Common Protocols

Digital Communications II

Michaelmas Term 2007
Based on Prof. Jon Crowcroft's notes, and thus transitively on
S. Keshav's "An Engineering Approach to Computer Networking"

Explore some common protocols

- Much discussion of 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
Session		TCAP
Transport		
Network		SCCP/MTP-3
Datalink	SDH/Sonet/PDH	MTP-2
Physical	Many	MTP-1

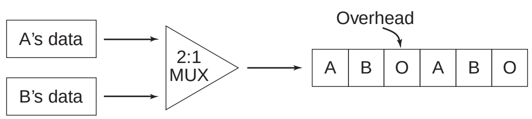
Traditional digital transmission

- Long distance trunks carry multiplexed calls
- Standard multiplexing levels
- Digital transmission hierarchy
  - DS0 is 64Kbps (a single call)

	US and Japan			Europe		
Digital Signal level (i.e. multiplexing)	T class.	# calls	Rate (Mbps)	E class.	# calls	Rate (Mbps)
DS1	T1	24	1.544	E1	32	2.048
DS2	T2	96	6.312	E2	128	8.448
DS3	T3	672	44.736	E3	512	34.368
DS4	T4	4032	274.176	E4	2048	139.264
DS5				E5	8192	565.148

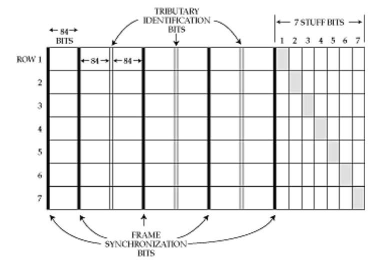
Plesiochronous hierarchy

- 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

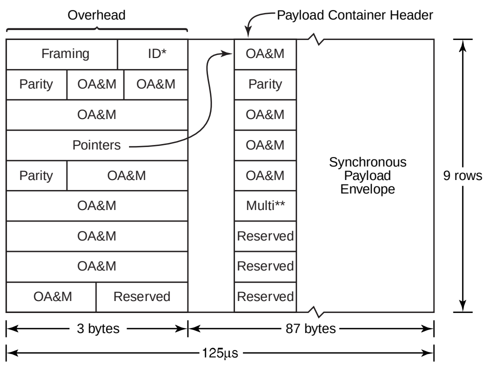
- Incompatible hierarchies around the world
- Data is spread out! Hard to extract a single call
- Cannot switch bundles of calls

Synchronous Digital Hierarchy

- All levels are synchronous – requires use atomic clocks
- Justification uses pointers

Data Rate (Mbps)	SONET Name	SDH Name
51.84	OC-1	STM-0
155.52	OC-3	STM-1
466.56	OC-9	n/a
622.08	OC-12	STM-4
933.12	OC-18	n/a
1244.16	OC-24	STM-8
1866.24	OC-36	n/a
2488.32	OC-48	STM-16
9953.28	OC-192	STM-64

SDH (SONET) frame

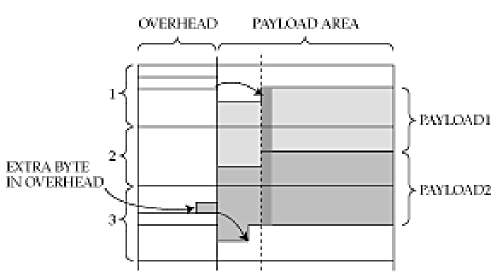


* ID field identifies the OC-1 number (1..N) in an OC-N frame
** Multi field indicates that payload spans multiple payload envelopes

SDH

- 9 rows, 90 columns
- Each payload container (SPE) served in 125 microseconds
- One byte = 1 call (1*8*(1000/125) kbit/s = 64kbit/s)
- 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



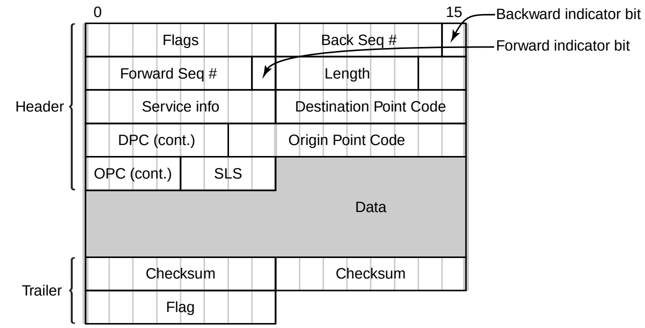
Signaling System 7 (SS7)

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 (Application Service Element)
  - 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 (Message Transfer Part) Header



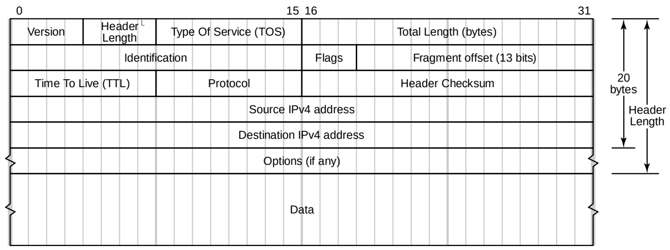
Internet stack

	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

IPv4



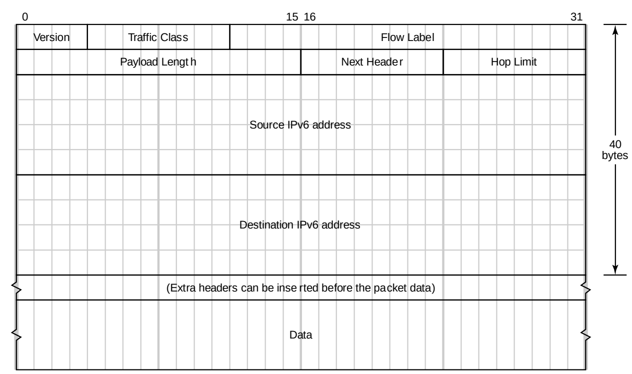
Fragmentation

- 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 packet with Don't fragment flag set
  - If ICMP error, try smaller

IP fields

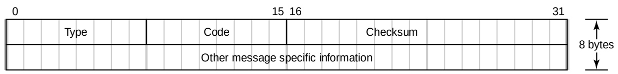
- TTL
  - Decremented on each hop
  - Decremented every 500 ms at endpoint
  - Terminates routing loops
- Traceroute
  - If router decrements to 0, send ICMP error packet
  - Source sends packets with increasing TTL and waits for errors
- Options
  - Record route
  - Timestamp
  - Loose source routing

IPv6



ICMP

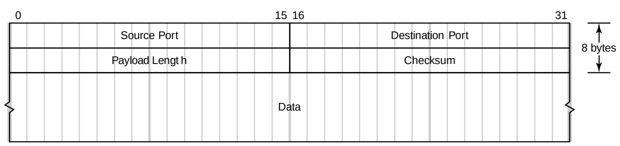
- Destination unreachable
- Source quench
- Redirect
- Router advertisement
- Time exceeded (TTL)
- Fragmentation needed, but Don't fragment flag set



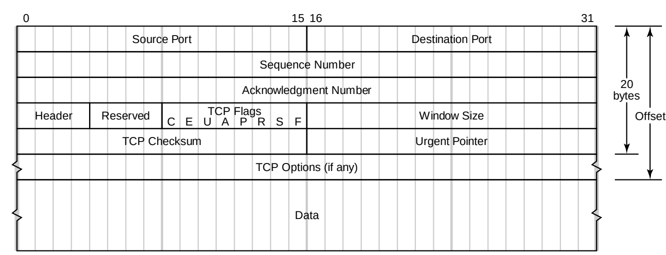
TCP

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

UDP



TCP



Fields

- Port numbers (16 bits each)
- Sequence and ACK number (32 bits each)
- Header length
- Window size
  - 16 bits → 64 Kbytes (more with scaling)
  - Receiver controls the window size
    - If 0: sender persistence timer (window change ACK)
  - *Silly window syndrome* – window reduces to tiny size
- Checksum
- Urgent pointer – seldom used (if not counting WinNuke)
- Options
  - Max segment size, window scale, …

HTTP

- Request/response – stateless connections
- Protocol is simple, browser is complex (and usually buggy)
- Address space encapsulation
- Request types
  - GET, HEAD, POST
  - PUT, DELETE
  - CONNECT, TRACE, OPTIONS
- Response
  - Status
  - Headers
  - Body

HTTP's WWW success

- Representational State Transfer (REST) approach
- Initially transfer was stateless
  - Facilitates efficient caching
  - Transparent proxies
- Cookies can violate statelessness
  - Observe different behaviours of the browser "back" button
- In contrast to RPC
  - Resource names are more complex
    - E.g. URIs that include attributes
  - Fewer specific action types
    - E.g. mostly GET, POST – very different from RPC

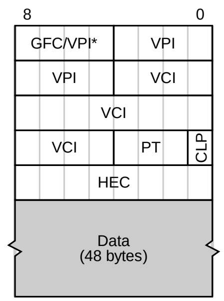
ATM stack

	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

- In User-Network Interface (UNI) cells there is a GFC field
  - Generic Flow Control
- In Network-Network Interface (NNI) cells the first byte is VPI data only



Virtual paths

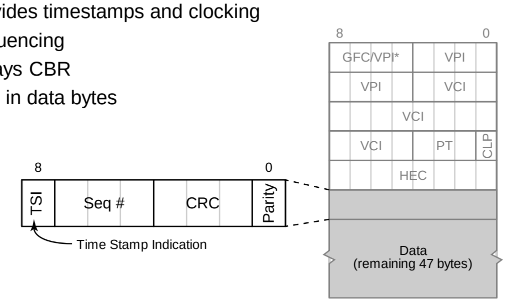
- VP essentially the high order bits of VCI
  - All VCIs in a VP share path and resource reservation
- Saves table space in switches
  - Faster lookup
- Avoids signalling
- May waste resources
- Dynamic renegotiation of VP capacity may help
- Set of virtual paths defines a *virtual private network*

ATM Adaptation Layer (AAL)

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

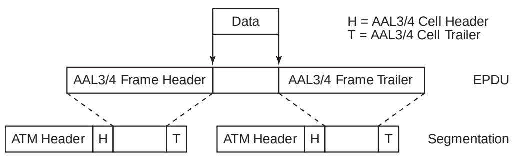
AAL 1

- 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)
  - (I.e. 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 3/4 Frame

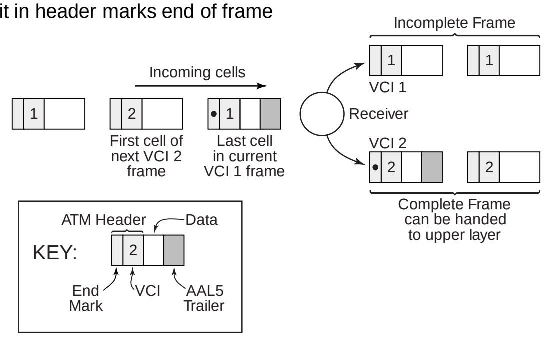
AAL 3/4 Frame Header				AAL 3/4 Frame Trailer			
CPI	Btag	BASize	Data	Pad	AL	Etag	Length
8	8	16		0-24	8	8	16

AAL 3/4 Cell

Type	Seq #	MID	Data	Length	CRC
2 bits	4 bits	10 bits	44 bytes	6 bits	10 bits

AAL 5

- Violates layering, but efficient (as a direct consequence)
- Bit in header marks end of frame



AAL5 frame format

AAL 5 EPDU

Data	Pad	UU	CPI	Length	CRC
<64Kbytes	0-47 bytes	8 bits	8 bits	16 bits	32 bits

Service Specific Connection Oriented Protocol (SSCOP)

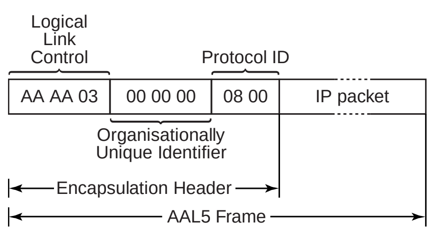
- Reliable transport for signalling messages
- Functionality similar to TCP
  - Error control (described below)
  - Flow control (static window)
- Four packet types
  - Sequenced data / poll / stat / ustat
- No ACKs (!)
- Sender polls, receiver sends status
  - Includes cumulative ACK and window size
- 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

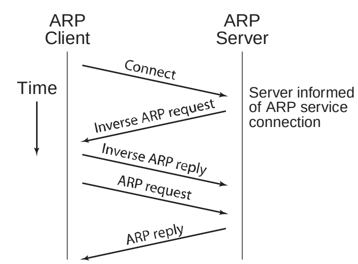
IP encapsulation in ATM

- 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 *multi-protocol encapsulation*



Resolving IP addresses to ATM addresses

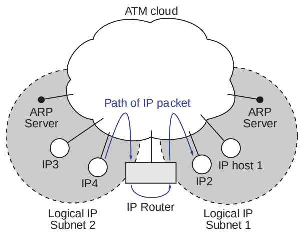
- Need something like ARP, but can't use broadcast
- Designate one of the ATM hosts as an ARP server



- Inverse ARP automatically creates database

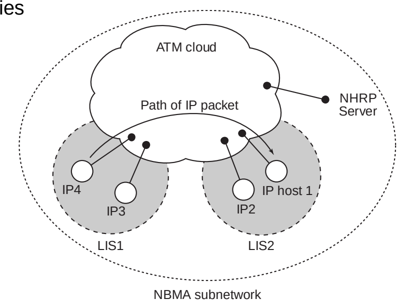
Creating an ATM-based IP subnet

- 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



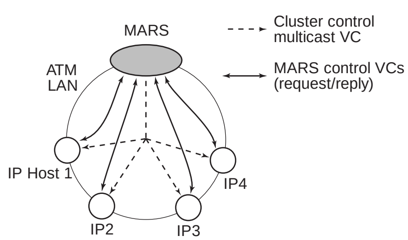
Next-hop routing

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



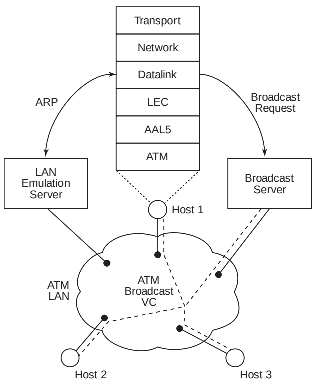
Resolving multicast addresses

- 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 principle
  - 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