



ATM, MPLS intserv, diffserv

TELE9751 week 11



Cisco Nexus context

- “Cisco introduced Nexus ® 5000 Series Switches both as high-bandwidth, low-latency, **access-layer switches** for rack deployment and as the basis for a unified network fabric that can help simplify data center infrastructure”
- “The classification can be based on CoS or **Differentiated Services Code Point (DSCP)** bits of the incoming packet”



Label Switching

Label switching: Labels of fixed length select state in switches that indicates how to switch packet (port, QOS, new label). <ZV]

Benefits:

- Simplify core network:
 - switches in network core merely lookup & swap fixed-length labels
 - complicated multi-field longest prefix matching to determine labels is done at edges
 - need only be done once
 - is done where there are relatively few flows
- Traffic management
- Provide new services, e.g. Virtual Private Networks

Used in both ATM and MPLS



ATM

Automated Teller Machine?

Air Traffic Management?

Another Terrible Mistake?

[J. Lane: “Is ATM a miracle or another terrible mistake?”, Telephony, p. 28, Oct. 24, 1994]

Another Technological Mirage?

[Vadim Antonov: ATM: Another Technological Mirage, www.pluris.com/ip_vs_atm]

In a similar vein to ISDN = It Still Doesn't Network

Asynchronous Transfer Mode – the technology underlying
Broadband Integrated Services Digital Networks. Still
(2017) in use for DSLAM interconnection/signalling.



ATM Resources

Keshav: Chapter 4 & § 15.4

Varghese: No explicit coverage. Coverage of specific issues:

4.2: Scheduling ATM flow control

4.15: ATM many-to-many VCs

Packets:

<http://uluru.ee.unsw.edu.au/~tim/zoo/index.html#ATM>



ATM Outline

- Short fixed-size packets (“cells”)
- Where ATM aimed to fit in
- ATM cell structure
- ATM Adaptation Layers
- Switching virtual paths and channels

Asynchronous Transfer Mode

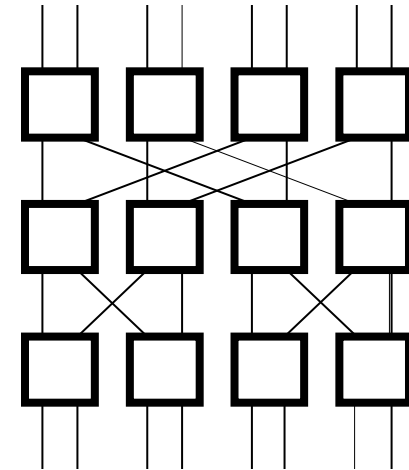
Short fixed-size packets (“cells”) carry information through the net
 Sometimes called “fast packet switching”.

Reasons for fixed size:

- ✓ switch design: speed:
 - ✓ simplicity: scheduling & buffer management
 - ✓ concurrency: space-division switches can have multiple ports operating synchronously

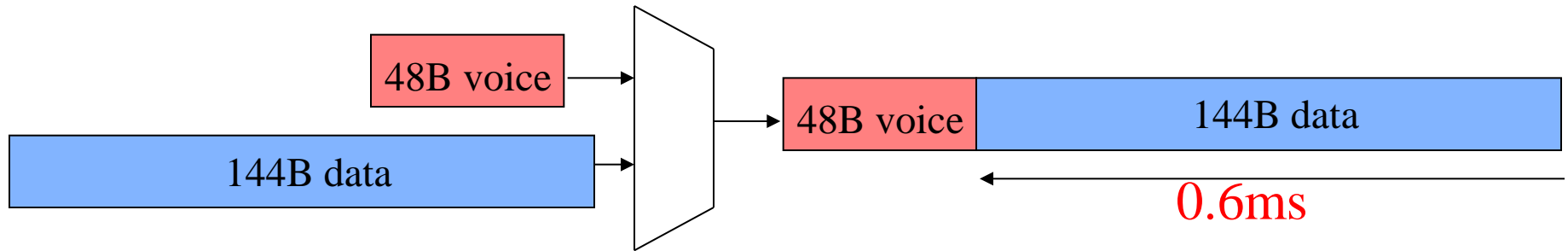
Reasons for size being short:

- ✓ minimise serialisation delays for sensitive traffic →

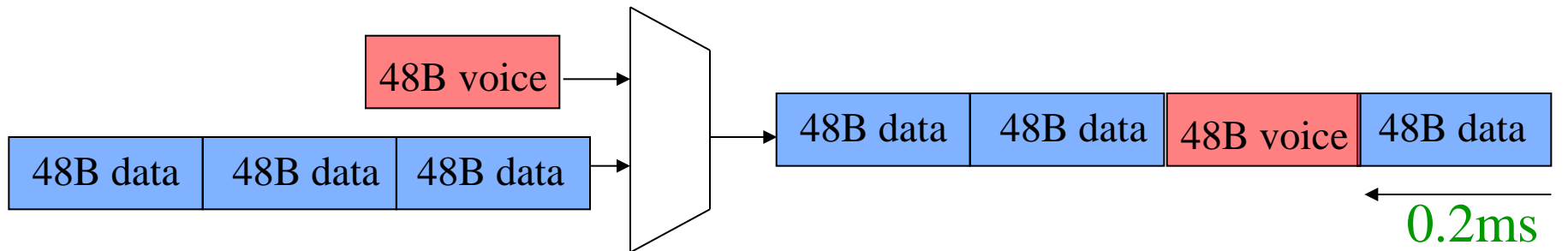




Minimising serialisation delays

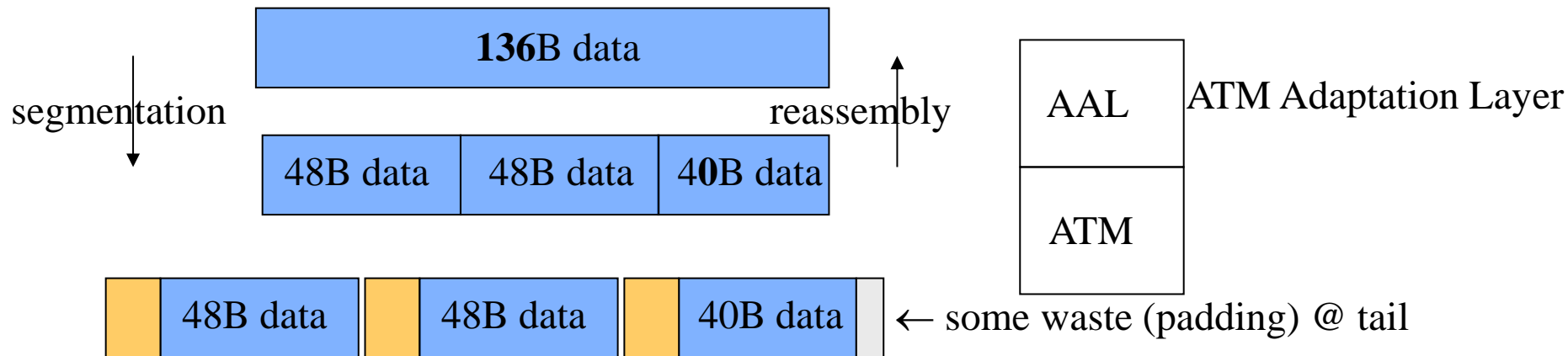


Multiplexing onto a 2.048Mb/s output port



Coping with a fixed size: Too small?

Applications with larger data units (e.g. 1KB TCP segments):
segment at sender, and *reassemble* at receiver

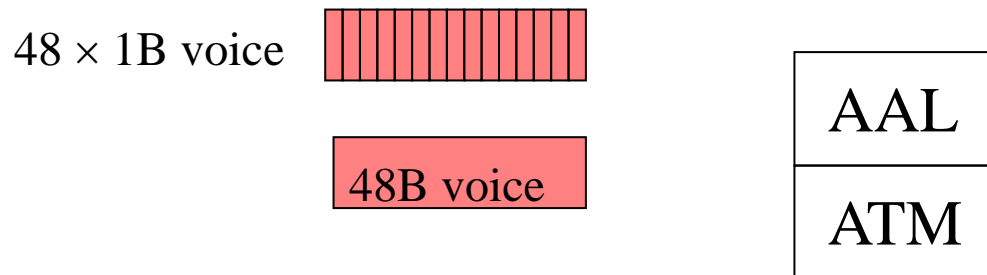


Too small => too much overhead per byte of payload

Coping with a fixed size: Too big?

Applications with smaller data units (e.g. 1B samples of streaming voice):

- Don't fill cell => waste
- **Packetise** payload: gather enough to fill a cell => delay



Deciding what fixed size

Powers-of-2 simplify implementation, e.g. memory mgt

ITU (CCITT at the time, part of UN) works on consensus:

US:

- Large country, long lines, install echo cancellers.
- Some audio delay OK to increase data efficiency
- Propose 64B payload

Europe:

- Smaller countries, no echo cancellers
- More emphasis on audio
- Propose 32B payload

Deadlock solved bureaucratically: split the difference: 48B

Similar solution for header: 5B

5B header = 9% bit overhead, considered OK with fibre

=> ATM cells are 53B: 5B header + 48B payload

pain to implement, most switches use 64B=53B+overhead

Why it is the *Asynchronous* Transfer Mode

Predecessor, Synchronous Transfer Mode, used circuit switching of time slots (e.g. by TSI switches)

Slots arrive “synchronously” (strictly, isochronously)

For efficient transfer of bursty media (in particular, data), want statistical multiplexing benefits of packet switching †:

“Asynchronous” because cells need not arrive periodically

=> ATM cells have identifiers VPI/VCI that allow receivers to associate them, despite unpredictable arrival times.

† Cell overhead has a cost (extra bits) but provides a benefit of releasing resources that would have been left idle. e.g. if 50% of traffic is bursty with mean utilisation of 40% then 9% cell overhead saves $50\% * 60\% = 30\%$ of link transmission

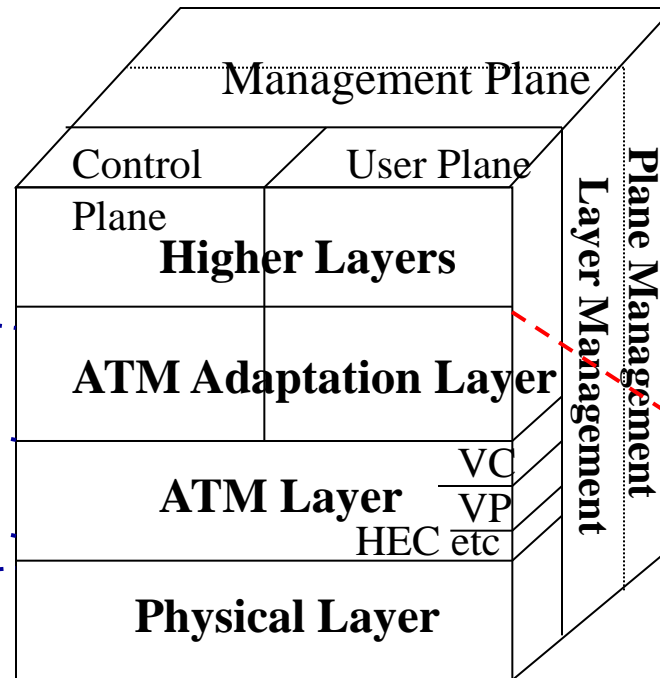


Outline

ATM protocol stack

Intended Reference
Model match

Application
Transport
Network
Data link
Physical



Effective RM match

Application	apps
Transport	TCP
Network	IP
Data link	AAL5 ATM
Physical	Optical

- Separation of control & user planes expedites payload transfer for user (complicated signalling is separate)
- Management has an explicit plane

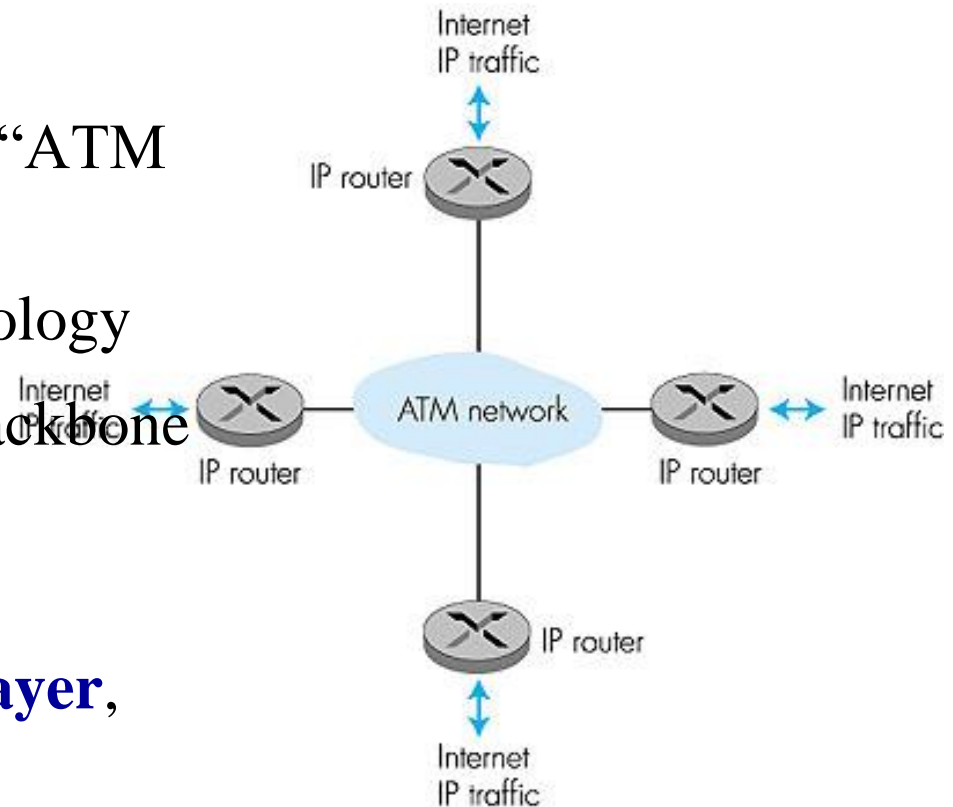
ATM: network or link layer?

Vision: end-to-end transport: “ATM from desktop to desktop”

- ATM *is* a network technology

Reality: used to connect IP backbone routers

- “IP over ATM”
- ATM as **switched link layer**, connecting IP routers



Why ATM failed

ATM to the desktop didn't happen:

- ✗ Applications didn't change to become QOS aware in time
- ✗ Apps still used IP (over ATM) to communicate to legacy apps; IP masked QOS improvements
- ✗ Network Interface Cards never broke vicious circle of costing too much to sell in volume to make them cheap

=> ATM used as a high-speed alternative to Ethernet; until Ethernet caught up.

Network service providers initially adopted ATM – e.g. see some traceroute traces.

Reasons:

- ✓ ATM's ability to segregate traffic
- ✓ high speed
- ✓ hedge bets

ATM's legacy:
MPLS ...

e.g. May 2004:

```
$ traceroute www.uwa.edu.au
traceroute to mysource (129.94.255.181): 6 hops:
 1  eebu4s1.uwn.unsw.edu.au [129.94.255.181]
 2  129.94.255.181
 3  gig2-2.nswrnosbb.nsw.net.au [192.111.1.1]
 4  203.15.123.178
 5  nsw-wa.atm.net.au [203.15.123.178]
 6  parnet2-uwa.parnet.net.au [203.15.123.178]
```

e.g. May 2005:

```
$ traceroute www.uwa.edu.au
traceroute to mysource (129.94.255.181): 6 hops:
 1  eebu4s1.uwn.unsw.edu.au [129.94.255.181]
 2  129.94.255.181
 3  192.111.1.1
 4  POS1-3.c
 5  POS6-1.e
 6  * * *
```

e.g. May 2009:

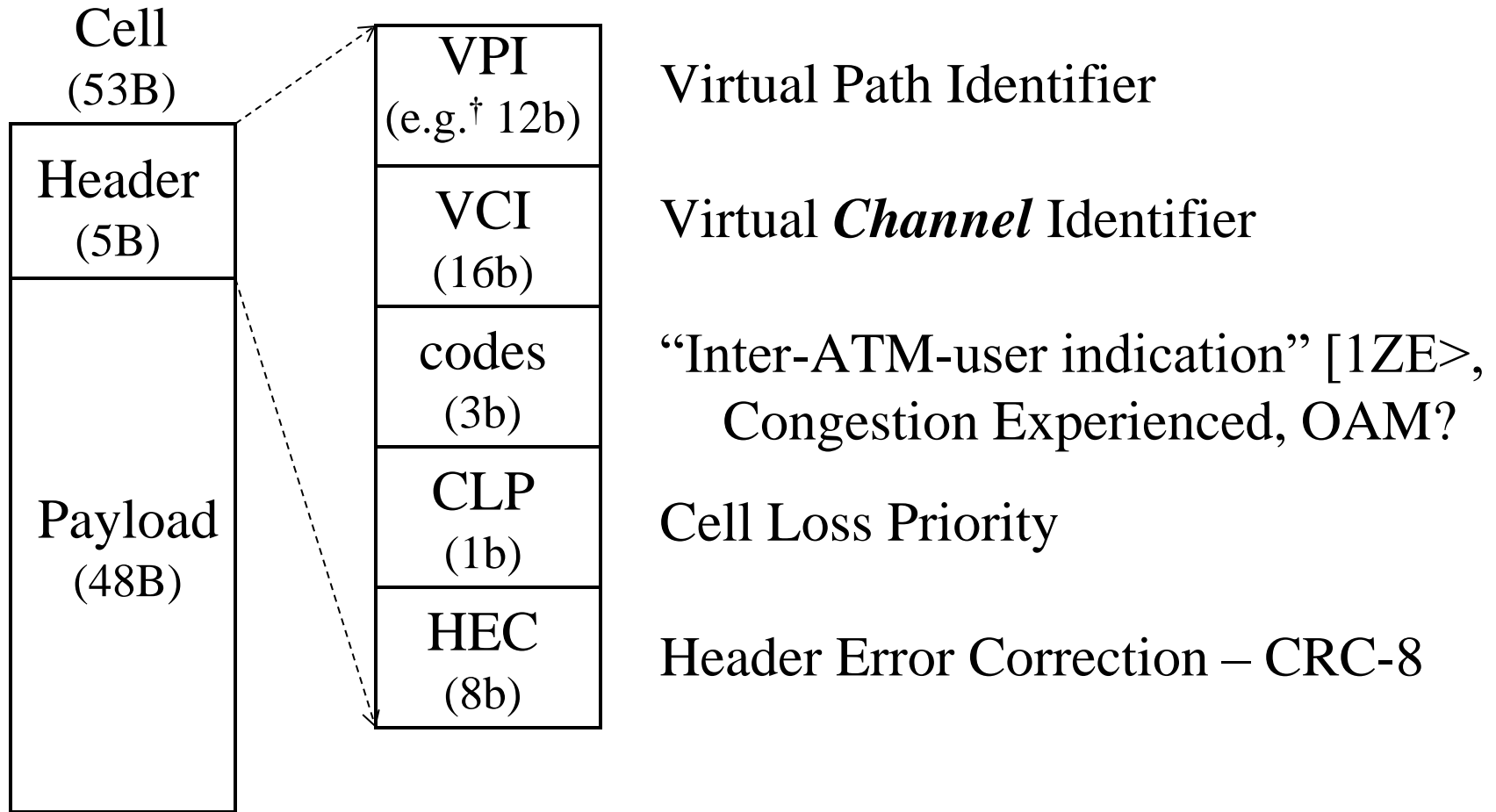
```
$ traceroute www.uwa.edu.au
traceroute to mysource (129.94.255.181): 6 hops:
 1  eebu4s1.uwn.unsw.edu.au [129.94.255.181]
 2  te-1-3.ombo
 3  te-7-2.unsw
 4  gigabiteth
 5  ge-4-1-0.bk
 6  so-0-1-0.bk
```




Outline



ATM cell structure: Key parts





Outline



ATM Adaptation Layers

Adapt the cell transfer service of ATM to application requirements.

Originally there were 4. AAL x , where $x=$

e2e timing relationship “real-time”

- 1 **Connection-oriented (CO) Constant Bit Rate (CBR)**, e.g. circuit emulation
1B header (Sequence # + CRC), 47B payload[†]
- 2 **CO Variable Bit Rate (VBR)**, e.g. VBR video, standardised late

no e2e timing relationship

- 3 **CO VBR** merged with AAL4
- 3/4 **Connectionless (CL) VBR**: each application packet is encapsulated with a 4B header and 4B trailer, then segmented into cells, each having 2B header and 2B trailer →

[†] AAL1/2 lives on in current video formats (MPEG) which encode video in 1VP* 188B chunks which were designed to fit neatly into 4x47B AAL1/2 payloads



Pay no attention to the details
The point is that it is complicated

AAL 3/4

Packet encapsulation:

CPI	Btag	BAsize	payload	pad	Etag	len
8b	8b	16b	≤64KB	8b-32b	8b	16b

Common Part Indicator = version number

Beginning(B)/End(E) tag: cells at B&E of packet should have matching B/Etags. Helps detect burst loss of last cell and 1st cell of next packet.

Buffer Allocation (BA) size: Suggests how much space the receiver should allocate to reassemble this packet. Streaming sources may only be able to estimate length => record actual length at end.

pad: so that Etag and length always start on 32b word boundaries

+ Cell encapsulation:

type	Seq	MID	payload	len	CRC
2b	4b	10b	44B	6b	10b

Type: Beginning/End/Continuation/Single-Segment Message

Seq: Cell sequence number

MID: Multiplexing identifier (multiple packets can be interleaved on one link)

- ✗ Many piecemeal mechanisms (Length, B/Etags, cell CRC) to detect errors.
- ✗ High overhead: Common TCP ACKs (20B TCP+20B IP header) + 8B packet encapsulation = 48B > 44B => 2 cells => 40B in 106B = 38% efficiency :-)

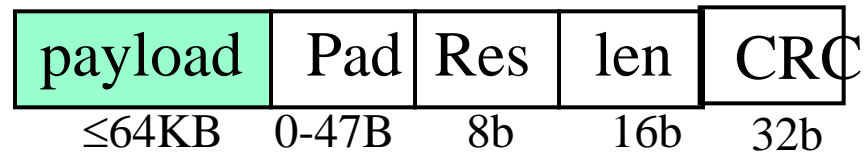


AAL 5

A computer industry response to the inefficiency and complexity of AAL 3/4.

No cell encapsulation.

“Inter-ATM-user indication” in cell header => is this is the last cell?



Packet-level CRC can detect with high probability all errors (loss, mis-sequencing, bit errors, duplication)

Became the dominant AAL



Outline

Virtual Channels and Paths

VCs identify *individual* flows of information between endpoints.

Multitudes of flows through network core switches

Inefficient for core switches to handle individual flows

e.g. 622Mb/s link carries 8,800 64kb/s voice channels

average call duration of 2 minutes => 150 connection establish/release requests per second = large signalling burden

=> **Virtual Path** = aggregate of VCs

Commerce analogy:

- core switches are like wholesalers, dealing in bulk (VPs)
- edge switches are like retailers, dealing with individuals (VCs)

Resources may be assigned to the VP

✓ good for predictable flows (set up VC without negotiating switch resources)
– e.g. in core of network where individual variations annul one another

✗ may waste resources when flows vary

The SDN equivalent of this is to mask the VC bits as “don’t care”

Switches translate identifiers

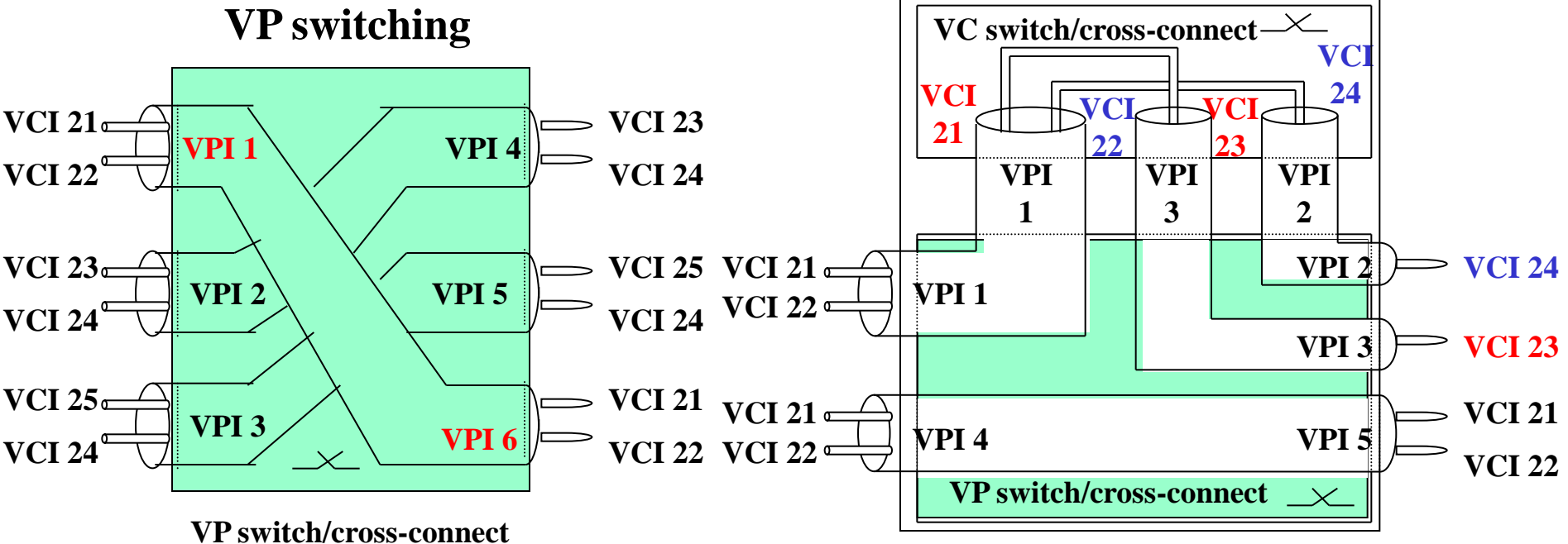


Figure based on ITU Recommendation I.311

This is (visually) similar to the layers of optical switching covered earlier <6R]



MultiProtocol Label Switching (MPLS)

MPLS Resources

Varghese: S 11.2. **Keshav:** not covered

Packets: <http://uluru.ee.unsw.edu.au/~tim/zoo/index.html#MPLS>

Standards:

- Primary: RFC3031
- Others: <http://www.ietf.org/html.charters/mpls-charter.html>

Advocate: <http://www.ipmplsforum.org/>

Tutorials:

- Papers:
 - R. Winter: “The Coming of Age of MPLS”, *IEEE Communications Magazine*, 49(4):78-81
 - A. Viswanathan et al.: “Evolution of Multiprotocol Label Switching”, *IEEE Communications Magazine*, pp. 165-73
 - W. Stallings: “MPLS”, *The Internet Protocol Journal* 4(3): 2-14
 - F. Palmieri: "GMPLS Control Plane Services in the Next-Generation Optical Internet", *The Internet Protocol Journal*, 11(3):2-18
- Courses: <http://www.nanog.org/mtg-9905/ppt/mpls/>

Book: B. Davie et al.: *Switching in IP networks: IP switching, tag switching, and related technologies*



MPLS outline

- Label Switching
 - Multiprotocol Label Switching
 - Comparison with ATM
- MPLS terminology
- MPLS label stack entries
 - Stacks of labels
 - Pushing & popping *multiple* labels
- How are labels carried?
- MultiProtocol Lambda Switching (MP λ S)
/ Generalized MPLS (GMPLS)
- Label Distribution
 - Setting up Label Switched Paths
- MPLS in use



Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS[†]):

- Initiated by desire to send IP over ATM, e.g. to interconnect routers.
(Original technique to interconnect routers was to emulate broadcast <D5] LAN on ATM, but that required large numbers of point-to-point VCs, unproven ATM routing protocols, & obscured QOS benefits of ATM)
- Advocated by Cisco, IBM, Ipsilon and others – some called “tag switching”
- Generalized in IETF to multiple protocols:
IP/IPX/NetBEUI/etc over ATM/Frame Relay/Ethernet/etc

[†] Like ATM, MPLS also has its detractors, e.g. “Much Preaching, Little Substance”

email to end-to-end mailing list, M



Comparison with ATM

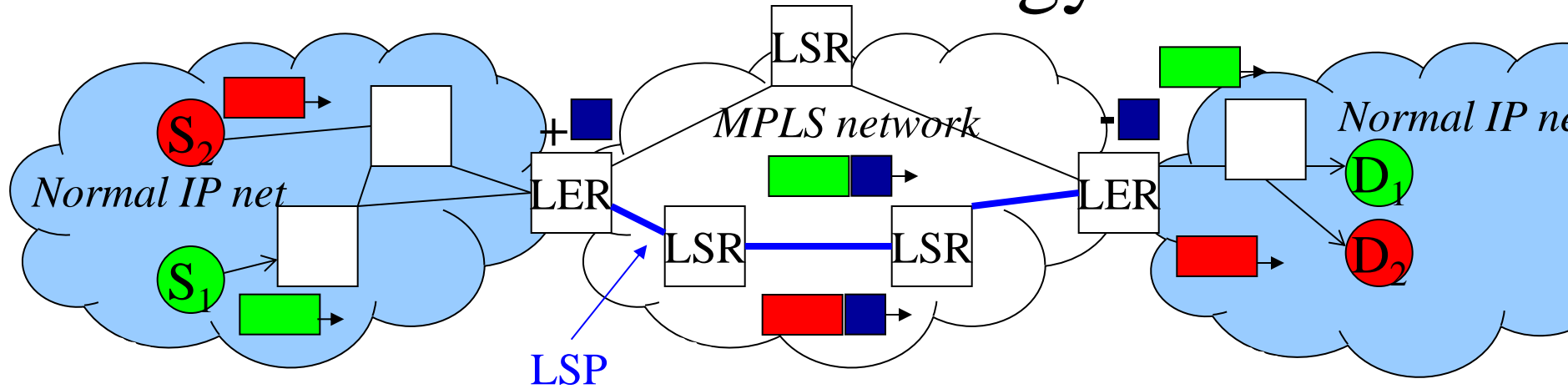
Similarities:

- Label switching
- Hierarchy of labels:
 - ATM has 2 levels: VPs aggregate VCs
 - Variable packet lengths allow more than 2 MPLS labels to be stacked

Differences (IP flavour):

- Packets have variable length
- “Routing” can be done using established IP protocols (e.g. BGP), rather than newer ATM protocols (e.g. across P-NNI).
Often described as “layer 3 routing + layer 2 switching”, where:
 - “Routing”: General determination of which port to send packets so that they reach their destination, satisfy QOS requirements, and minimise network load.
 - “Switching”: Using a label to index a table to determine which port to send a particular packet at a specific instant. Tables may vary over time as routing protocols discover better routes.

MPLS terminology



Label Edge Router (LER) – pushes (+) / pops (-) labels

Label Switched Router (LSR) – switches using MPLS labels

Label Distribution Protocol – enables a downstream router (closer to destination) to allocate a label to an upstream router.

Label Switched Path (LSP) – similar to ATM VC/VP

Forwarding Equivalence Class (FEC): A set of packets that are all treated the same way by a router

Not necessarily by *all* routers – e.g. packets with different DAs may belong to same FEC in the midst of the network

Conventional IP routers determine FEC at each hop, MPLS routers (LERs) determine FEC only at entry to domain.

Stacks of labels

Packet can have multiple labels (in a “stack”), each indicating how it should be switched in different domains.

Labels can be pushed onto the stack at:

- **source** (source routing), or
- **ingress LER** to domain: Helps scalability/hierarchical routing: Domain routes aggregate (common outer label of stack) rather than individual flows.
 - At exit to domain, the common label is popped, and future routing is done according to next label in the stack.
 - Pushing a label may extend the packet length beyond the Maximum Transmission Unit (MTU) => LERs need to determine the MTU, and be prepared to fragment packets if label pushing extends length beyond MTU (c.f. bridge which discards frames that are too long <PA])

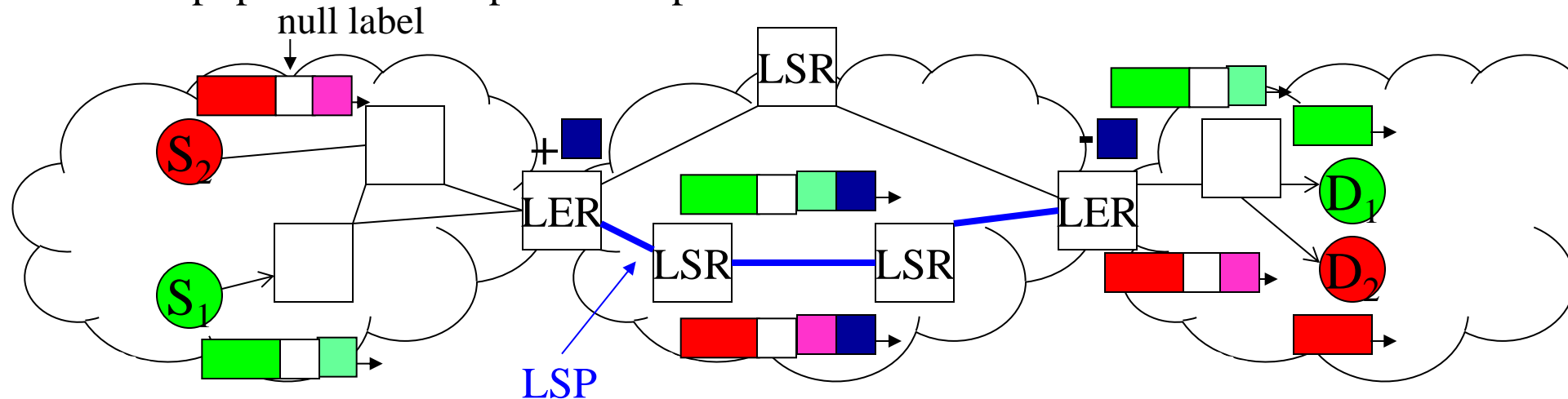
Labels are popped off the stack by egress routers

Last label indicates protocol: IP (and in theory other protocols, e.g. IPX)

Pushing & popping *multiple* labels

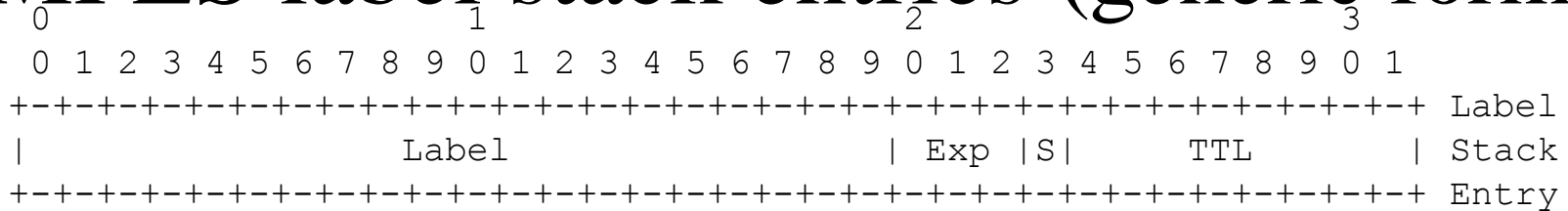
Transit network (middle):

- pushes on label (**blue**) as packets enter
- this label is used for switching within this network
- pops off label as packets depart





MPLS label stack entries (generic format)



20b **Label.** Values $[16, 2^{20}-1]$ identify FECs. Other special values:

0: **IPv4 Explicit Null Label:** End of LSP, next protocol is IPv4

1: **Router Alert Label:** bring pkt to router's attention, next label=FEC

2: **IPv6 Explicit Null Label** Not particularly multiprotocol yet!

3: **Implicit Null Label:** For LDP only, not for packets

4-15: **Reserved**

3b **Experimental:** Can be used for ECN [RFC5129]

1b **S: Bottom of stack:** =1 for last label in stack

8b **Time-To-Live,** adjusted by switches & returned to IP @ egress

This is the generic format of the "shim layer", added when the link layer (e.g. Ethernet) cannot convey labels

MultiProtocol Lambda Switching (MP λ S) / Generalized MPLS (GMPLS)

Traffic engineering tasks for optical networks:

- Coordinating use of wavelengths, e.g. ensuring wavelength continuity
- Restoration services: e.g. identify backup lightpaths that don't share same risks (e.g. fibre or duct)

Need a signalling protocol to allow dynamic reconfiguration of a network of devices from varied manufacturers

GMPLS generalises MPLS idea of a label to include anything that is sufficient to identify a flow. e.g. wavelength or port

Some differences from MPLS, e.g. no optical equivalents of label merging or label push/pop operations. label swapping = wavelength conversion



Label Distribution

Labels are allocated by downstream router & distributed to upstream router using:

- a Label Distribution Protocol (LDP – RFC 3036), a Constraint-Based LDP (RFCs 3212-3214) or
- extending routing protocols (e.g. BGP – RFC 3107), or
- signalling (e.g. **RSVP**) protocols to carry labels.



Setting up Label Switched Paths

What causes a Label Switched Path to be set up? Either:

- **Data driven:** On demand by source transmitting packets
- **Control driven:** When routing topology changes, or explicit signalling request to establish a LSP

How is the LSP determined?:

- Layer 3 hop-by-hop routing, of either:
 - Initial packets that are transmitted & stimulate data-driven setup
 - Control signals
- an Explicit Route (source route) specified by the source



MPLS in use

[May 2015]

```
$ traceroute www.uwa.edu.au
```

```
traceroute to www.uwa.edu.au (130.95.128.140), 30 hops max, 38 bytes
```

```

1  eebu4s1.uwn.unsw.EDU.AU.92.171.149.in-addr.arpa (149.171.92.2)
2  libcr1-po-5.gw.unsw.edu.au (149.171.255.165)  0.328 ms  0.328
3  unswbr1-te-8-1.gw.unsw.edu.au (149.171.255.105)  0.434 ms unsw
4  bfw1-ae-1-3053.gw.unsw.edu.au (129.94.254.76)  0.600 ms  0.814
5  unswbr1-v1-3054.gw.unsw.edu.au (129.94.254.82)  0.840 ms  0.814
6  138.44.5.0 (138.44.5.0)  1.113 ms  1.152 ms  1.130 ms
7  et-1-1-0.pe1.rsby.nsw.aarnet.net.au (113.197.15.12)  2.320 ms
   MPLS Label=321104 CoS=3 TTL=1 S=0
8  et-0-3-0.pe1.prka.sa.aarnet.net.au (113.197.15.42)  19.024 ms
   MPLS Label=300832 CoS=3 TTL=1 S=0
9  et-0-3-0.pe1.knsg.wa.aarnet.net.au (113.197.15.45)  45.889 ms
10 138.44.176.3 (138.44.176.3)  45.454 ms  45.390 ms  45.396 ms
11 * * *
12 * * *
```

tracert on Windows machines tends to not show MPLS information



Quality of Service in the Internet: Intserv and Diffserv



Broad outline

- Integrated Services (intserv): RSVP
- Differentiated Services (diffserv)
- Tying it all together

Overview papers:

X. Xiao and L. Ni: “Internet QoS: A big picture”, *IEEE Network Magazine*, 31(2): 8-18.

covers intserv, diffserv, MPLS

C. Metz: “IP QOS: Traveling first class on the Internet”, *IEEE Internet Computing*, 3(2):84-8

covers intserv, diffserv



RSVP outline

General architecture

- RSVP is separate from routing protocols

- Soft state: Definition, Benefits, Costs

- How RSVP supports multicast

Messages

- Formats

- PATH & RESV

Using RSVP

- scalability

- RSVP over reservationless networks

RSVP Resources

Varghese: S 14.5

Keshav: pp. 471-5

Packets:

<http://uluru.ee.unsw.edu.au/~tim/zoo/index.html#RSVP>

Standards: RFCs 2205-2216

Advocate: <http://www.isi.edu/div7/rsvp/>

Tutorials:

L. Zhang, et al.: “RSVP: A New Resource ReSerVation Protocol”, *IEEE Network Mag.*, 7(5):8-18

Chapter 6 of Jha and Hassan



Resource reSerVation Protocol (RSVP)

- Offers 2 types of service:
 - **Guaranteed service** [RFC 2212]: Provides firm end-to-end delay bounds
 - **Controlled-load service** [RFC 2211]: No firm delay bounds, but service shouldn't deteriorate as network load increases.

Both involve admission control.

- Separate from routing protocols →
- Uses “soft state” →
- Designed to support group communication →



RSVP is separate from routing protocols

Designed to be separate from the routing protocol

- ✓ Simplifies RSVP
- ✓ Allows operation with existing routing protocols
- ✗ May hinder ability to provide QOS guarantees (Route that provides necessary QOS depends on the ~~connection's~~ session's profile & requirements of traffic => depends on signalling)



Soft and hard state

RSVP uses soft state (PATH and RESV messages are periodically resent)

Compare with ATM, which uses “hard state” ...

Under *normal* conditions:

(“normal”: systems using state info remain available, e.g. haven’t crashed)

Hard state persists until it is explicitly released.

Soft state persists only for a moderately short interval.

It must be refreshed if it is to persist longer.

Failure to refresh implies request to release.

i.e. soft state is *often* refreshed, whereas hard state is not.



Soft and hard state examples

Student accounts on a computer [Keshav, p. 109]:

Hard state: Account lasts indefinitely, until student asks for it to be deleted.

Account may never be deleted because the student:

- is unable to ask (e.g. dies or wins the lottery)
- forgets to ask
- doesn't want to ask: they quite like access to the system

Soft state: Account lasts one session only. Long-term students have to renew their accounts each session.

Intermediate between virtual circuits and datagrams <NU>].



Pros and cons of soft state

- ✓ **Adaptive:** routing path changes redirect endpoint refresh messages, creating new state.
- ✓ **Robust:** Packets can be lost and switches can crash, but service will continue until either there is no path (service is impossible) or end-system loses state

Soft state *simplifies signalling*, since switches needn't explicitly adapt to changes, or recover from failures.

- ✗ **Extra transmissions to refresh the state**

RSVP supports *group* communication

Designed at the outset to support group communication:

1. **Receiver oriented** (c.f. sender-oriented ATM)

- Receivers initiate reservations (source initiates a multicast group, but receivers initiate their connection to the group)
 - Merger of reservations enables many receivers without many requests imploding at source
 - Reservations can be heterogeneous, according to differing requirements
 - It is the *receiver* that experiences QOS & usually pays for service

2. State information is continually refreshed →

3. Reservations are for “sessions” →

RSVP supports *group* communication

(continued)

2. **Refreshing of soft state reflects updates** to group membership.
Multicast membership changes more often than unicast => continuous refreshing is appropriate.
3. **Reservations are for “sessions”**: traffic *to* a particular multicast group (not *from* a particular source)
 - A session may have multiple sources, e.g. videoconference
 - Reserving resources for sessions, rather than sources, (recall “multiaccess” service <D5]) allows admission of more sessions: often only one or a few sources transmit at any instant.



Outline



Sample RSVP message

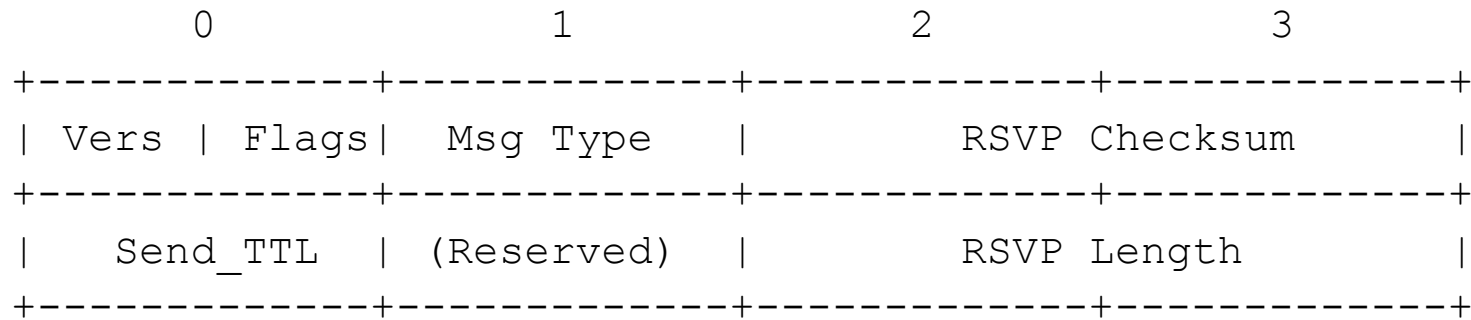
- ⊕ Ethernet II, Src: Cisco_9d:94:01 (00:90:92:9d:94:01), Dst: Cisco_c3:b8:47
- ⊕ Internet Protocol, Src: 17.3.3.3 (17.3.3.3), Dst: 16.2.2.2 (16.2.2.2)
- ⊖ Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPV4-LSP, Des
 - ⊖ RSVP Header. PATH Message.
 - RSVP Version: 1
 - Flags: 00
 - Message Type: PATH Message. (1)
 - Message Checksum: 0xdb58 [correct]
 - Sending TTL: 254
 - Message length: 264
 - ⊕ SESSION: IPV4-LSP, Destination 16.2.2.2, Tunnel ID 1, Ext ID 11030303.
 - ⊕ HOP: IPV4, 210.0.0.1
 - ⊕ TIME VALUES: 30000 ms
 - ⊕ EXPLICIT ROUTE: IPV4 210.0.0.2, IPV4 204.0.0.1, IPV4 207.0.0.1, ...
 - ⊕ LABEL REQUEST: Basic: L3PID: IP (0x0800)
 - ⊕ SESSION ATTRIBUTE: SetupPrio 0, HoldPrio 0, SE style, [sys17-3_t1]
 - ⊕ SENDER TEMPLATE: IPV4-LSP, Tunnel Source: 17.3.3.3, LSP ID: 1.
 - ⊕ SENDER TSPEC: IntServ, Token Bucket, 625000 bytes/sec.
 - ⊕ ADSPEC

<http://uluru.ee.unsw.edu.au/~tim/zoo/index.html#RSVP>



RSVP message format: Header

Common header (for all message types):



Flags: only 1b ever used to control refresh overhead – see RFC 2961

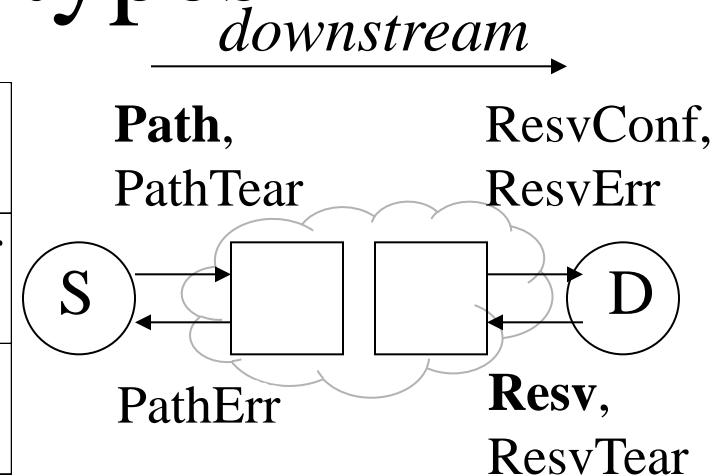
Message types: See next slide [171>

TTL: Compared with IP’s TTL to detect non-RSVP hops on path.

Objects follow the header, encoded as Type-Length-Value triplets

RSVP message types

Transmission direction	Commands	Responses	Errors
Downstream	Path , PathTear	ResvConf	ResvErr
Upstream	Resv , ResvTear		PathErr



Path: Gathers information about path from source

Resv: Reserves resources

***Tear:** Release resources before soft-state timeout

***Err:** Error interpreting message, or honouring reservation



PATH message

4 roles:

- records network capacity on path from source (**AD_SPEC**)
- informs receivers of source's capabilities (**SENDER_TSPEC**)
- refreshes state in routers (e.g. sent every 30 seconds)
- marks the path from the source (**RSVP_HOP**), for return of RESV ...

```

<Path Message> ::= <Common Header> [ <INTEGRITY> ]
  <SESSION>                IDs session, e.g. multicast IP address
  <RSVP_HOP>              IDs previous node for RESV return
  <TIME_VALUES>          specifies refresh rate†
  [ <POLICY_DATA> ... ]  e.g. access credentials
  [ <sender descriptor> ] describes path state

```

```

<sender descriptor> ::=
  <SENDER_TEMPLATE>      IP address & port of source
  <SENDER_TSPEC>         Leaky Bucket spec. of source traffic
  [ <ADSPEC> ]          summary of upstream capabilities

```

(Receivers use IGMP to join group, informing routers of route for PATH msg)

[†] Larger TIME_VALUES => lower overhead, but less responsive to network changes.
 Receivers may send RESVs with period $\approx \# rx$ for scalability => router needs to know period.



RESV message

Role: makes reservation & acks PATH message

- Default is for no explicit confirmation for RESV: Receiver knows of success through payload delivery
- RESV error messages (in case reservation isn't possible)

```

<Resv Message> ::= <Common Header> [ <INTEGRITY> ]
  <SESSION>
  <RSVP_HOP>
  <TIME_VALUES>
  [ <RESV_CONFIRM> ]           IP address of receiver requesting confirmation
  [ <SCOPE> ]                 Lists targets of this RESV
  [ <POLICY_DATA> ... ]
  <STYLE>                     Indicates if RESV is for all or some sources in session
  <flow descriptor list>

```

```

<flow descriptor list> ::= <empty> |
  <flow descriptor list>
  <flow descriptor>          Defines flow & source that it applies to

```



Outline



RSVP scalability

- Scales *well* in terms of *source overhead* for increasing numbers of receivers.
- Scales *poorly* in terms of *switch overhead* for increasing numbers of sessions through that switch:

Each switch on the end-to-end path must process signalling, not just to establish & release the state, but also continuous refresh messages.

RSVP is best at edges of network, where there are few flows



RSVP over reservationless networks

May need to connect networks supporting reservations with reservationless networks (LANs, or IP)

- RSVP messages still pass over these networks
- QOS can only be assured if the reservationless networks are not the bottleneck

Common arrangement: interconnect RSVP networks with a **diffserv** network =>

- edges (where scalability isn't critical) have per-flow state
- network core only has state for aggregates (e.g. all video packets)

Need to map between services:

<u>intserv</u>		<u>diffserv</u>
Guaranteed	↔	Expedited (premium)
Controlled load	↔	Assured

For details, see RFC 2998



Outline

- Diffserv



diffserv resources

Varghese: S 14.8.2, **Keshav:** not covered

Standards:

Primary: RFC2475

Others: RFC 2597 (Assured Forwarding),
RFC 3246 (Expedited Forwarding)

<http://www.ietf.org/html.charters/diffserv-charter.html>

Introductory papers:

F. Baumgartner et al.: “Differentiated Services: A new approach for Quality of Service in the Internet”, *Proc. High Performance Networks*

Cisco Systems: DiffServ - The Scalable End-to-End QoS Model

http://www.cisco.com/warp/public/cc/pd/iosw/ioft/iofwft/prodlit/difse_wp.htm



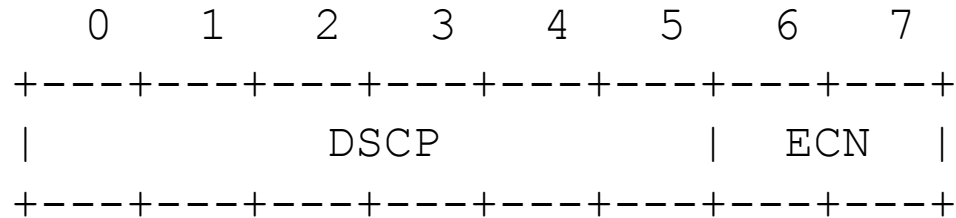
Diffserv terminology

Per Hop Behavior (PHB): Defines the behaviour of a switch when forwarding certain packets, e.g. what service they receive

Behavior Aggregate (BA): A group of packets passing through a common network point that have a common label (DSCP) so that they experience the same PHB at that point.

Different BAs experience different behavior => “*Differentiated Services*”

Differentiated Services Code Point (DSCP): Replaces IP ToS field



DSCP interpretation can be customized for specific domains (i.e. Autonomous Systems/Networks) => remarking at domain boundary

Per Hop Behaviors

Best-effort service (DSCP=000000); may be lost or delayed

Assured Forwarding (AF) (aka “assured service”) (DSCP= $x_1x_2x_3y_1y_20$)

- ‘Assured’: gets lower *loss* rate than normal ‘best effort’ (no mention of delay)
- Policed <FM]: Excess traffic is converted to best effort
- Theoretically assured for a *source* (not S-D pair), but that leads to inefficient provisioning, since source may transmit to arbitrary destination – allocate resources on all possible links?!

12 types (AF xy) ($x_1x_2x_3=001,010,011$ or 100 , $y_1y_2=01,10$ or 11^\dagger)

- 4 classes (x), each assigned separate resources (buffer&transmission capacity)
- 3 drop precedences (y) per class. Prob(drop) $\propto 1/y$

Expedited Forwarding (EF) (aka “premium service”) (DSCP=101110)

- low loss (like AF) *and* low delay, e.g. for voice/video
- Traffic entering network must be rate limited, e.g. through a shaper <FM]

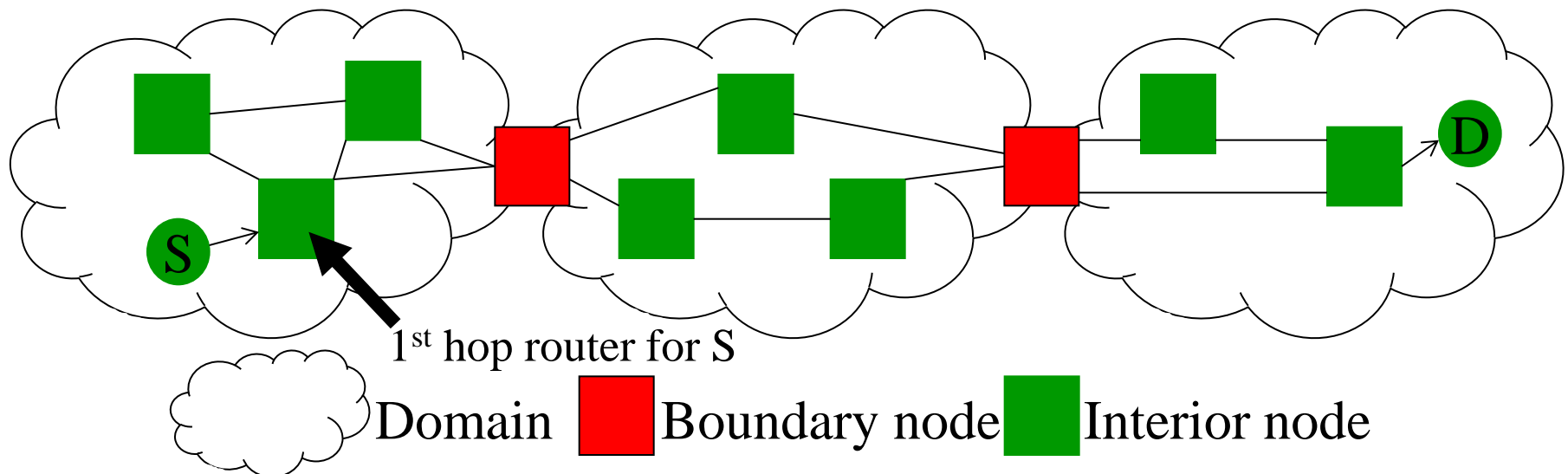
$^\dagger y_1y_2=00$ is reserved for backward compatibility with traffic that uses IP precedence

Diffserv entities

Domains: Contiguous network regions, with common administrative ownership, set of service provisioning policies and PHB definitions.

Boundary node: Point of entry (ingress) or exit (egress) from domain. They regulate traffic & set DSCP labels (like MPLS LER)

Interior nodes: Within a domain, but not a boundary node
 Forward transit traffic, with service determined by label (like MPLS LSRs)
 May mark locally generated traffic (may also be done by source)



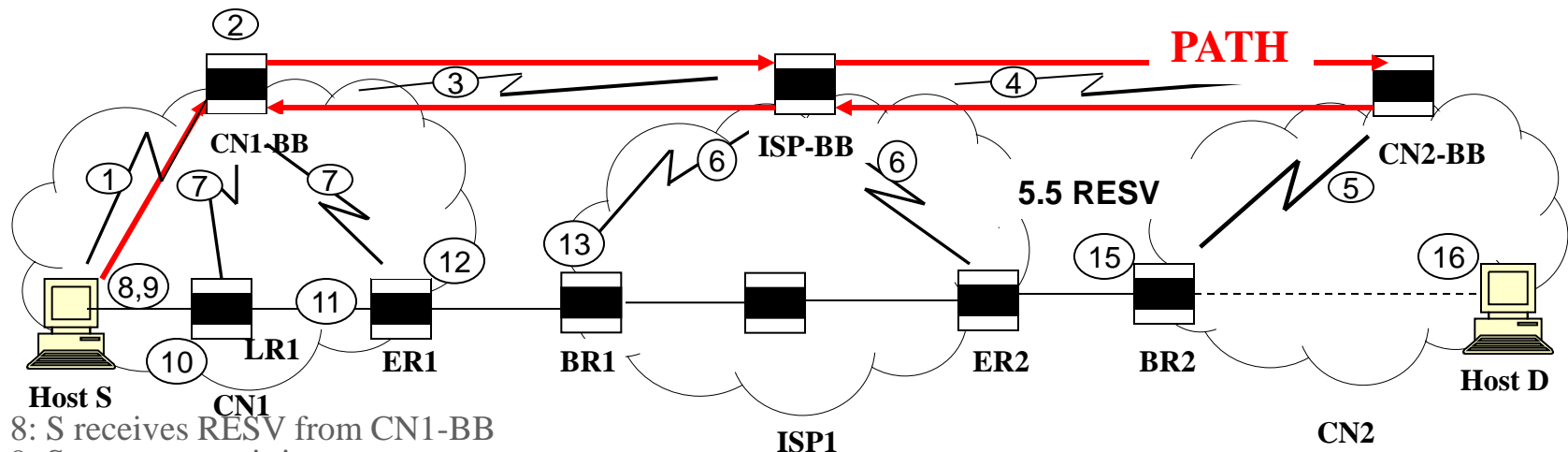
Bandwidth Brokers

Each Domain has a Bandwidth Broker (BB), responsible for admission control for that domain (c.f. ATM/MPLS/RSVP where admission control is distributed amongst switches in domains)

Signalling between BBs is done using a protocol such as **RSVP** or LDAP (Lightweight Directory Access Protocol – RFC 1777)

BBs can aggregate multiple requests before forwarding an aggregated request to the next BB.

Fig. from Xiao



8: S receives RESV from CN1-BB

9: S starts transmitting

10: LR1 classifies & forwards



Multicasting complicates diffserv

Diffserv is *source* oriented

But source may not know locations of multicast members & membership may change rapidly

=> difficult for network to anticipate load that a multicast flow will create

=> difficult to assure performance

For details, see Internet Draft draft-bless-diffserv-multicat-03.txt: R. Bless & K. Wehrle: IP Multicast in Differentiated Services Networks



Outline

- Putting it all together



	Intserv	diffserv
Unit of operation => granularity of isolation & guarantees	individual sessions	aggregates
Signalling frequency	frequently refreshed for each flow	rarely (SLAs)
Based on	reservation for <i>rx</i>	prioritization of <i>sources</i>
Able to guarantee absolute service to end-users?	yes	no
Service scope (end-to-end requires extensive network upgrade)	end-to-end	domains
Complexity	routers must process signals for individual flows => poor scalability	simple internal nodes, boundary nodes must classify & regulate traffic
Compatibility with multicast traffic	good	problematic



Tying it all together

	MPLS	RSVP	diffserv
Terminology	Label Switched Router		Interior Node
	Label Edge Router		Boundary Node
	Forwarding Equiv. Class	session	Behavior Aggregate
Role of RSVP	may distribute labels	reserves for rx	may reserve for source/domain

↑ RSVP can be used with *all* of the above (not intended to confuse you!)



Things to think about

- **Critical thinking:** What lessons can be learned from repeated unsuccessful attempts to provide network QoS (eg. ATM) and how might they be avoided in current/future protocols?
- **Engineering methods:** Each of these protocols have defined interfaces for accessing the service provided by the system and a procedure for signalling desired service across the interface.
- **Links to other areas:**
 - Several systems offer both connectionless and connection-oriented control, e.g. MPLS's "Data driven" and "control driven" setup <1K0], and "Tell and go" and "Tell and wait" setup for optical burst switching <RJ]
 - HTTP cookies are another example of soft state
- **Independent learning:** Read about how Carrier Ethernet offers QoS across Metro Ethernet Networks, and how a Stream Reservation Protocol (like RSVP) has been added to IEEE 802.1Q



The end