Digital Communications II

Ian Pratt

Michaelmas 2003

Computer Science Tripos Part II

Asynchronous Transfer Mode
Router Architecture
Access Networks
Wide-Area Networks

Remaining Topics

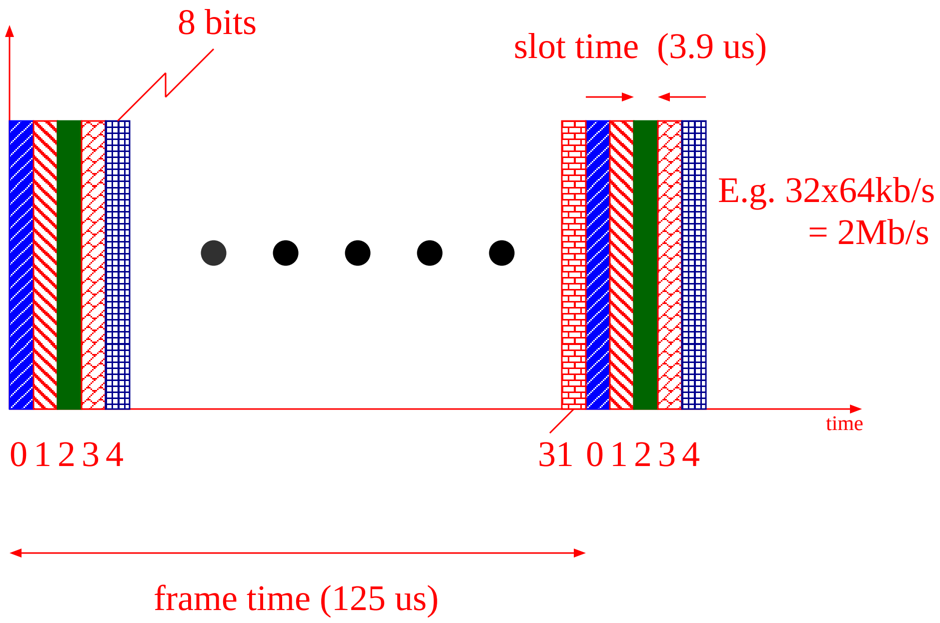
1. ATM Case-study
   - ATM *vs.* PTM *vs.* STM
   - virtual circuits *vs.* datagrams
   - QoS : CBR, VBR, ABR
2. Router Architecture
   - high-performance router design
   - longest prefix match
   - switch fabrics
3. Access Networks
   - POTS *vs.* ADSL *vs.* Cable modems
   - 802.11b, 3G, Satellite
4. Wide Area Networks
   - photonics and long haul links
   - DWDM
   - SDH/SONET

Asynchronous Transfer Mode Casestudy

ATM Case-study Overview

- Time Division Multiplexing
  - STM *vs.* PTM *vs.* ATM
- B-ISDN ATM
  - Cells and headers
  - Virtual Channels and Paths
  - ATM Adaption Layers
  - Quality of Service
  - Signalling
  - ABR
- Reflections on ATM
  - What ever happened to ATM Everywhere?
  - ATM *vs.* IP

Synchronous Transfer Mode

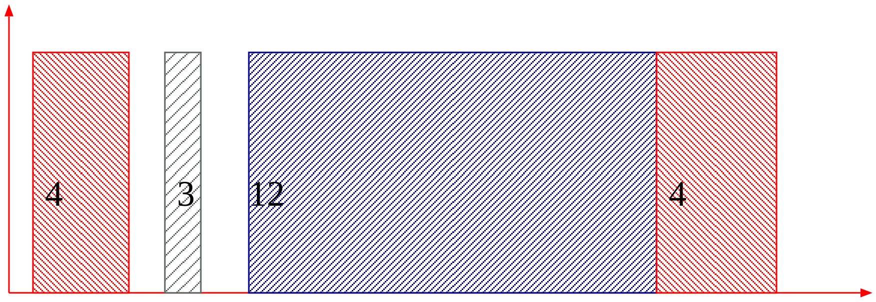


- time divided into *frames*, frames contain slots
- each time slot is a *circuit*
- data rate of a circuit constant, as is delay
- can group circuits, or multiplex circuits further
- switch according to pre-determined schedule
  - channel indicated by *when* symbols appear, not what they are
  - requires out-of-band signalling

⇒ Circuit Switching

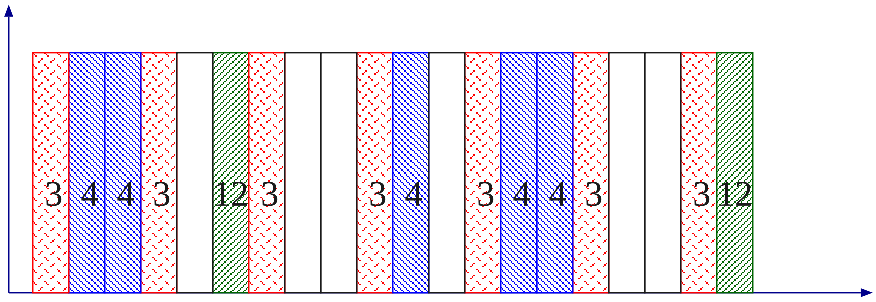
- (S)TDM
- inefficient for variable bandwidth use?

Packet Transfer Mode



- variable sized data units
- in-band addressing
  - some symbols in group must indentify subchannel
  - no signalling strictly necessary
    - datagrams
  - alternatively, use *virtual channels*
- statistical multiplexing gain
  - no need to make peak reservations
  - but, QoS tricky
- media access jitter bounded by max packet size
- Packet Switches harder to build. Need:
  - address decoder
  - complex buffering
  - arbiter

Asynchronous Transfer Mode



- time is divided into fixed sized slots called *cells*
  - bounds jitter
  - bounds loss of efficiency for packet data
  - Cambridge fast ring, 32 byte payload (+4)
  - B-ISDN ATM, 48 byte payload (+5)
- slots must contain addressing information
  - compact representation beneficial
  - lightweight virtual circuits or *virtual channels*
  - cells tagged with VC identifier
- VCIs assigned through signalling
  - typically allocate resources for VC too
  - hence, by prior arrangement different VCs can be treated differently by network
- Need segmentation and reassembly for data
  - packet delimiter (in-band or in header)

Why ATM?

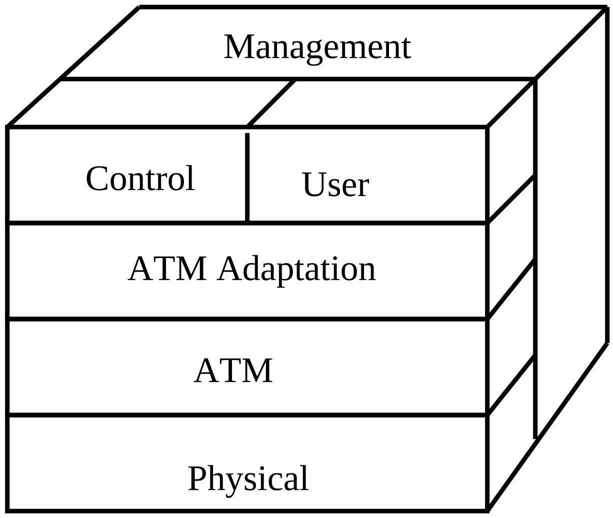
	STM	ATM	PTM
	circuit	cell	packet
framing	fixed	fixed	variable
access	fixed	variable	variable
bandwidth	fixed	fixed or variable	variable
delay	fixed	fixed or variable	variable
mux gain	no	yes	yes
rate adaption	no	yes	yes
connection setup	req'd	req'd	optional
understanding	good	so-so	so-so
fast switch design	easy	hard	hardest
synchronous services	ideal	fair	bad(?)
VBR (e.g. media)	no	good	so-so
elastic data	possible	so-so to good	ideal(?)

- *The ATM Compromise:* Can carry circuits and packets. Pays a penalty for both, but single switching and multiplexing mechanism.

- *The ATM Advantage:* Can support range of desired channel characteristics between circuit and packet, in particular variable rate delay sensitive traffic.

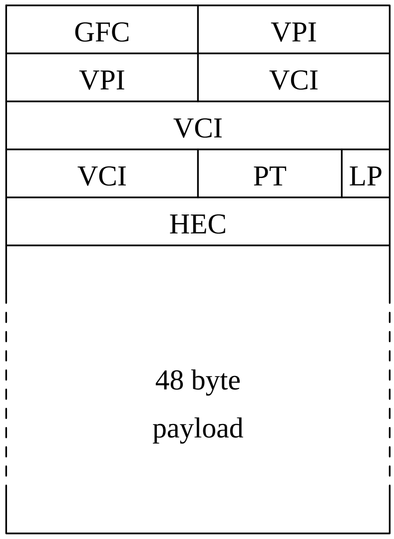
- *ATM Flexibility:* We have no idea what the traffic demands on networkswill be in the future. ATM may not be optimal, but it will be able to cope and it will be in place.

B-ISDN Protocol Reference Model



- AAL
  - segmentation and reassembly
- ATM
  - cell header generation/extraction
  - cell VPI/VCI translation
  - cell multiplex and demultiplex
- physical
  - cell rate decoupling
  - HEC header generation /verification
  - cell delineation
  - transmission frame generation/recovery
  - bit timing/encoding, physical medium

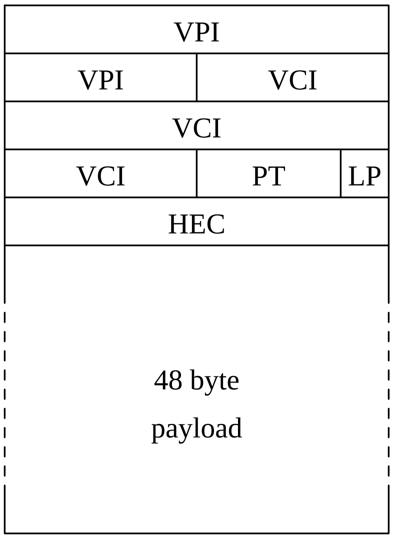
BISDN header I



Today's UNI cell format

- Generic Flow Control
  - to support multiaccess
  - not going to be used?
- routing
  - virtual circuit
  - virtual path

BISDN header II



Today's NNI cell format

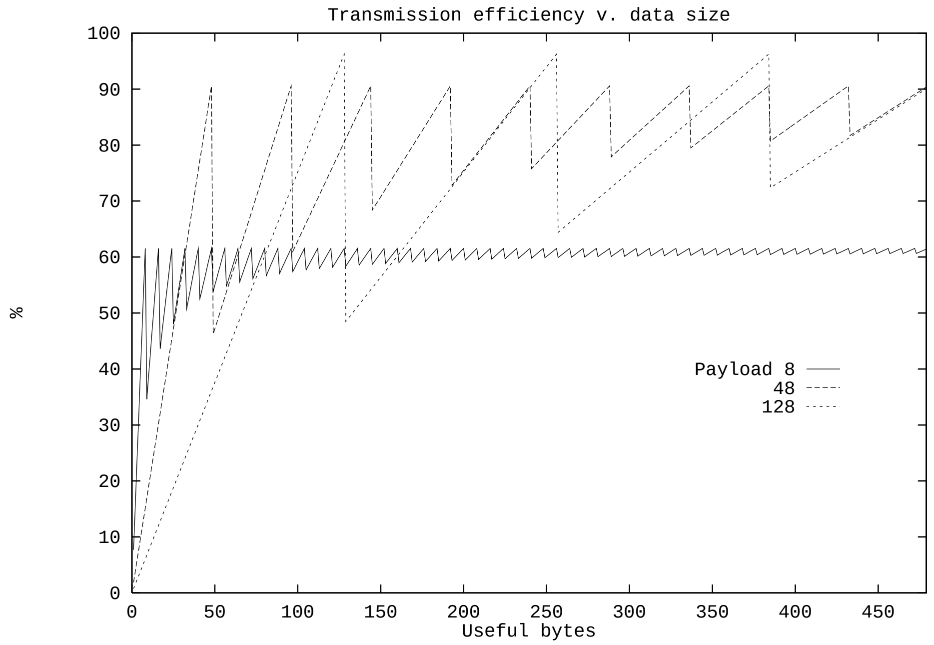
- Payload Type
  - management cell v. user cell
  - user cell then two bits . . .
    - congestion experienced
    - *The Bit* - user data

- Cell Loss Priority
  - network/end system usage
  - semantic priority

- Header Error Check
  - 8 bit CRC

Why 48?

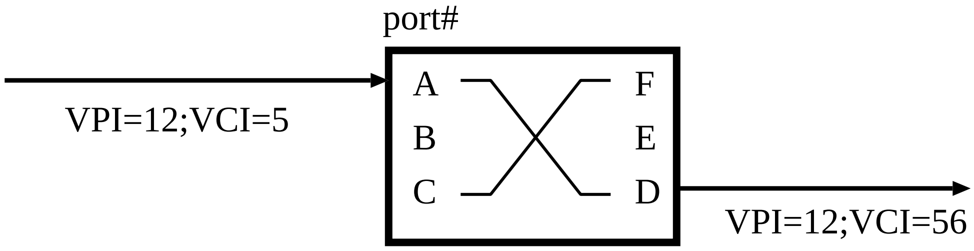
- packetization delay for streams
  - voice echo cancellation
- access jitter
  - only relevant on low-bandwidth links
- efficiency
  - amortizing header tax *vs.* avoiding wasteage
  - E.g. for 5 byte headers:

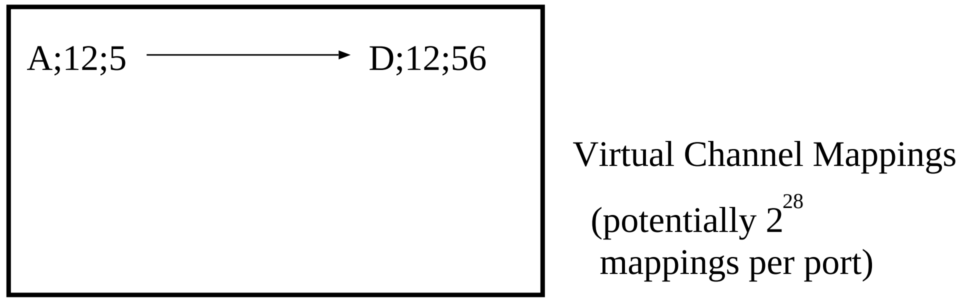


Virtual Channel Connections

- a *virtual channel* is the logical channel made up of the sequence of ATM cells on a link which have the same VPI/VCI.

- a *virtual channel connection* is an end to end concatenation of of one or more VCs.





- an ATM VC switch concatenates virtual channels together. It must identify each virtual channel and thus must examine both the VPI and VCI fields. It will in general overwrite the VPI and VCI fields as it forwards cells.

⇒ link-local VPI/VCI allocation

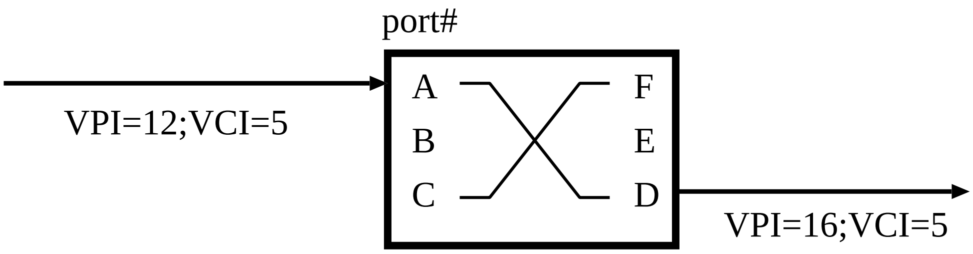
- by convention, a bi-directional pair of VC's are usually created

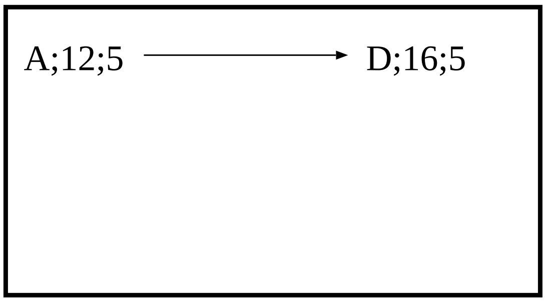
- established manually or using signalling

Virtual Paths and Virtual Path Connections

- a *virtual path* is a collection of up to 2^{16} virtual channels

- a *virtual path connection* is an end to end concatenation of of one or more virtual paths.





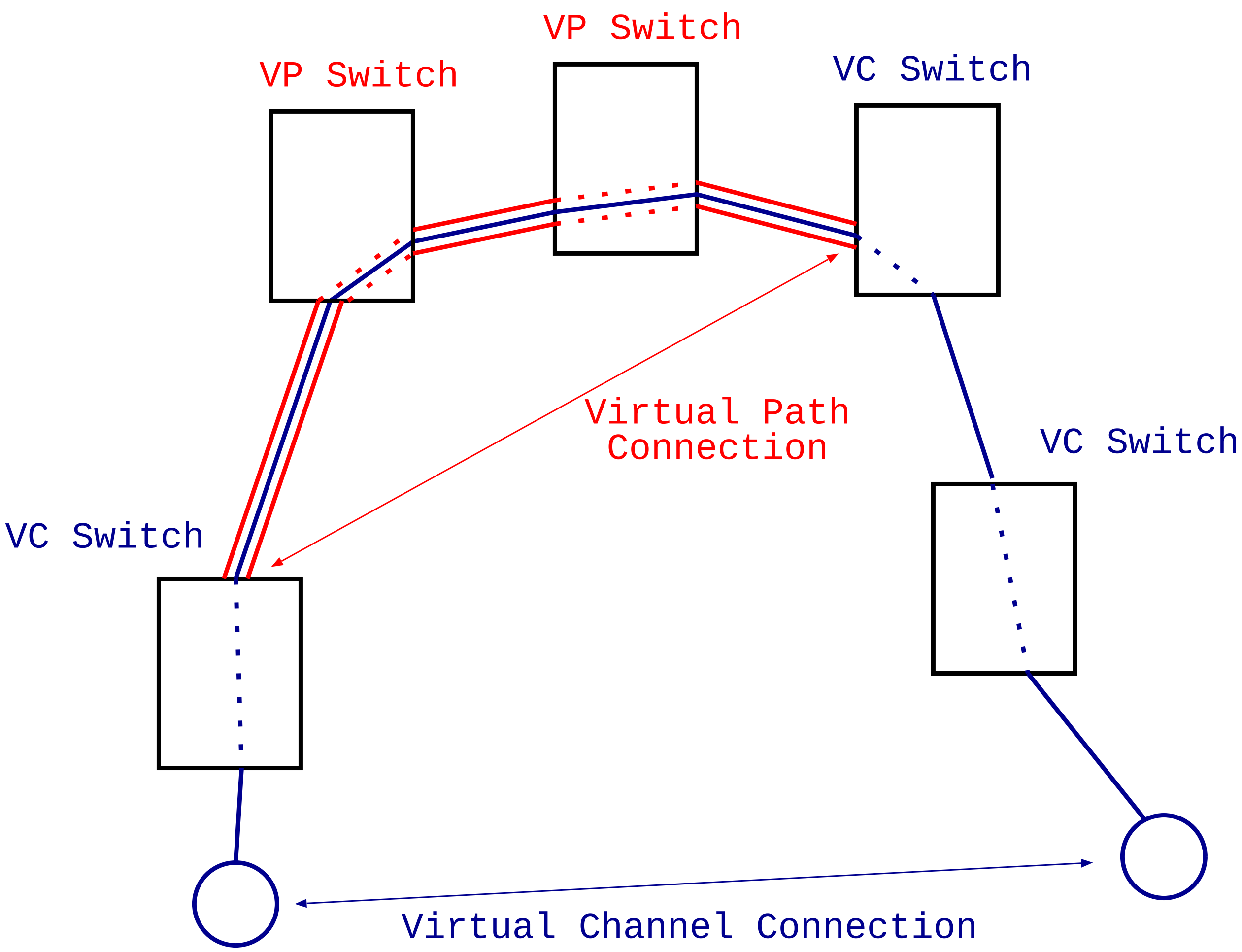
Virtual Path Mappings

(potentially 2^{12} mappings per port)

- an ATM VP switch concatenates virtual paths together. The constituent virtual channels in the paths are thus linked en masse without the switch looking at VCIs' it need only look at VPIs and must preserve the VCI value

- an ATM VP/VC switch does both. It looks at the VP field to decide whether to map the entire VP or whether to examine the VCI

Why Virtual Paths?



- eases table space in switches
- reduces need for setting up state in switches
- reduces number of things to manage
- but, virtual path switch cannot offer different service guarantees to different VCs within a VP

ATM Adaptation Layers

Turn ATM into “normal” services.

- Convergence Sub-layer (CS)
- Segmentation and Reassembly (SAR)

Three orthogonal aspects were identified:

- time relation between source and destination (Sync / Async),
- bit rate (Constant / Variable),
- connection mode (Connection Oriented / Connection Less).

Originally from the eight possibilities, four chosen:

Type	Timing	Bit Rate	Mode
1	Sync	Constant	CO
2	Sync	Variable	CO
3	Async	Variable	CO
4	Async	Variable	CL

Current status:

- AAL-1 used for circuit emulation
- AAL-2 abandoned
- AAL-3 and AAL-4 merged
- New AAL-5 protocol most widely used

AAL-1

4 SN	4 SNP	SAR-PDU payload

header 1 octet — payload 47 octets

SN   sequence numper
SNP   sequence number protection (CRC)

- circuit emulation: voice, H.261
- Null Convergence Sublayer (CS)

AAL-3/4

8 CPI	8 Tag	16 BAsize	CPCS-PDU payload	8 Tag	16 Length

CPI   common part indicator: which CS is in use
Tag   Message sequence number, used in header and trailer
BAsz Indicates to receiver maximum buffer size req'd for reassembly
Len   Total CPCS-PDU length

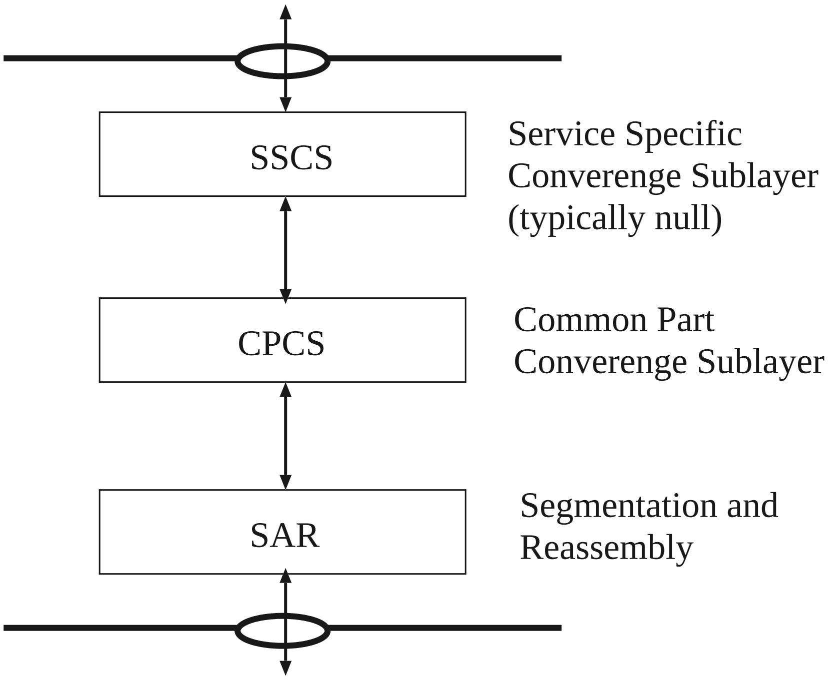
2 ST	4 SN	10 MID	SAR-PDU payload	6 LI	10 CRC

header 2 octets — payload 44 octets — trailer 2 octets

ST   Segment type: Begining of message, end, continuation
SN   Sequence number
MID Message Identifier: enables message multiplexing on a VC
LI   Number of data octets used in this cell (for end of message)

- Multiplexing over VC
  - CO mode / CL mode
- Buffer allocation indicator

AAL-5



- connection oriented
- no per cell overhead in payload

Use *The Bit* to indicate last cell in block

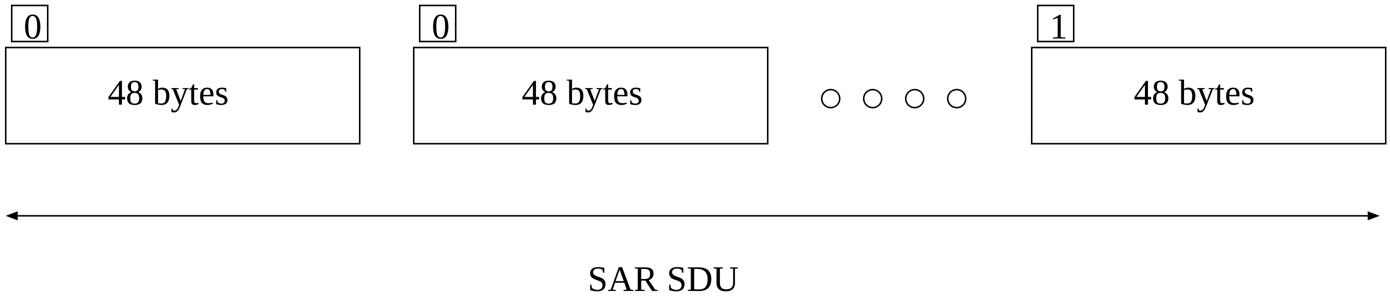
- last 32 bits contain 32 bit CRC
- 16 bit length field, 16 bit reserved
- if multiplexing done, done in SSCS

AAL-5 is the most commonly used AAL:

- most efficient for IP
- easiest to implement in s/w
  - though done in h/w anyway thesedays....

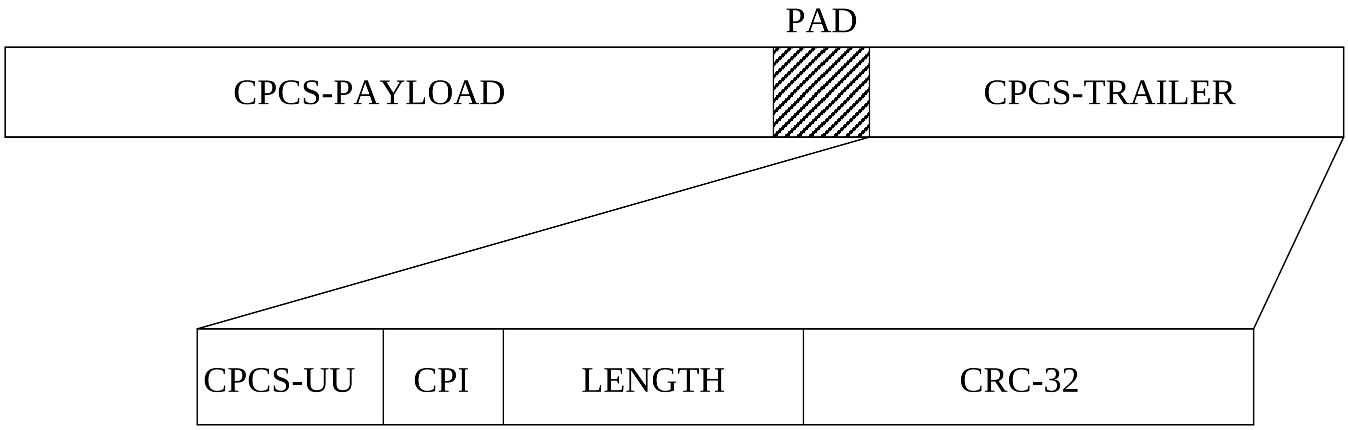
AAL-5 Details

SAR sublayer:



- SAR SDU is made up of the payloads of a sequence of cells, the last one with the AUU indication set to 1

CPCS sublayer:



- PAD: 0..47 bytes, pad trailer to cell boundary

- Reserved: 2 bytes (CPCS-UU, etc.)

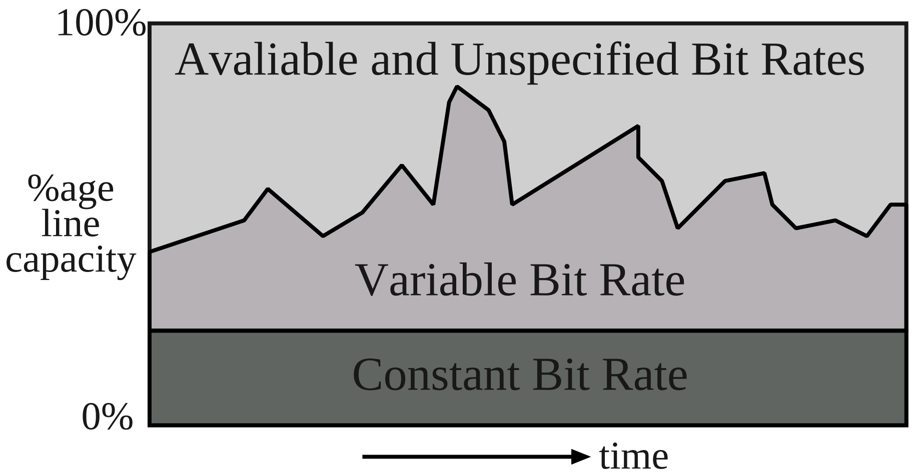
- LENGTH: 2 bytes

- CRC: 4 bytes

ATM Layer Service

Service type & parameters associated with every VC:

- CBR (Constant Bit Rate): i.e. a circuit
  - Peak Cell Rate (PCR) = 1 / T
- VBR (Variable Bit Rate): some resource guarantee
  - PCR, Sustainable Cell Rate (SCR), Max Burst Size (MBS)
  - 2 leaky buckets: small peak serviced at PCR, then larger bucket serviced at SCR
  - VBRrt: (video, silence suppressed voice) as above with Cell Delay Variation Tolerance
- ABR (Available Bit Rate): managed network flow for “data” services
  - Minimum Cell Rate (MCR)
- UBR (Uncommitted Bit Rate): best effort

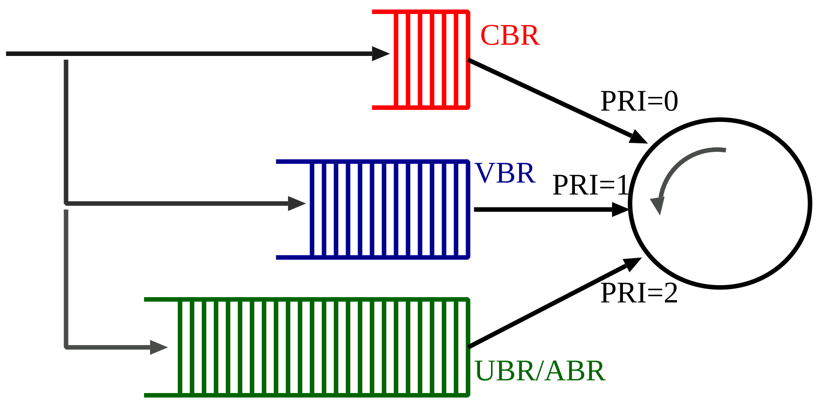


Resources and QoS

- End systems negotiate a contract with network
  - E.g. VBR: (PCR, SCR, MBS)
  - Possibly CDVT and Cell Loss Rate
- End systems perform *Traffic Shaping*
  - pace cell transmission
  - buffer traffic until admitted by scheduler
  - (may allow transmission with CLP=1)
- Network performs *Traffic Policing* at ingress
  - clumping and spreading may occur in network due to queuing

  $\Rightarrow$ may cause contract to be violated
    - Only police at edge
- Switches perform scheduling between cells
  - multiple FIFO queues with priority
  - per VC/VP queueing with weighted round-robin
  - (implement cell loss priority)

Switch Buffering Strategy Example



1. allow constant bit rate traffic to synchronise to the frame structure used by scheduling algorithms (tiny)

2. smooth out delay sensitive variable bit rate (VBR) traffic which may instantaneously exceed available bandwidth (medium)

3. delay insensitive traffic which has no guaranteed bandwidth (large)

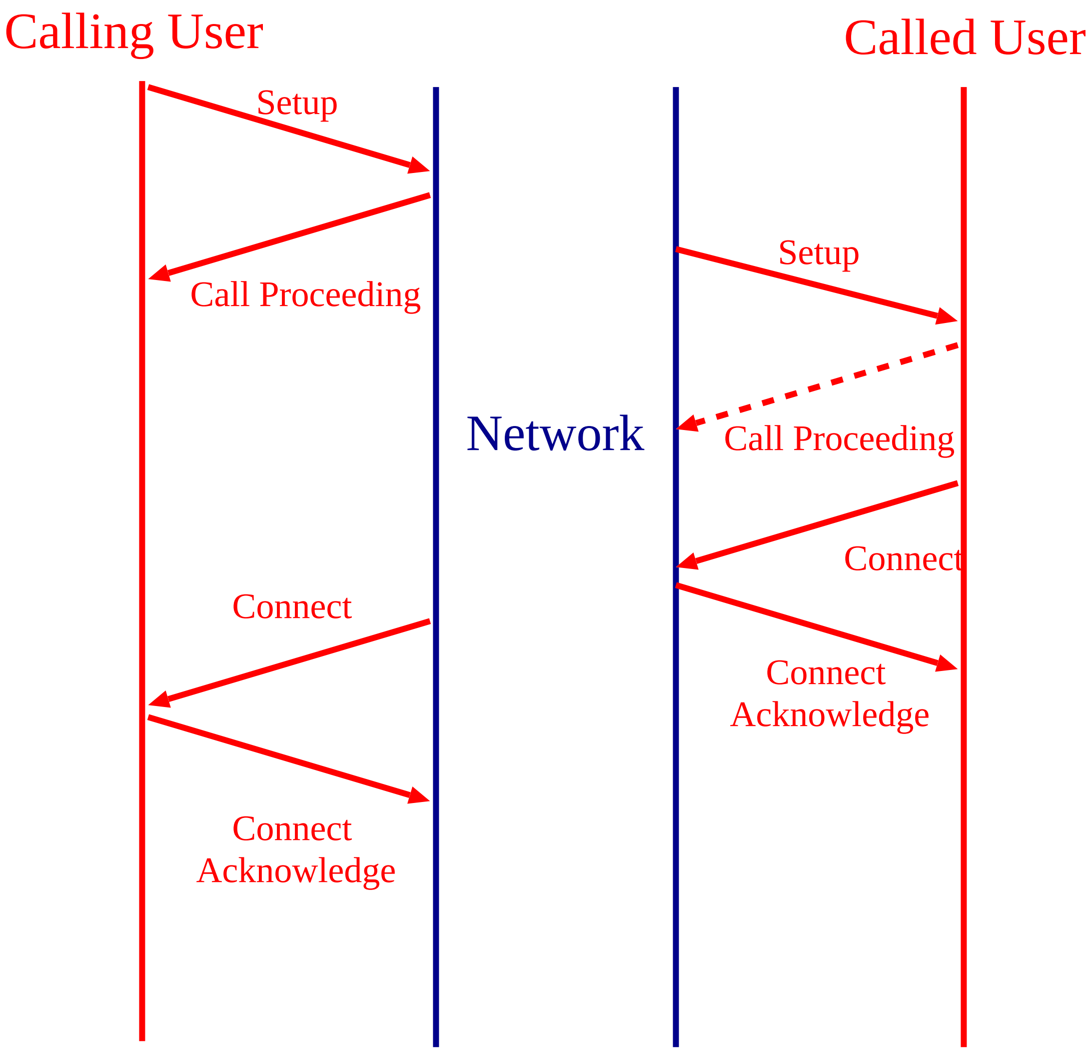
Call Admission Control

- may need to reject (renegotiate) some ‘calls’
- peak allocation would be pessimistic
- what level of over-commit is appropriate?
  - somewhere between peak and mean
  - depends on flow behaviours, correlations, amount of buffering in switches etc.
  - probabilistic guarantees
- two approaches:
  - analytic models of source behaviour
    - models would need to be complex
    - often hard to know parameters in advance
  - on-line measurement of *Effective Bandwidth*
    - measure behaviour of aggregate traffic over time
- over-provisioning makes allocation easier

⇒ maximising link utilization may not be the best strategy

Control and Management

- UNI and NNI signalling protocols
- Runs over reliable transport
- NNI does topology discovery, routeing (with support for hierarchies), signalling, addressing, maintenance etc.
- UNI: (call ref, remote SAP, AAL, QoS, ...)



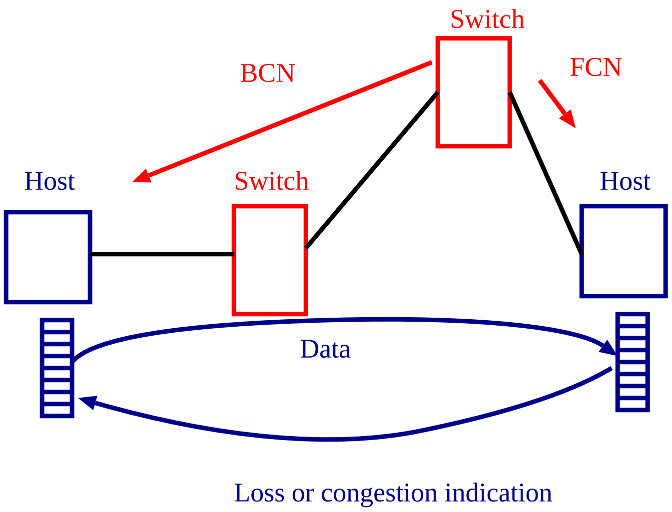
Congestion Control

- want to limit/eliminate loss and/or recover when it occurs

- Different techniques:

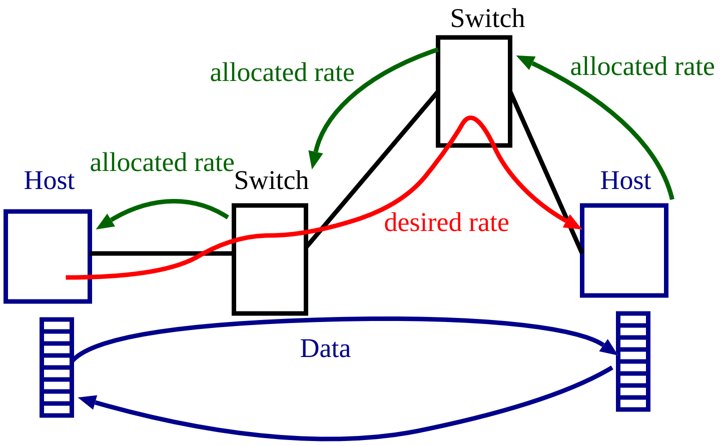
  1. schemes in which the network takes no active part (end to end congestion control)
     - TCP
  2. schemes in which the network ensures that information is only sent from one hop to the next if there is buffer available for it (hop-by-hop credit-based flow control)
     - excellent for System Area Networks
     - harder when b/w delay product large
     - don't wish to give flows more buffer credits than they actually need (depends on rate achieved)
     - how to signal credits, deal with lost credits etc.
  3. schemes in which the network gives an indication of its state to end systems
     - congestion notification
     - explicit rate control

Network Congestion Indication



- For ABR VCs, one in every 32 cells is a Forward Resource Management (FRM) cell
  - no user data
- QoS specified by Min & Peak Cell Rates
- Congestion Indication (CI) & No Increase (NI) bits
- end systems implement:
  if $CI = 1$ **then** $rate = rate * (1 - \alpha)$
  elseif not $NI = 1$ **then** $rate = rate + PCR * \alpha$
- (end systems clip on MCR and PCR)
- end systems reflect FRM into Backwards RM
- switches can also set bits in passing BRMs, or even generate BRMs directly

Rate Based Flow Control



Rate Based Flow Control

- RM cells also contain Explicit Rate field
- switches modify rate (downwards) in passing FRMs or BRMs
  - ER determined by bottleneck
- End system reflects ER in BRM
- End systems always uses min of ER and CI/NI calculated rate
- enables faster convergence on correct rate

Discard Control

- random discard of cells can be catastrophic compared to random discard of packets

- each cell discarded may destroy an entire packet,

- better to discard all cells from same packet

- essentially three choices of “intelligent cell discard”

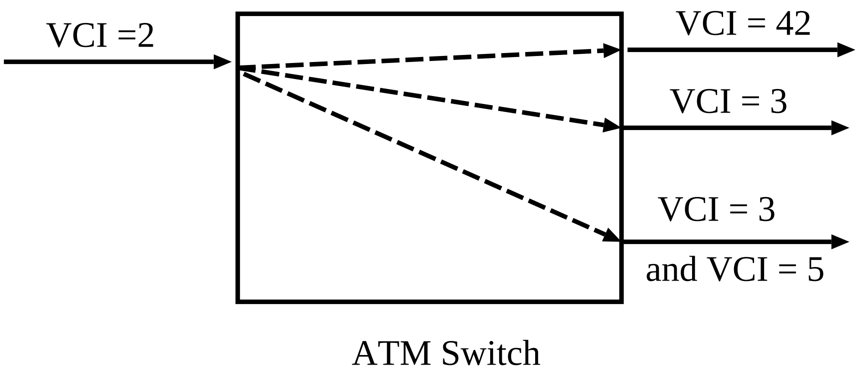
  1. when a cell is discarded discard all cells on the same VCI until one with the ATM User-User bit in the header is set (assumes AAL5 and discards ends of packets — on average half a packet)

  2. before you have to discard a packet find a cell with the AUU bit set and discard the cells on the same VCI which follow up to but not including the next one with the AUU bit set. (assumes AAL5 but discards entire packets)

  3. randomly select a VCI and drop a large quantity of cells from that VCI (does not assume AAL5)

- some original proposals for the ATM User-User bit were to use it also as a ATM layer indicator of “ATM user recovery”

Multicast



- ATM switch makes multiple copies of cells and maps to different outgoing VC's
- "copy fabric" either in front of or part of switching fabric
- conferencing, video distribution etc.
- many-to-one multicast not possible for AAL-5 VC's, though potentially possible with AAL-3/4

Time-scales and Actions

Call	
	call acceptance (end system and network) call renegotiation (end system and network)
	renegotiation if burst greater than round trip delay (end system and network)
Burst	
	smoothing (network $\rightarrow$ policing) (end systems $\rightarrow$ traffic shaping)
	flow control (network – network or user – network)
	buffering with queueing discipline (network or end system)
Cell	deterministic scheduling for constant bit rate

Integrated Broadband Network

- ATM as the Grand Unifying technology
  - core, LAN, desktop, home area, wireless
- Only really a commercial success in the core
  - Provides ISPs with QoS management facilities for their IP traffic
- Native ATM required new APIs for end-systems
  - expressing QoS
  - different SAPs
- Carrying IP over ATM
  - LANE, CLIP, ATMARP
  - VC per flow, or per src/dst pair ?
- Early switches (and networks):
  - had small buffers
  - weak VBR support
  - no ABR support
  - no Discard Control
  - lack of signalling
    * when arrived, heavyweight and complex
- Many SANs use Async Transfer Mode...

ATM Reality

- Small, fixed size cells
  - ✔ low access jitter on low bit rates
  - ✘ requires SAR hardware in NICs
  - ✘ arbitration frequency very high in core
    - ∗ fixed size helps switch fabric scheduling
- Virtual Circuits
  - ✔ simple switch routeing decisions
    - ∗ SRAM read vs longest prefix match
  - ✔ gives operator control over routeing
  - ✘ requires pre-signalling, “hard state”
    - ∗ fine for aggregates, pain for individual connections
  - ✘ “hard state” in switches
- Traffic shaping, class based queuing
  - ✔ very useful for network management
  - ✔ resource guarantees needed for live media
    - ∗ over-provisioning not always possible/economic
  - ✘ requires extra h/w and state in switches
- ABR
  - ✔ complex, but probably a good thing