Common Protocols

An Engineering Approach to Computer Networking

The grand finale

- Previous chapters presented principles, but not protocol details
  - these change with time
  - real protocols draw many things together
- Overview of real protocols
  - standards documents are the final resort
- Three sets of protocols
  - telephone
  - Internet
  - ATM

Telephone network protocols

	Data Plane	*Control Plane (SS7)*
App	Voice/Fax	ASE/ISDN-UP TCAP
Session		
Transport		
Network		SCCP/MTP-3
Datalink	Sonet/PDH	MTP-2
Physical	Many	MTP-1

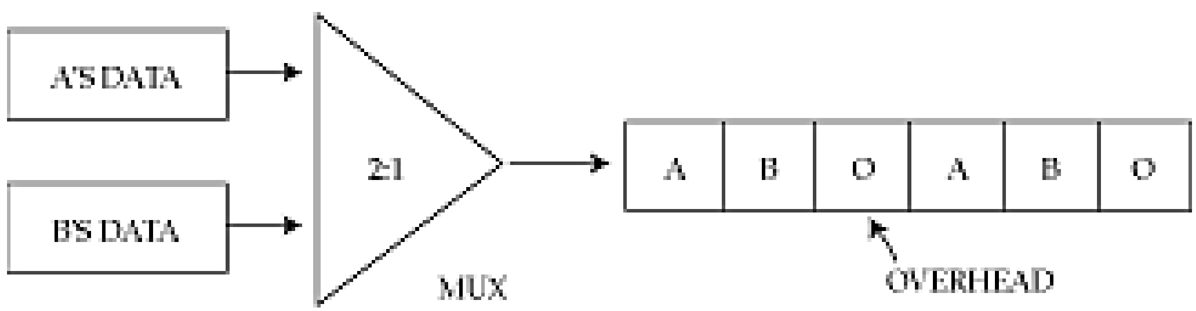
Traditional digital transmission

- Long distance trunks carry multiplexed calls
- Standard multiplexing levels
- Digital transmission hierarchy

	US and Japan		
Multiplexing level	Name	# calls	Rate (Mbps)
1	DS1	24	1.544
2	DS2	96	6.312
3	DS3	672	44.736
4	DS4	4032	274.176

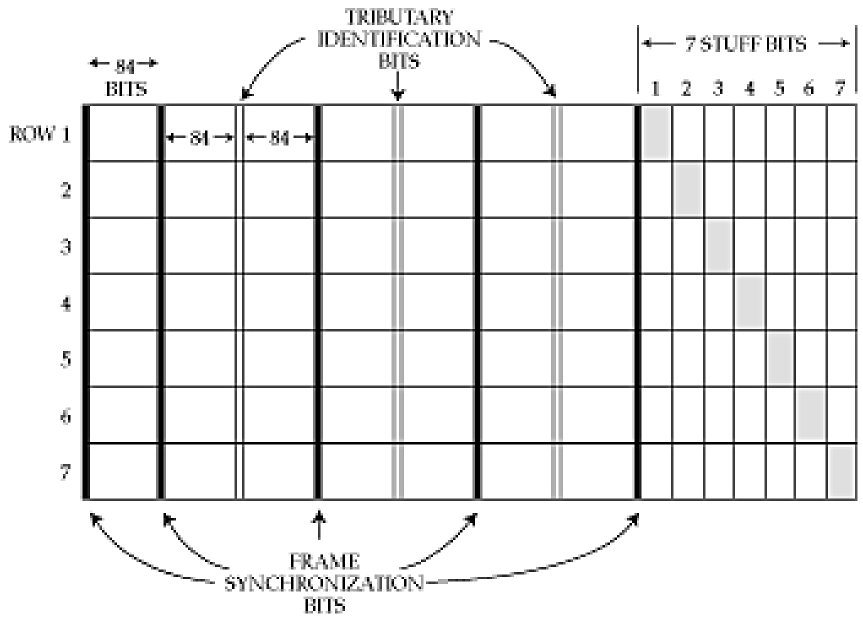
Plesiochronous hierarchy

- Plesiochronous = nearly synchronous
- Tight control on deviation from synchrony
- What if stream runs a little faster or slower?
- Need *justification*



Justification

- Output runs a bit faster always
- Overhead identifies bits from a particular stream
- If a stream runs faster, use overhead to identify it
- Overhead used everywhere except at first level (DS1)



Problems with plesiochrony

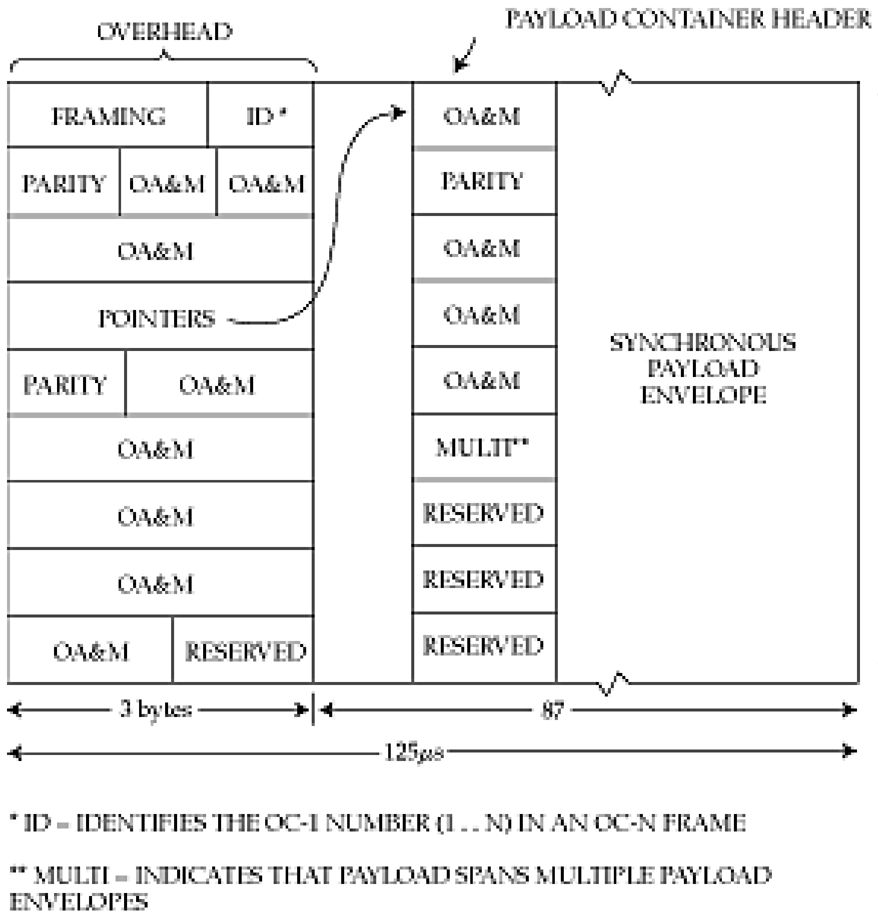
- Incompatible hierarchies around the world
- Data is spread out! Hard to extract a single call
- Cannot switch bundles of calls

Synchronous Digital Hierarchy

- All levels are synchronous
- Justification uses pointers

	Data Rate (Mbps)	US Name
1	51.84	OC-1
2	155.52	OC-3
3	466.56	OC-9
4	622.08	OC-12
5	933.12	OC-18
6	1244.16	OC-24
8	1866.24	OC-36
9	2488.32	OC-48
	9953.28	OC-192

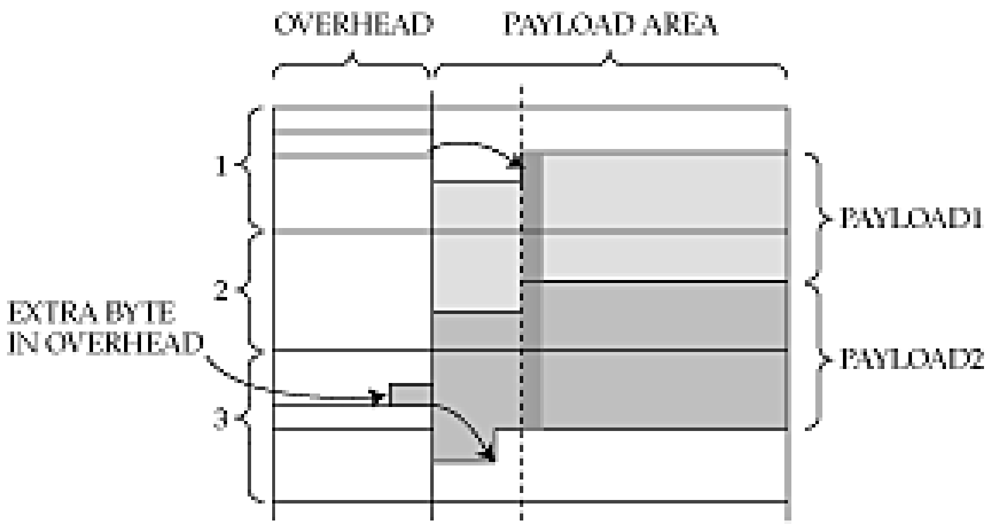
SDH (SONET) frame



SDH

- 9 rows, 90 columns
- Each payload container (SPE) served in 125 microseconds
- One byte = 1 call
- All overhead is in the headers
- Pointers for justification
  - if sending too fast, use a byte in the overhead, increasing sending rate
  - if sending too slow, skip a byte and move the pointer
  - can always locate a payload envelope, and thus a call within it => cheaper add drop mux

SDH justification



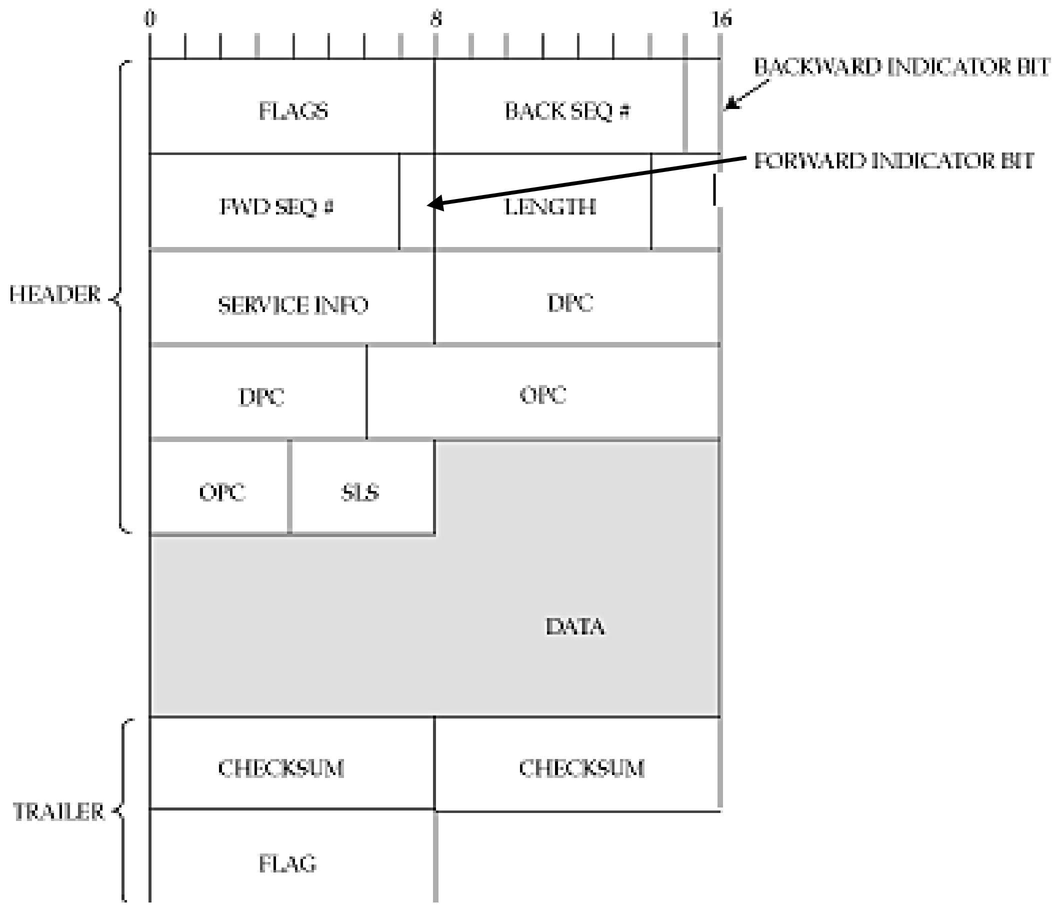
Signaling System 7 (SS7)

OSI layer name	SS7 layer name	Functionality	Internet example
Application	Application Service Element	Application	FTP
	Transaction Capabilities Application part	RPC	RPC
Transport	Signaling Connection Control Part	Connections, sequence numbers, segmentation and reassembly, flow control	TCP
Network	Message Transfer Part 3 (MTP-3)	Routing	IP
Datalink	MTP-2	Framing , link-level error detection and retransmission	Ethernet
Physical	MTP-1	Physical bit transfer	Ethernet

SS7 example

- Call forwarding
- To register
  - call special number
  - connects to ASE
  - authenticates user, stores forwarding number in database
- On call arrival
  - call setup protocol checks database for forwarding number
  - if number present, reroutes call
- SS7 provides all the services necessary for communication and coordination between registry ASE, database, and call setup entity

MTP Header


0
8
16
BACKWARD INDICATOR BIT
FORWARD INDICATOR BIT
FLAGS
BACK SEQ #
FWD SEQ #
LENGTH
HEADER
SERVICE INFO
DPC
DPC
OPC
OPC
SLS
DATA
CHECKSUM
CHECKSUM
TRAILER
FLAG

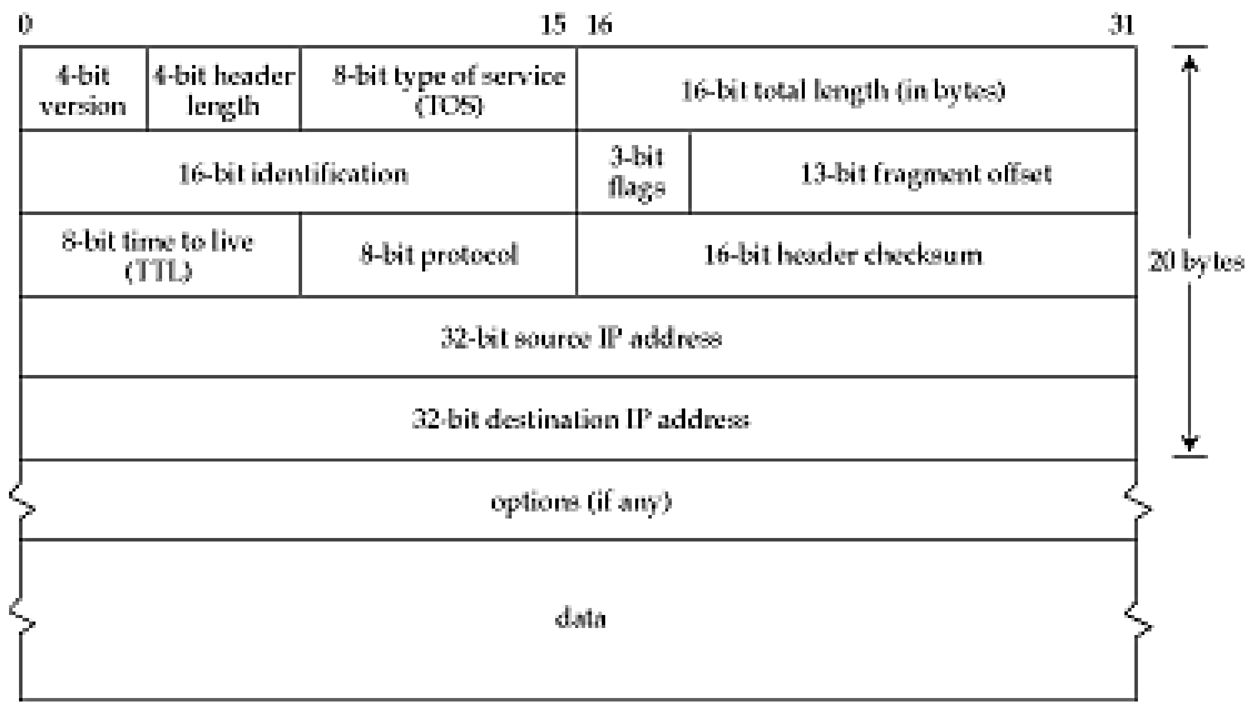
Internet stack

	Data Plane	*Control Plane*
App	HTTP	RSVP/OSPF
Session	Sockets/Streams	
Transport	TCP/UDP	
Network	IP	IP/ICMP
Datalink	Many	Many
Physical	Many	Many

IP

- Unreliable
- Best effort
- End-to-end
- IP on everything- interconnect the world

IP


0
15
16
31
4-bit version
4-bit header length
8-bit type of service (TOS)
16-bit total length (in bytes)
16-bit identification
3-bit flags
13-bit fragment offset
8-bit time to live (TTL)
8-bit protocol
16-bit header checksum
32-bit source IP address
32-bit destination IP address
options (if any)
data
20 bytes

Fragmentation

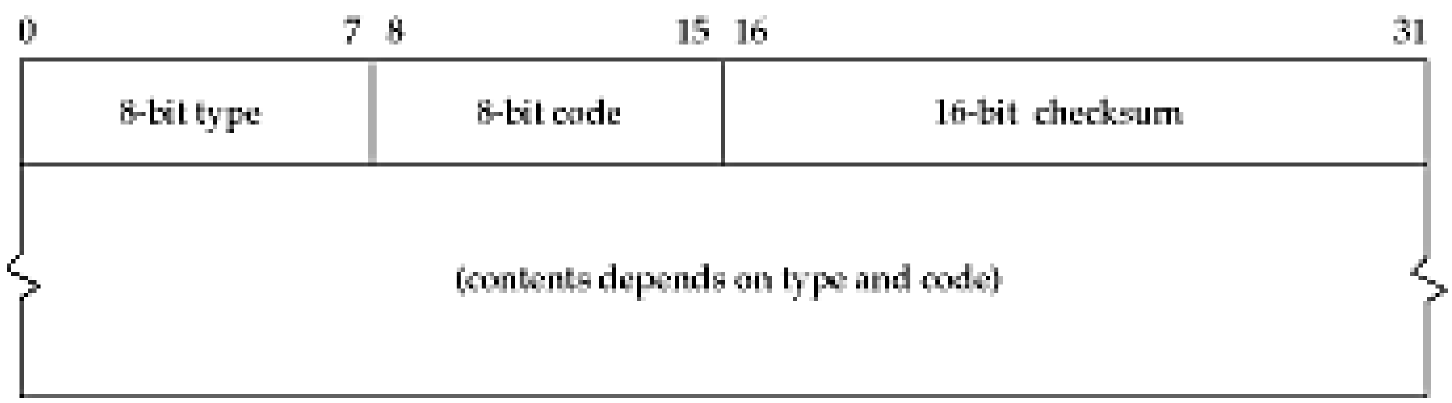
- IP can fragment, reassemble at receiver
- Fragment offset field
- More fragments flag and Don’t fragment flag
- Reassembly lockup
  - decrement timer and drop when it reaches 0
- Fragmentation is harmful
  - extra work
  - lockup
  - error multiplication
- Path MTU discovery
  - send large pkt with Don’t fragment set
  - if error, try smaller

IP fields

- TTL
  - decremented on each hop
  - decremented every 500 ms at endpt
  - terminates routing loops
- Traceroute
  - if router decrements to 0, send ICMP error packet
  - source sends packets with increasing TTL and waits for errors
- Options
  - record route
  - timestamp
  - loose source routing

ICMP

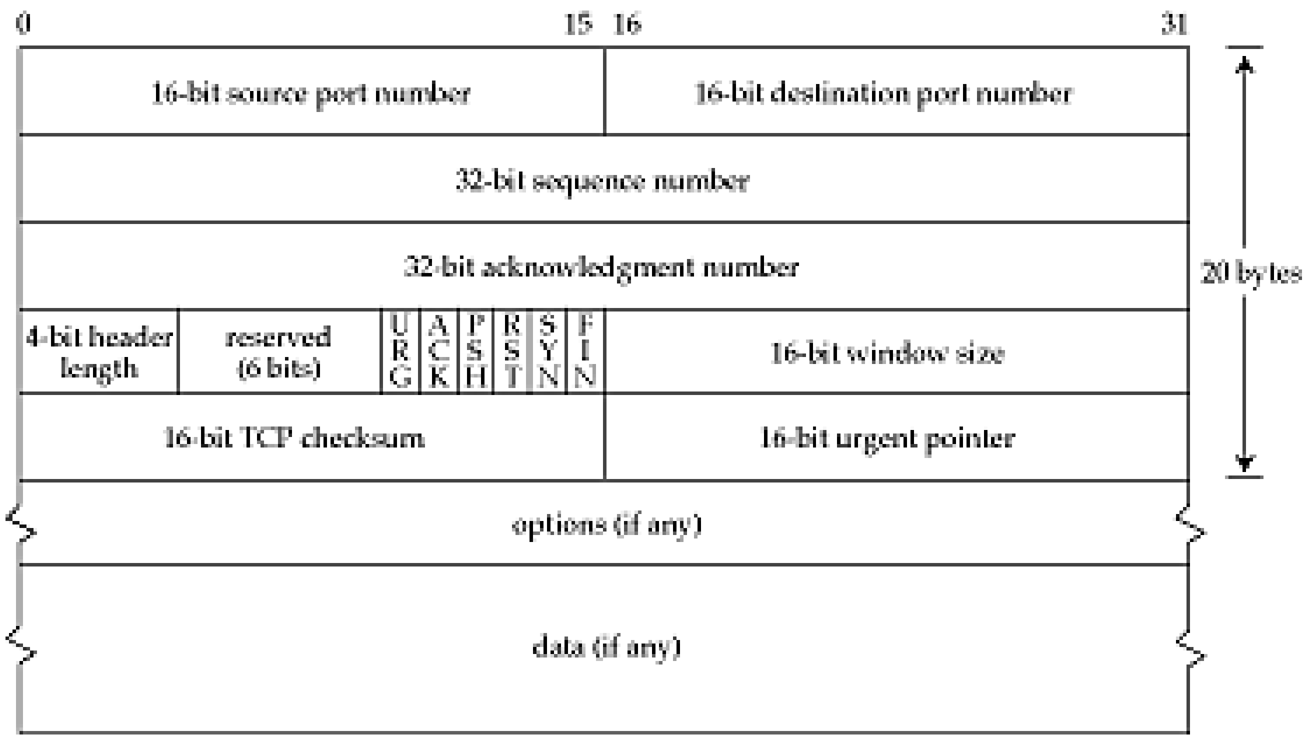
- Destination unreachable
- Source quench
- Redirect
- Router advertisement
- Time exceeded (TTL)
- Fragmentation needed, but Dont frag flag set



TCP

- Multiplexed
- Duplex
- Connection-oriented
- Reliable
- Flow-controlled
- Byte-stream

TCP


0
15
16
31
16-bit source port number
16-bit destination port number
32-bit sequence number
32-bit acknowledgment number
4-bit header length
reserved (6 bits)
URG
ACK
PSH
RST
SYN
FIN
16-bit window size
16-bit TCP checksum
16-bit urgent pointer
20 bytes
options (if any)
data (if any)

Fields

- Port numbers
- Sequence and ack number
- Header length
- Window size
  - 16 bits => 64 Kbytes (more with scaling)
  - receiver controls the window size
  - if zero, need sender persistence
  - silly window syndrome
- Checksum
- Urgent pointer
- Options
  - max segment size

HTTP

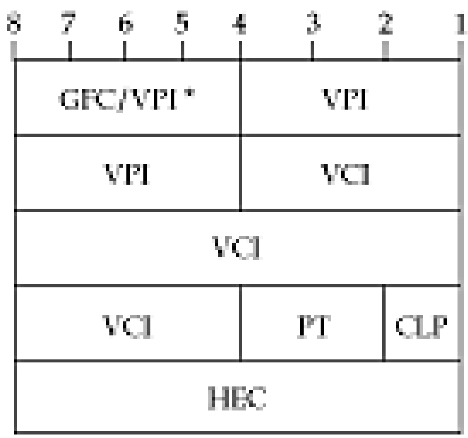
- Request response
- Protocol is simple, browser is complex
- Address space encapsulation
- Request types
  - GET
  - HEAD
  - POST
- Response
  - status
  - headers
  - body

ATM stack

	Data Plane	*Control Plane*
Application		UNI/PNNI
Application		Q.2931
Session		
Transport		SSCOP
Network	AAL1-5	S-AAL (AAL5)
Data Link	ATM	ATM
Physical	Many	Many

ATM

- Connection-oriented
- In-sequence
- Unreliable
- Quality of service assured



*GFC IN UNI & VPI IN NNI

Virtual paths

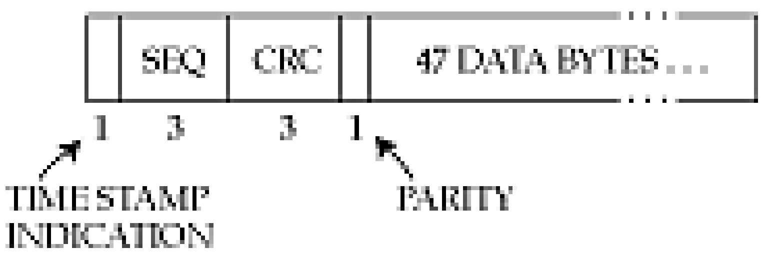
- High order bits of VCI
- All VCIs in a VP share path and resource reservation
- Saves table space in switches
  - faster lookup
- Avoids signaling
- May waste resources
- Dynamic renegotiation of VP capacity may help
- Set of virtual paths defines a *virtual private network*

AAL

- Was supposed to provide “rest of stack”
- Scaled back
- 4 versions: 1, 2, 3/4, 5
- Only 1, 3/4 and 5 important in practice

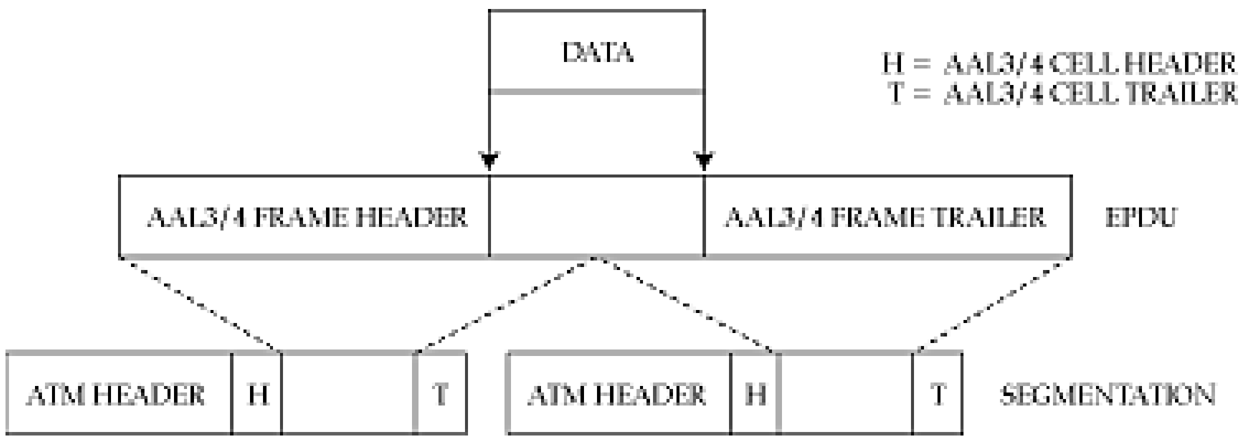
AAL 1

- For synchronous apps
  - provides timestamps and clocking
  - sequencing
  - always CBR
  - FEC in data bytes



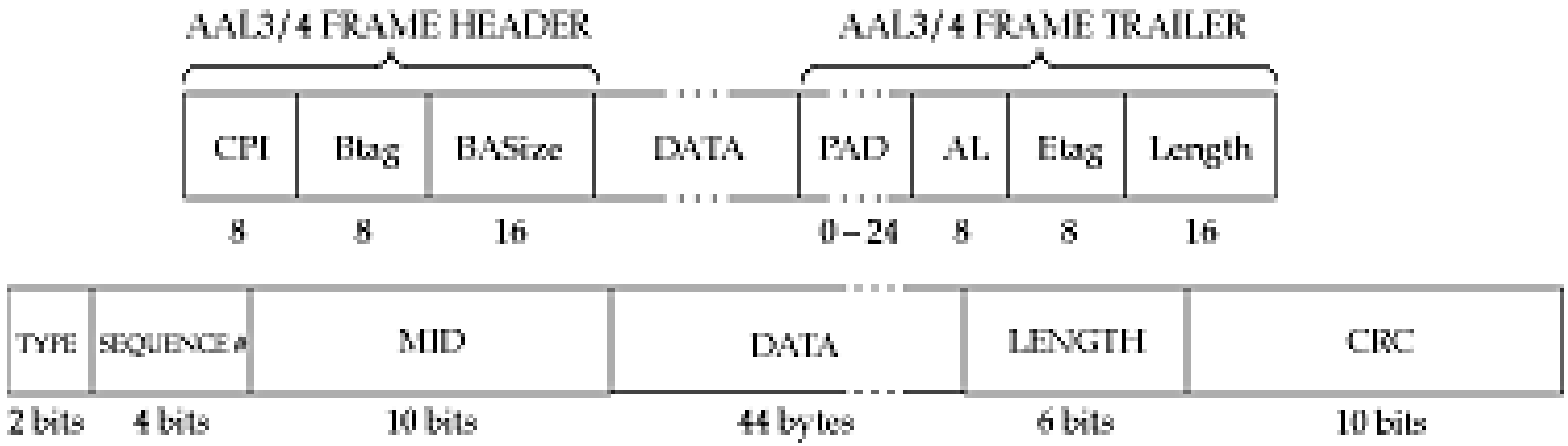
AAL 3/4

- For data traffic (from a telco perspective!)
- First create an encapsulated protocol data unit EPDU
  - (common part convergence sublayer-protocol data unit CPCS-PDU)
- Then fragment it and add ATM headers



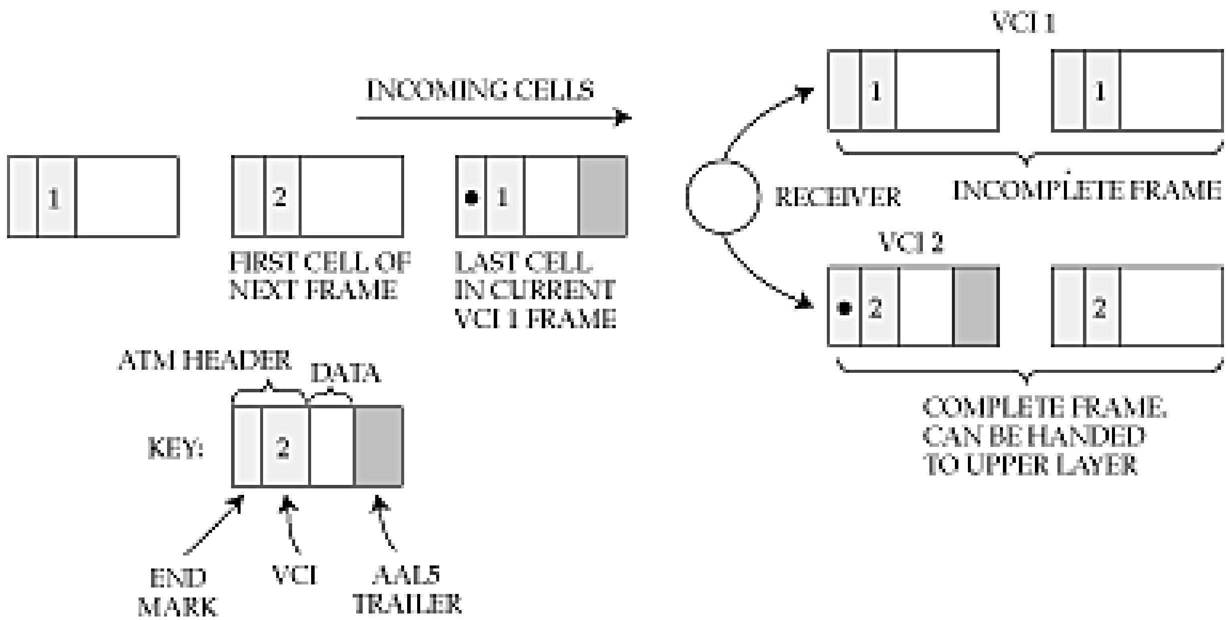
AAL 3/4

- Error detection, segmentation, reassembly
- Header and trailer per EPDU *and* per-cell header!

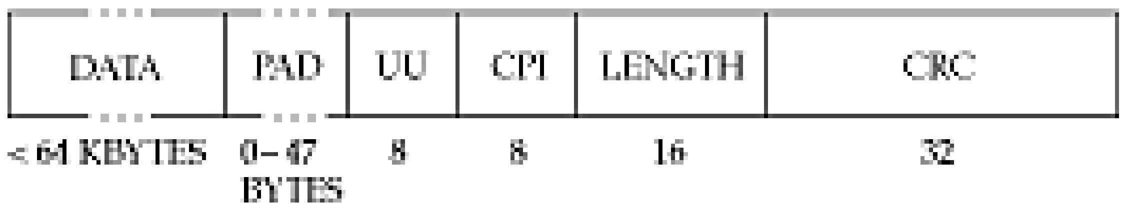


AAL 5

- Violates layering, but efficient
- Bit in header marks end of frame



AAL5 frame format



SSCOP

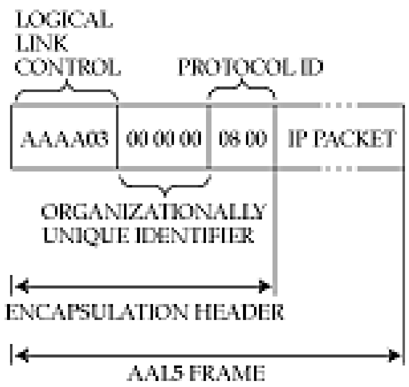
- Reliable transport for signaling messages
- Functionality similar to TCP
  - error control (described below)
  - flow control (static window)
- Four packet types
  - sequenced data / poll / stat / ustat
- No acks!
- Sender polls, receiver sends status
  - includes cumulative ack and window size
- If out of order, sends unsolicited status (ustat)
- Key variable is poll interval

IP-over-ATM

- Key idea: treat ATM as a link-level technology
  - ignore routing and QoS aspects
- Key problems
  - ATM is connection-oriented and IP is not
  - different addressing schemes
  - ATM LAN is point-to-point while IP assumes broadcast
- Basic technologies
  - IP encapsulation in ATM
  - Resolving IP addresses to ATM addresses
  - Creating an ATM-based IP subnet
  - Mapping multicast groups to ATM

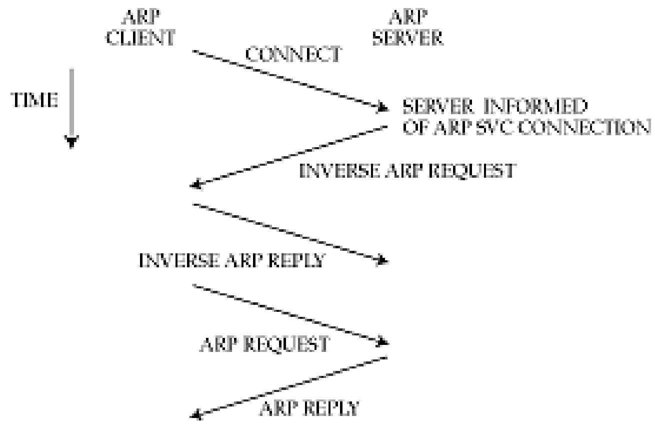
IP encapsulation in ATM

- Put data portion of IP packets in AAL5 frame
  - works only if endpoints understand AAL5
- Instead, place entire IP packet with AAL5 frame
- General solution allows *multiprotocol encapsulation*



Resolving IP addresses to ATM addresses

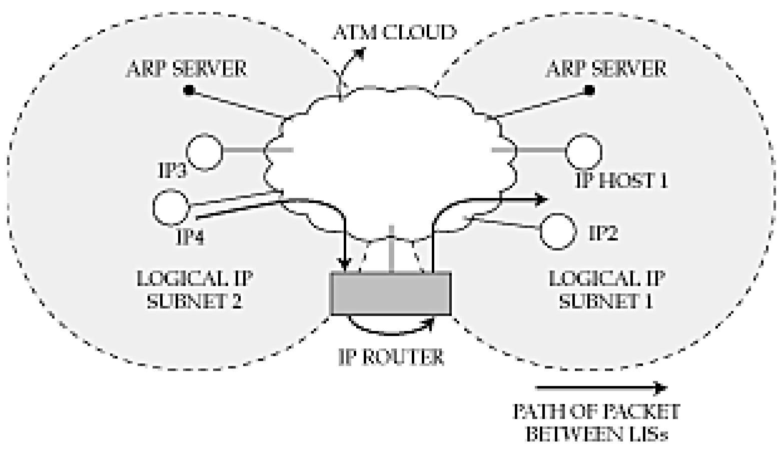
- Need something like ARP, but can't use broadcast
- Designate one of the ATM hosts as an ARP server



- Inverse ARP automatically creates database

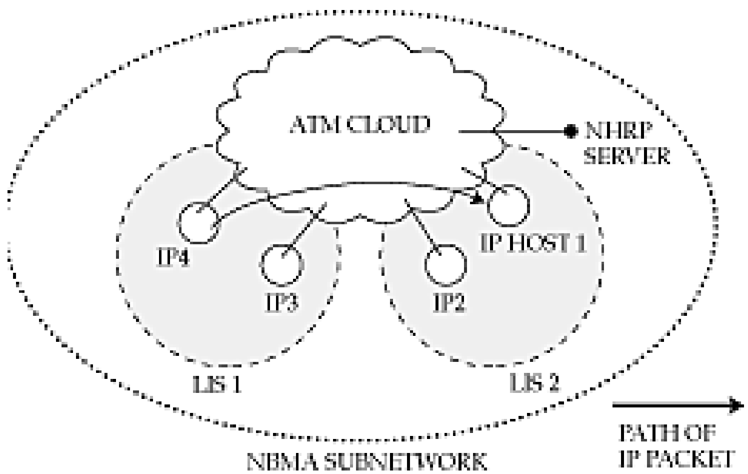
Creating an ATM-based IP subnet

- IP assumes free availability of bandwidth within a subnet
- If all hosts on ATM are on same IP subnet, broadcast reaches all => congestion
- Partition into *logical IP subnets*
  - at the cost of longer paths between ATM-attached hosts



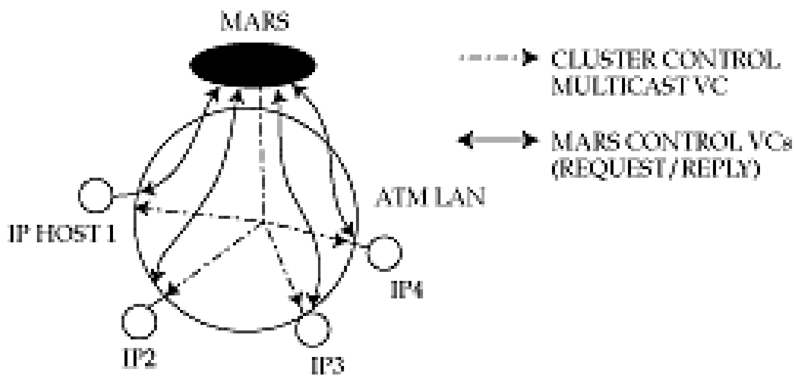
Next-hop routing

- Avoids long paths
- Next-hop server stores IP-to-ATM translations independent of subnet boundaries
  - like DNS



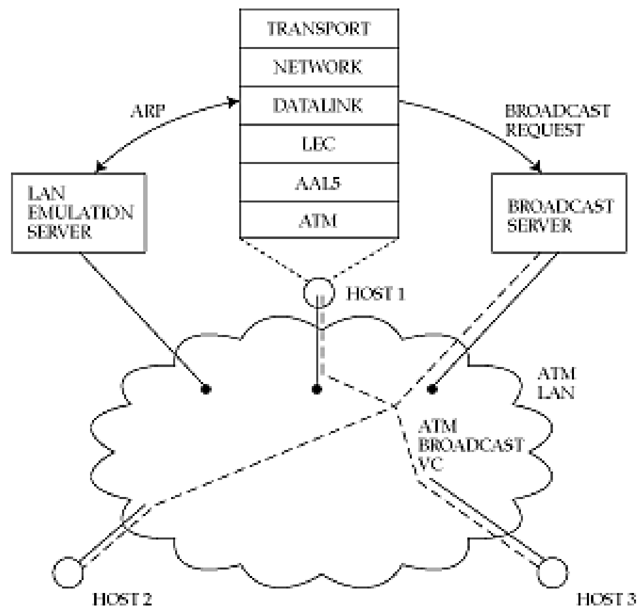
Resolving multicast addresses

- ARP server cannot resolve multicast addresses (why?)
- Actively maintain set of endpoints that correspond to a particular Class D address
- *Multicast Address Resolution Server* provides and updates this translation



LAN emulation

- If destination is on same LAN, can use ATM underneath datalink layer
- Need to translate from MAC address to ATM address
- Also need to emulate broadcast for Ethernet/FDDI



Cells in Frame (CIF)

- Solutions so far require expensive ATM host-adapter card
- Can we reuse Ethernet card?
- Encapsulate AAL5 frame in Ethernet header on point-to-point Ethernet link
- CIF-Attachment Device at other end decapsulates and injects the frame into an ATM network
- Software on end-system thinks that it has a local host adapter
- *Shim* between ATM stack and Ethernet driver inserts CIF header with VCI and ATM cell header
  - may need to fragment AAL5 frame
  - can also forward partial frames
- Cheaper
  - also gives endpoints QoS guarantees, unlike LANE

Holding time problem

- After resolution, open an ATM connection, and send IP packet
- When to close it?
- Locality
  - more packets likely
  - hold the connection for a while to avoid next call setup
  - but pay per-second holding time cost
- Optimal solution depends on pricing policy and packet arrival characteristics
- Measurement-based heuristic works nearly optimally
  - create the inter-arrival time histogram
  - expect future arrivals to conform to measured distribution
  - close connection if expected cost exceeds expected benefit