement of dumb modems, and cable modem use over terrestrial TV network plant. The cost of these networks is lower than the original phone net, and offers instead of a maximimum of 56kbps, somewhere in the range 10kbps (GSM) to 20Mbps (VDSL).

We live in "interesting times"

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)

DigiComm II

Bills for QoS are rarely to pay for actual costs. Typically they are used as part of marketing and control, at least in much of the economics of networks literature. The normal way to set a tariff is to calculate (dynamically) what percentage of users will pay how much for the capacity an ISP does have, and offer the rest as best effort.

If an ISP offers VNP (or virtual leased line like services) then a simpler scheme is to offer pro-rata tariff based on actual costs - since the ISP buys wholesale, but is offering this serive retail, this will more than cover real costs.

However, to make life seem more palatable, it is usual in the networking literature to consider a set of plausible parameters for a QoS based bill:

- •time-of-day
- •duration
- •capacity
- •delay, jitter
- •loss
- •priority (for pre-emption)

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).

DigiComm II

In many dynamic shared systems (e.g. road transport) congestion pricing and e-payment are offered as a mechanism for achieving an optimal market. At the extreme, this can be an auction per packet for capacity on offer. At a lesser extreme, this might be done in a bourse on a daily or weekly or monthly basis by ISPs who then offer a tarrif for their QoS bill-of-fare for the end user to pick from.

This has been seen in ts latter form to work for international and mobile telephony in the last few years. It remains to be seen if it will work for IP. Note that telephony only offers 1 service (just like standard IP) and not really a QoS range. Adding in consideration of the user and the users' understanding of the price and performance may undermine such approaches - the INDEX Project at Berkeley in the US has been studying this recently, but we don't have any really widely applicable solid results yet.

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...

DigiComm II

Once an ISP offers per-flow QoS, they have to compete with other such ISPs on the basis of call-blocking probability. This is calculated by looking at the set of source traffic models, and the traffic matrix (and its evolution over the working day and over time), and computing where to allocate more capacity.

Problems for IP exist here:

- •Traffic ource model of TCP is understood to some extent, but not in the large
- •Traffic model for RTP is not obvious
- •Traffic matrix is not at all obvious
- •Correlated calls seem much more likely than in telephone networks
- •aggregate traffic behaviour is not analytically understood.

Current solutions to provisioning often are based on monitoring the churn in the customer base and triggering acquisition and deployment of more capacity whenever that exceeds some threshold.

Implementation Novelties

Active Networks &
MPLS

DigiComm II

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.

DigiComm II

Active Networks includes quite a wide range of ideas now, ranging from servlets, through packet headers containing insutrvtions and not just data, through to agents.

It's a "hot research topic", but not without controversy and critics.

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”

DigiComm II

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

DigiComm II

Multi-Protocol Label Switching (MPLS) is a technology that is currently being developed within the IETF to allow **fast-forwarding** of IP-packets. The IETF charter for the workgroup is:

http://www.ietf.org/html.charters/mpls-charter.html

There is now a high performance demands from network users with the use of large files, multimedia applications and real-time flows.

MPLS is a label-based scheme that allows packets to be forwarded based on short, fixed-length labels rather than using the normal IP-routing table lookups. This is not to say that MPLS is a replacement for routing, indeed it relies on the normal routing protocols to allow the correct assignment and distribution of labels. Essentially, the mechanism attaches a label to an IP packet and uses this to make hop-by-hop forwarding decisions, bypassing the normal destination-based, longest-prefix match IP-forwarding mechanism at a router.

The aim of MPLS is to provide a fats forwarding mechanism. This should also help in provisioning of QoS capability in an IP network.

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

DigiComm II

IP routers have to perform a number of tasks, including maintaining routes. This is achieved by the exchange of routing information using routing protocols like OSPF (Open Shortest Path First) and BGP (Border Gateway Protocol). Based on this routing information, the router makes **forwarding decision**, i.e decides which of its output ports a packet must be sent to.

The problem of making fast **forwarding decisions** is inherent in any datagram network, such as IP networks. A description of the task is quite simple – to move a packet from an input port to an output port as fast as possible. However, there is a process involved in making the forwarding decision that is a major factor in determining the overall performance of the router.

The task of **routing** is a more far-reaching task in that it is achieved by the interaction between and involvement of all the routers in a network, whereas the task of forwarding is specific to an individual router.

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)

DigiComm II

To make a forwarding decision for an IP packet the following steps take place at a router:

1. a packet arrives at an input port and the packet may need to be buffered

2. the router must read the destination address of the packet

3. based on the destination address, the router selects candidate routing table entries, and for each candidate entry, saves the next hop address, the address mask for the address and the output port for that entry

4. after all the candidate entries have been found an entry must be selected by using the longest prefix match using the routing entry address mask and the destination address in the packet

5. when the appropriate candidate entry has been selected, the packet is placed on the appropriate output queue

Steps 3 and 4 in this process may require the consideration of other information such as routing metrics, policy-based routing, security information, etc. In general, this may slow down the forwarding process, although clever caching and recent developments in table-lookups and packet classification can help.

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

DigiComm II

A packet **flow** (or stream) can be thought of as a sequence of packets that are semantically related. the relationship is application specific, e.g. it could be all packets with the same source and destination address. A flow exists if there are at least two packets that are semantically related and their inter-arrival delay is 60s or less.

Flows may be carrying traffic that is QoS sensitive, e.g. audio or video data, or has other real-time constraints.

When identifying flows, it must be appreciated that there may be a need to have aggregation capability for flows, else in the backbone there may be many thousands of flows that need to be monitored and maintained.

If flows can be identified then it seems that the forwarding decision process need only be executed once, for the first packet in the flow, as all packets for the same flow/stream will be subject to the same forwarding decision. So, the impetus for MPLS is that if the forwarding decision is initially executed using normal routing mechanisms and then identified with a short, fixed-length label, the much simpler task of matching labels enables faster and easier forwarding for subsequent packets in the stream/flow.

There are several names for such mechanisms including:

• cut-through routing

• short-cut routing

• layer-3 switching

and specific vendors have their own names for proprietary schemes.

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

DigiComm II

The Multi-Protocol label Switching (MPLS) WG of the IETF is seeking to define a standard that will support fast-forwarding mechanisms.

It is intended that the use of MPLS in place of traditional IP forwarding will allow better performance and scaling in certain IP network scenarios. Its is intended that such mechanisms will help scaling an and performance of IP networks in certain environments, i.e. where it is likely that the layer-2 technology will offer a faster forwarding mechanism than the layer-3 forwarding of IP.

MPLS is designed to be complementary to existing routing mechanisms. Indeed, routing information is used to establish the forwarding entries used by MPLS.

Although independent of any particular bearer technology and any particular layer-3 technology, there is particular interest in finding MPLS solutions tailored to provide IP-over-ATM and IP-over-FR (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

DigiComm II

The MPLS WG work is still in progress. No RFC documents had been produced at the time of writing. The main requirements for MPLS are that:

• it can be integrated with existing level-3 routing protocols

• will support unicast and multicast

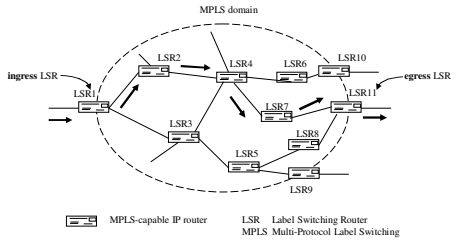
• has the potential to be used as a QoS control mechanism

• will enable traffic engineering mechanisms such as source routing and hierarchical routing

The MPLS architecture is based around the label-swapping paradigm that is used in virtual circuit (VC) networks. Flows are labelled with a short, fixed-length label that is used for identifying flows. The label value can be part of a generic shim header (defined by the MPLS WG), or could be inserted into existing header fields in various technologies, e.g. the VPI/VCI in ATM cells.

The main network element that allows MPLS use is the Label Switching Router. This is a MPLS-capable IP router that can effectively perform link-level forwarding of IP packets.

MPLS architecture [2]



DigiComm II

The MPLS enabled network consists of **Label Switching Routers (LSRs)** that use label-swapping to perform packet forwarding. A LSR is a MPLS-capable IP router. An IP packet flow/stream enters the MPLS network – **MPLS domain** – via an **ingress** LSR. Generally, this is where the IP packet is given a label, but the MPLS work does not preclude direct labelling by individual hosts. Once the packet is labelled, it is forwarded to the next LSR along the **Label Switched Path (LSP)**. At the point where the packet leaves the MPLS network, the final router, the **egress** LSR, forwards the packet towards its final destination. The label may be removed at the egress LSR, or it maybe removed by the penultimate hop LSR, and this is a matter for local configuration. If the packet is being forwarded to another MPLS domain at the egress router, then another label value may be assigned. In , we see an example of an MPLS network, with a packet flow/stream marked by arrows. All packets in the same flow/stream are forwarded along the same path. LSR1 is the ingress LSR and LSR11 is the egress LSR. The LSP for the flow/stream is given by the concatenation of the LSRs through the MPLS network, i.e. {LSR1, LSR2, LSR4, LSR7, LSR11}.

With respect to the forwarding direction, LSRs have **upstream** and **downstream** relationships, e.g. LSR2 is upstream from LSR4 and LSR7, and LSR4 and LSR7 are downstream from LSR2.

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

DigiComm II

As the packet passes to the ingress router, a normal IP forwarding lookup is performed for that packet. However, the result of the lookup is to identify:

• the **Forwarding Equivalency Class (FEC)** for the packet

• the label that is associated with the FEC

The label is then attached to the IP packet and forwarded to the next LSR in the LSP. At the next LSR, the label is used in an exact-match table lookup to identify:

• the next LSR in the LSP

• the new label value

The new label is written to the packet and the packet is forwarded along the LSP. This process continues until the egress LSR (or the penultimate hop LSR) where the label is completely removed and the packet continues on its journey (unless forwarding to another MPLS domain). Within the MPLS network, the overhead of the normal IP-level lookup (using longest-prefix matching) is avoided and this should improve performance. The key to this mechanism is the efficient allocation, distribution and maintenance of labels.

LSRs maintain a **Label Information Base (LIB)** containing label-FEC bindings and a table of **Next Hop Label Forwarding Entries (NHLFEs)** that are indexed using the label value.

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**

DigiComm II

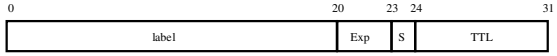
Labels are short, fixed-length identifiers used to identify packets that will receive the same forwarding treatment. Their value has only local significance and they are used in exact-match lookups to identify how a packet should be forwarded. Labels effectively identify Forwarding Equivalency Classes (FECs). Packets belonging to the same FEC will receive the same forwarding treatment. Label-based forwarding uses a simple exact match mechanism for making forwarding decisions.

The labels identify packets with the same FEC. Its is possible for packets from different flows/streams to share the same FEC for part of their journey, i.e. an FEC does not map one-to-one with a flow/stream. For example, packets from different sources headed from the same destination might share the same FEC for some of the last hops of their journey.

The MPLS terminology is to use the word stream for identifying packets in transit that have the same FEC.

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



DigiComm II

The shim header sits between the IP header and level 2 header. The shim header is 32 bits in order to aid fast processing in modern hardware platforms. The Experimental bits may be used as Class of Service (CoS) identifiers, but this is for further study. Labels can be stacked (explained below) and the S bit is set when this label is the final one in the stack. The TTL identifier is copied from the IP packet header at the when the label is added and its value is decremented at each LSR, as the TTL would be decremented in a normal IP packet at each router hop. When the label is removed, the TTL value in the label is copied to the IP packet header.

Specific technologies, such as ATM and FR, will have different label encodings mechanisms that take advantage of existing header fields, e.g. the FR DLCI or the ATM VCI/VPI.

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)

DigiComm II

The label identifies a FEC and the label may have a granularity that is chosen to suit particular needs of a particular administration. In fact, the way the semantics of a label (effectively the description of the description of the FEC) is chosen may reflect some policy controlled by requirements based on, Quality of Services (QoS), traffic engineering, cost or any other administartion specific criteria. The granularity chosen could be:

• **destination IP-prefix:** effectively each routing table entry has a FEC and so its own unique LSP. This may be an easy policy to implement based on information gained from the normal routing information carried in IP routing protocols. This may not scale where the label-space is limited or where there are large numbers of routes.

• **egress router:** all packets heading for the same egress router of an MPLS network might be given the same FEC. This may allow aggregation of several routes and would scale better than using the destination IP-prefix, but is a much more coarse-grained FEC, and offers less control of individual traffic streams/flows.

• **application flow:** this level of granularity could be used in conjunction with a resource reservation protocol or other QoS control mechanism.

MPLS does not constrain any policy-based definition of the FEC or the label granularity, and indeed the current documents specify other granularities such as host-address pairs, network-address pairs, source specific tree and shared tree, where the latter two are applicable to 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)

DigiComm II

Genrally, the label distribution mechanism for a flow starts at the egress router and is passed upstream, i.e. the distrbution is perfomed by the action of downstream nodes.

Each LSR receives a lebl fromdown stream with information about the FEC to which that label is bound. For that label, it allocates and distributes its own label upstream and maintains the incoming-outgoing labels mapping.

To allow distribution of the label information, in general, there are several options. The information about label values and FECs could be piggy-backed onto messages generated by existing routing protocols. This would mean that no new protocol is required. However, there may be a large difference in the operation of the level 3 network and the level 2 network, especially with respect to the timescales over which forwarding decisions or routing changes are made. So, a **Label Distribution Protocol (LDP)** is also being designed to allow label information to be distributed between LSRs. LSRs are said to engage in **LDP peering** and have **LDP adjacency** when they are using the LDP to communicate.

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

DigiComm II

Generally, label distribution can occur in two ways:

• **downstream-on-demand:** a LSR recognises a packet for a FEC and asks its next hop to provide a label binding for the FEC before assigning its own label binding

• **downstream:** a LSR recognises a FEC assigns a label binding and advertises this label binding to its peers

Where FECs correspond to address prefixes that are used by IP-routing, LSP control can occur in two ways.

• **ordered distribution:** an LSR can only distribute a label binding if it is the egress router for the stream, or if it already has a downstream label binding for a FEC.

• **independent distribution:** an LSR recognises that a stream belongs to a particular FEC, starts MPLS forwarding, and advertises its label binding to its peers, even though it may not have received a label binding from a downstream LSR.

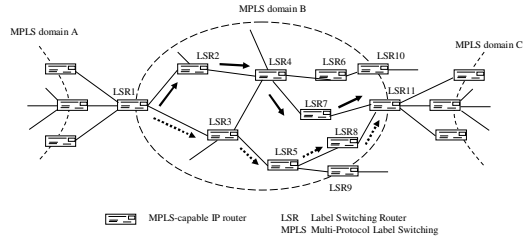
Note that for label-based forwarding to work across the entire domain, all the downstream nodes must be aware of the stream and the relevant FECs, and have a label binding for the FEC. So, to ensure that downstream nodes have this information, it is advisable for label distribution to begin from the egress node, and this would ensure that a packet is never forwarded from an upstream LSR unless a downstream LSP exists. So, ordered distribution is preferred, in general. Although it may have a higher latency in some cases, it is more robust to looping, and can be more easily configured for QoS-based or policy-based forwarding. Independent distribution follows the conventional IP-forwarding model, but relies on the fact that the LDP algorithm must allow fast convergence.

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

DigiComm II

Label stacking [2]



DigiComm II

Labels can be stacked to allow hierarchical forwarding or mechanisms akin to source routing. Multiple labels are used. The label nearest the IP-header is at the bottom of the stack, and is said to be the lowest level label. The label that is used is always the one at the top of the stack.

Here, we have several MPLS domains (these may be coincident with administrative domains such as IP autonomous systems). In this example, packets from two different streams arrive at LSR1 with labels, L1 and L2, respectively, and are identified all to be destined for LSR11. However, LSR1 may be configured to offer separate paths to each stream through MPLS domain B (the solid arrows and the dotted arrows). It does this by using different labels for each stream. It adds another label, L3, in front of L1 for one LSP and a separate label, L4, for the other LSP, i.e. a new label is said to be **pushed** on to the stack. In domain B, this top-level label is used to forward packets through domain B. When the packets from each stream arrive at LSR11, it **pops** the top-level labels in each stream and makes its forwarding decision based on the original labels (L1 and L2) for each stream (respectively). This provides fast forwarding at the inter-domain level (using L1 and L2) and at the intra-domain level (using L3 and L4).

Label stacking could be a useful mechanism for enabling QoS-based or policy-based forwarding, providing VPN functionality and traffic engineering.

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 …

DigiComm II

Currently, there are three broad strategies in label assignment:

• **control traffic driven (topology-based):** here control traffic such as routing information is used to identify FECs. Generally, labels are pre-assigned based on network routes. ARIS (Aggregated Routing and IP Switching - IBM), Tag-switching (Cisco) and IP-Navigator (Ascend) use this strategy.

• **request driven:** explicit requests from control protocols (application-level or via management tools) that pass through a LSR may result in that LSR assigning a forwarding entry for a particular stream. Such explicit requests might be from a resource reservation mechanism like RSVP.

• **data traffic driven:** as streams/flows are "detected" by and LSR by examination of the data traffic, the LSR initiates the label assignment process and an LSP is established. IP switching (Ipsilon) and Cell Switch Router (Toshiba) are examples of the use of this strategy.

Each strategy has its pros and cons, which are highlighted in the MPLS documents.

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?

DigiComm II

MPLS is not a panacea for fast, QoS-capable IP networks. Indeed the MPLS WG recognises that in some cases MPLS will not be suitable. MPLS is seen as a complement to "traditional" IP-routing.

However, "traditional" IP-routing is being re-though and re-engineered. Not only are developers and manufacturers giving ploughing more resources into hardware solutions for IP routers and equipment, but the Internet community has ongoing research into faster table-lookup and packet classification algorithms.

Additionally, mechanism like CIDR (Classless Inter-Domain Routing) help address aggregation allowing better scaling of IP routing/forwarding information. A range of novel algorithms in the literate in the last 5 years show that it is possible to do classication of very large numbers of types of flow in near or equal to O(C) time.

Also, IPv6, which is fast approaching stability and ratification within the IETF, has a much simpler, less cluttered header, has the potential for direct flow identification (unlike IPv4) as well as being more amenable than IPv4 to hardware processing.Having said this, some of the QoS scheduling schemes are now affordable in router hardware. Coming from the other direction, novel switch-router platforms are making "full metal jacket" WFQ affordable too!

Issues regarding traffic engineering are at the fore. Internet service providers see the market in offering differentiated services, policy-based traffic handling and establishing service-level-agreements (SLAs) that are more than the simple "best-effort" offered through much of today's Internet.

The current thinking is that they key to providing flexible, QoS-capable, customer tailored services in the future is the integration of hardware with smart software, and not just making the hardware platform faster to run the same old 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.