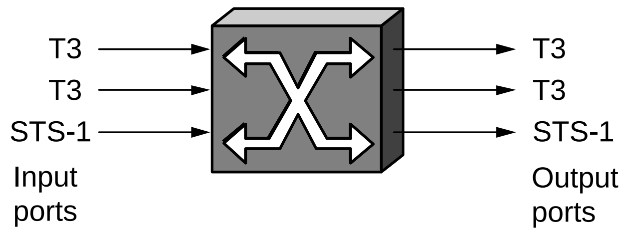
Switching

Outline

- Datagram Switching
- Virtual Circuit Switching
- Circuit Switching
- Asynchronous Transfer Mode (ATM)

Scalable Networks

- Switch
  - Forwards frames from input port to output port
  - Port selected based on address in frame header



- Advantages
  - Cover large geographic area (tolerate latency)
  - Support large numbers of hosts (scalable bandwidth)

Forwarding at each switch

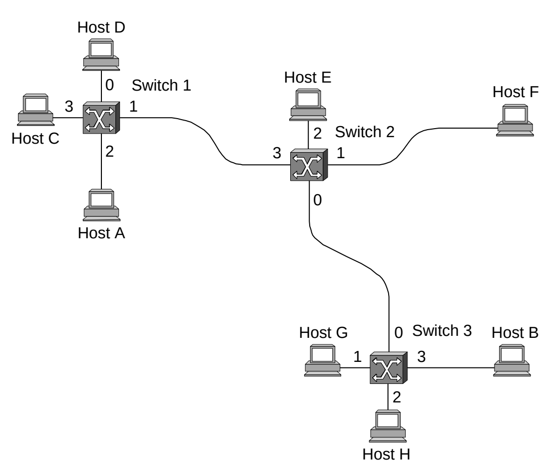
- Datagram (e.g., LAN Switches, IP)
  - Based on complete destination address within the packet. Any valid destination must be forwarded correctly.
- Virtual Circuits (e.g., MPLS, ATM, Frame Relay)
  - Based only on a label with the packet header. Only packets whose "virtual circuit" has been set up ahead of time must be forwarded correctly.
- Circuits (not packets)
  - Based implicitly on either time slot or wavelength. No forwarding information needed in data. Only those circuits whose path has been set up ahead of time must be forwarded correctly.

Datagram Switching

- No connection setup phase
- Each frame forwarded independently
- Sometimes called *connectionless* model

- Analogy: postal system

- Each switch maintains a forwarding (switching) table

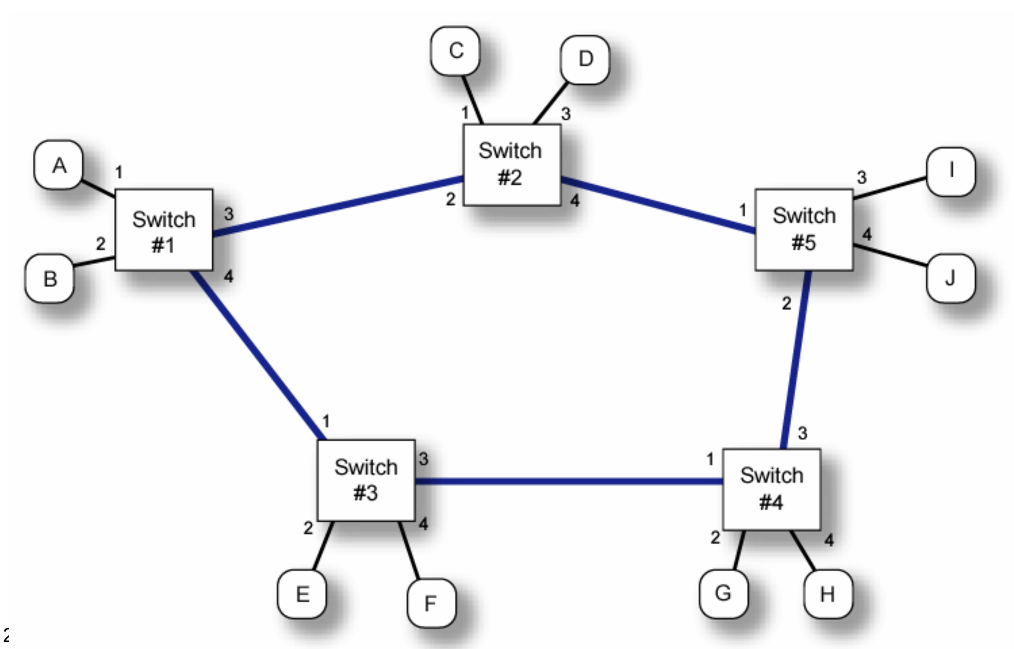


Datagram Model

- There is no round trip delay waiting for connection setup; a host can send data as soon as it is ready.
- Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
- Since packets are treated independently, it is possible to route around link and node failures.
- Since every packet must carry the full address of the destination, the overhead per packet is higher than for the connection-oriented model.

Example Network

- Switches 1-5, Hosts A-J



Datagram Forwarding Example

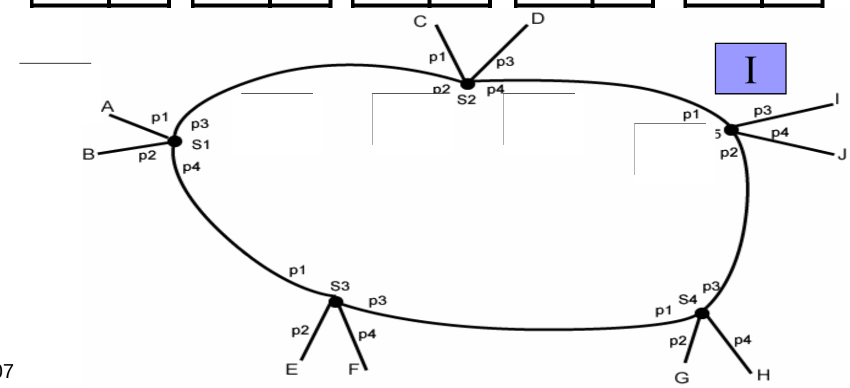
Switch #1	
Dest	Port
A	1
B	2
C	3
D	3
E	4
F	4
G	4
H	4
I	3
J	3

Switch #2	
Dest	Port
A	2
B	2
C	1
D	3
E	2
F	2
G	4
H	4
I	4
J	4

Switch #3	
Dest	Port
A	1
B	1
C	1
D	1
E	2
F	4
G	3
H	3
I	3
J	3

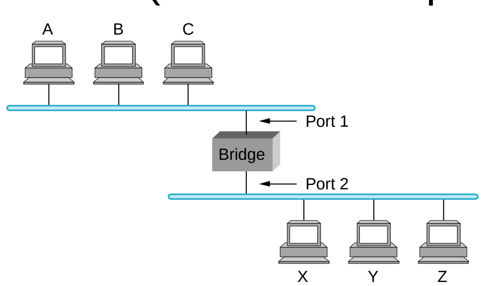
Switch #4	
Dest	Port
A	1
B	1
C	3
D	3
E	1
F	1
G	2
H	4
I	3
J	3

Switch #5	
Dest	Port
A	1
B	1
C	1
D	1
E	2
F	2
G	2
H	2
I	3
J	4



Bridges and Extended LANs

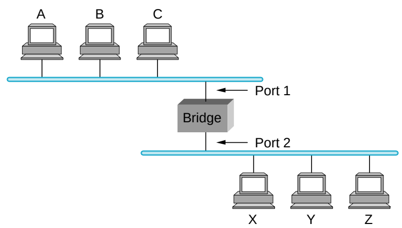
- LANs have physical limitations (e.g., 2500m)
- Connect two or more LANs with a *bridge*
  - accept and forward strategy
  - level 2 connection (does not add packet header)



- Ethernet Switch = Bridge on Steroids

Learning Bridges

- Do not forward when unnecessary
- Maintain forwarding table

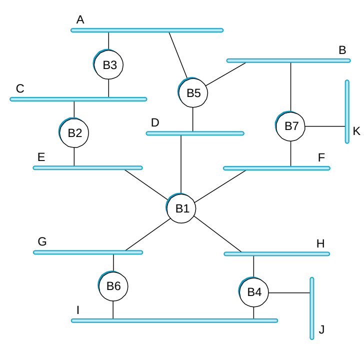


Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

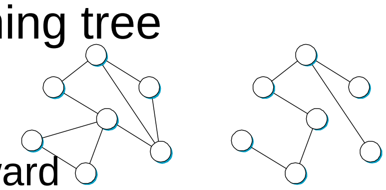
- Learn table entries based on source address
- Table is an optimization; need not be complete
- Always forward broadcast frames

Spanning Tree Algorithm

- Problem: loops



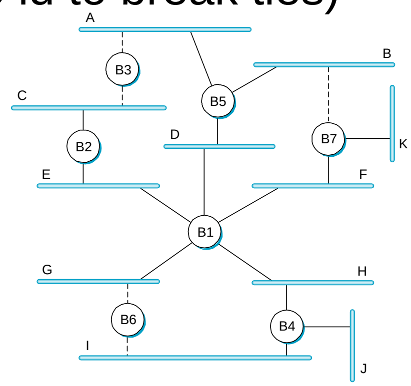
- Bridges run a distributed spanning tree algorithm
  - select which bridges actively forward
  - developed by Radia Perlman
  - now IEEE 802.1d specification



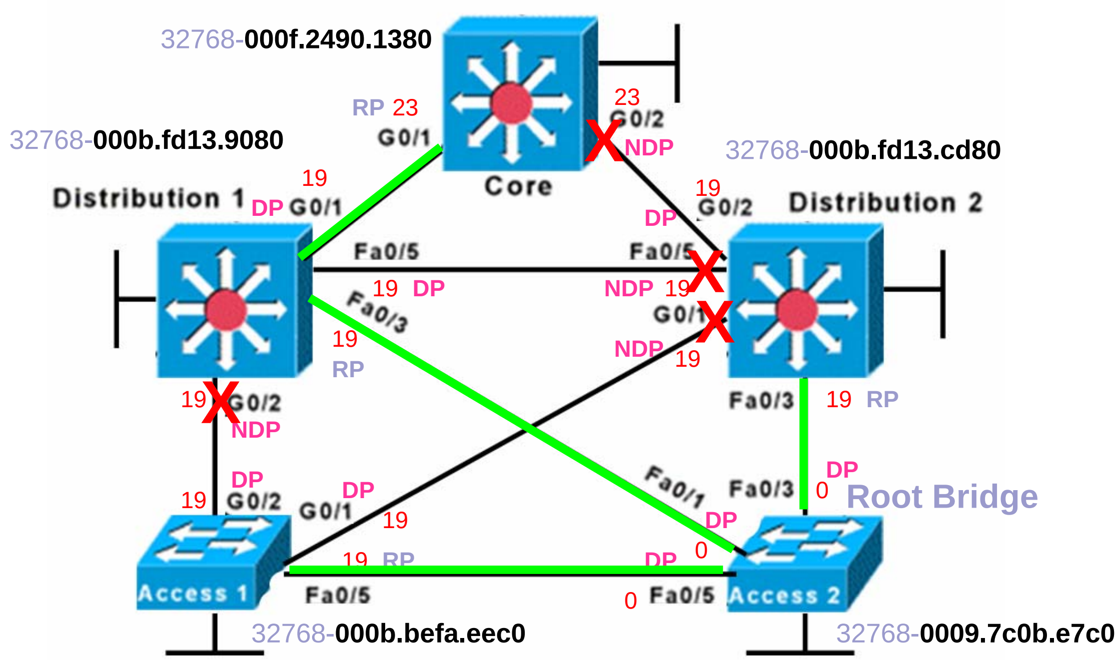
(a) (b)

Algorithm Overview

- Each bridge has unique id (e.g., B1, B2, B3)
- Select bridge with smallest id as root
- Select bridge on each LAN closest to root as designated bridge (use id to break ties)
- Each bridge forwards frames over each LAN for which it is the designated bridge



Spanning-Tree



Algorithm Details

- Bridges exchange configuration messages
  - id for bridge sending the message
  - id for what the sending bridge believes to be root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current best configuration message for each port
- Initially, each bridge believes it is the root

Algorithm Detail (cont)

- When learn not root, stop generating config messages
  - in steady state, only root generates configuration messages
- When learn not designated bridge, stop forwarding config messages
  - in steady state, only designated bridges forward config messages
- Root continues to periodically send config messages
- If any bridge does not receive config message after a period of time, it starts generating config messages claiming to be the root

How STP works?

1. Elect a single bridge to be the *Root* (smallest ID)
2. Calculate shortest path from themselves to the *Root* bridge
3. Elect a *Designated* Bridge for each LAN as the one closest to the *Root* bridge
4. For each bridge choose the *Root* port (shortest path to *Root* bridge)
5. Select the ports to be *Designated*
6. All other ports are backup/alternate (not forwarding state)
7. Each bridge initially assumes to be *Root* (*Root Path Cost* = 0)
8. Under normal circumstances the algorithm stabilizes and the chosen *Root* transmit *CM BPDUs* every *Hello Time*

Broadcast and Multicast

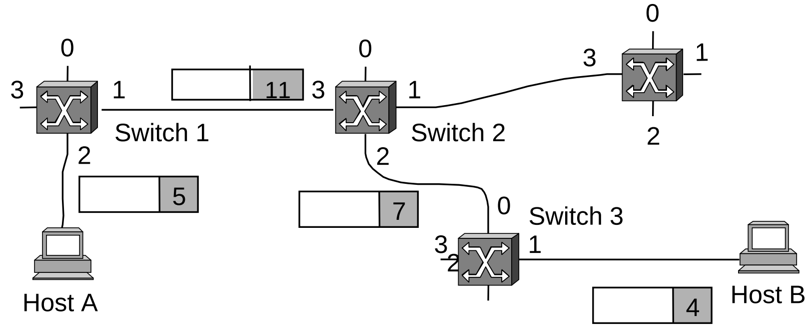
- Forward all broadcast/multicast frames
  - current practice
- Learn when no group members downstream
- Accomplished by having each member of group G send a frame to bridge multicast address with G in source field

Limitations of Bridges

- Do not scale
  - spanning tree algorithm does not scale
  - broadcast does not scale
- Do not accommodate heterogeneity

Virtual Circuit Switching

- Explicit connection setup (and tear-down) phase
- Subsequence frames follow same circuit
- Sometimes called *connection-oriented* model
- Analogy: phone call
- Each switch maintains a VC table



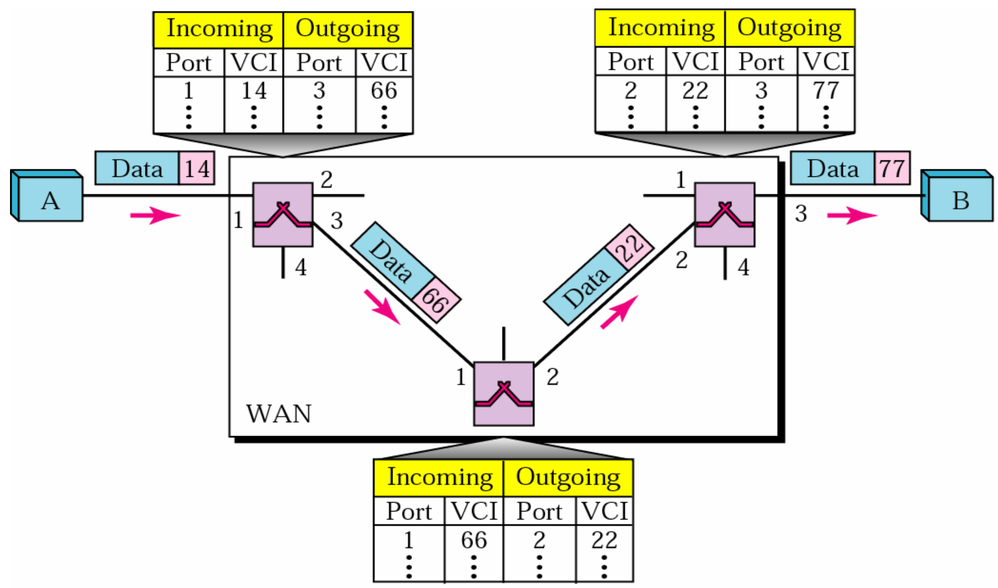
Virtual Circuit Forwarding

- Packets are forwarded based on a label in the header
- Labels are not destination addresses, usually much shorter
- Labels need to be unique on a link but not in a network, i.e., we can reuse labels on each link.
- Switch forwarding tables consist of a map between (input port, packet label) to (output port, new packet label)
- Table entry for each virtual circuit rather than for each destination (the datagram case)
- Technologies: MPLS, Frame Relay, ATM, X.25

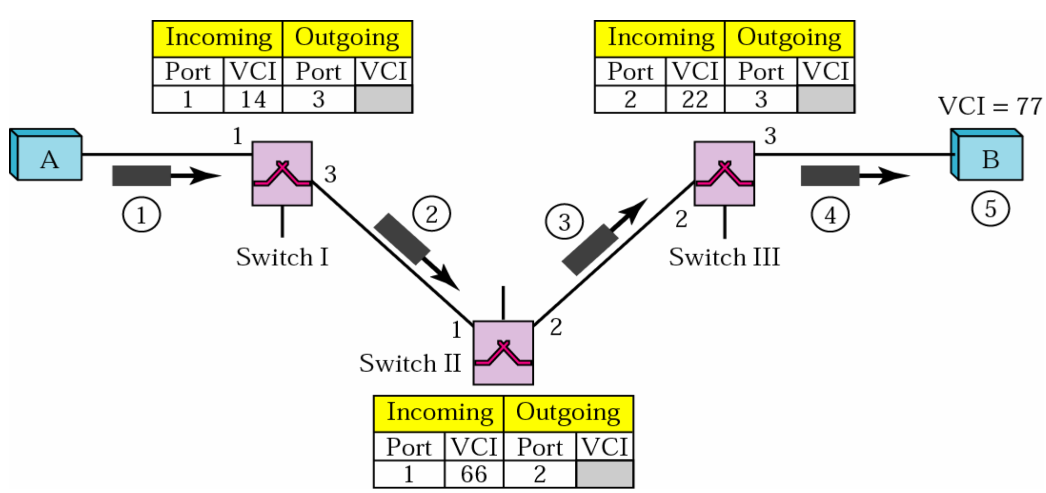
Virtual Circuit Model

- Typically wait full RTT for connection setup before sending first data packet.
- While the connection request contains the full address for destination, each data packet contains only a small identifier, making the per-packet header overhead small.
- If a switch or a link in a connection fails, the connection is broken and a new one needs to be established.
- Connection setup provides an opportunity to reserve resources.

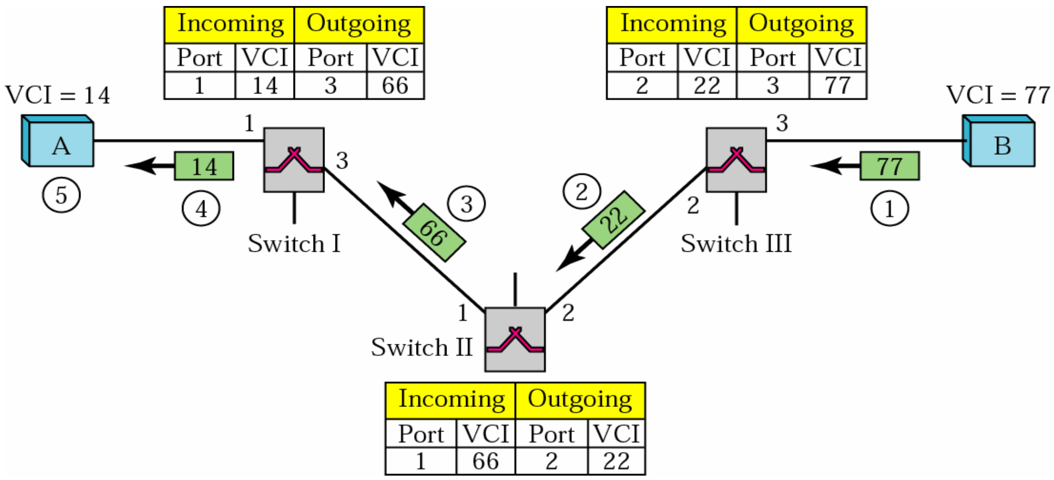
Source-to-destination data transfer



SVC setup request



SVC setup acknowledgment



Incoming		Outgoing	
Port	VCI	Port	VCI
1	14	3	66

Incoming		Outgoing	
Port	VCI	Port	VCI
2	22	3	77

Incoming		Outgoing	
Port	VCI	Port	VCI
1	66	2	22

VC Forwarding Table Example

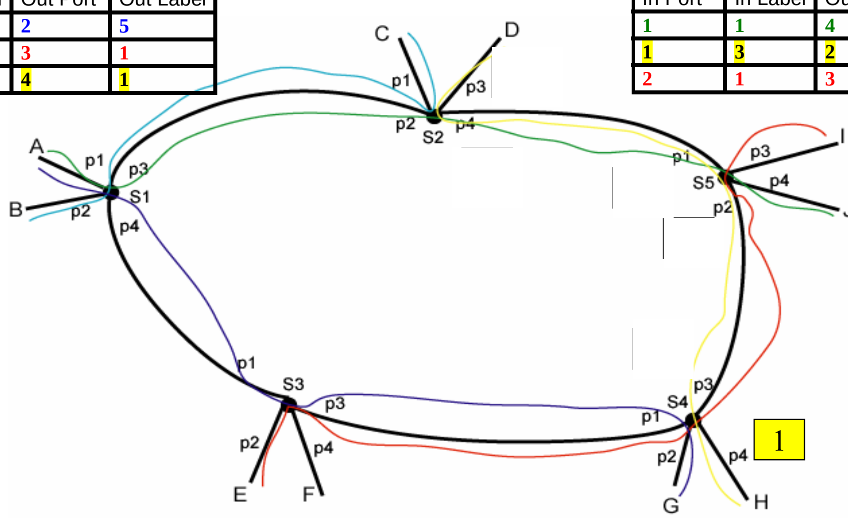
Switch #1			
In Port	In Label	Out Port	Out Label
1	2	3	5
2	1	3	1
1	1	4	1

Switch #2			
In Port	In Label	Out Port	Out Label
2	5	4	1
2	1	1	1
3	6	4	3

Switch #3			
In Port	In Label	Out Port	Out Label
1	1	3	3
2	1	3	1

Switch #4			
In Port	In Label	Out Port	Out Label
1	3	2	5
1	1	3	1
3	1	4	1

Switch #5			
In Port	In Label	Out Port	Out Label
1	1	4	2
1	3	2	1
2	1	3	1



“Real” Circuit Forwarding

- No more packets
- Bit streams are distinguished by port and
  - Time slots in the TDM case
  - Wavelength in the WDM case
  - Frequency in the FDM case
- Switching independent of bit stream contents
- TDM example (same connections as VC case)
  - Host A to Host J, Host B to Host C, Host E to Host I, Host D to Host H, and Host A to Host G

“Real” Circuit Tables Example

Switch #1

In Port	In Slot	Out Port	Out Slot
1	2	3	5
2	1	3	1
1	1	4	1

Switch #2

In Port	In Slot	Out Port	Out Slot
2	5	4	1
2	1	1	1
3	6	4	3

Switch #3

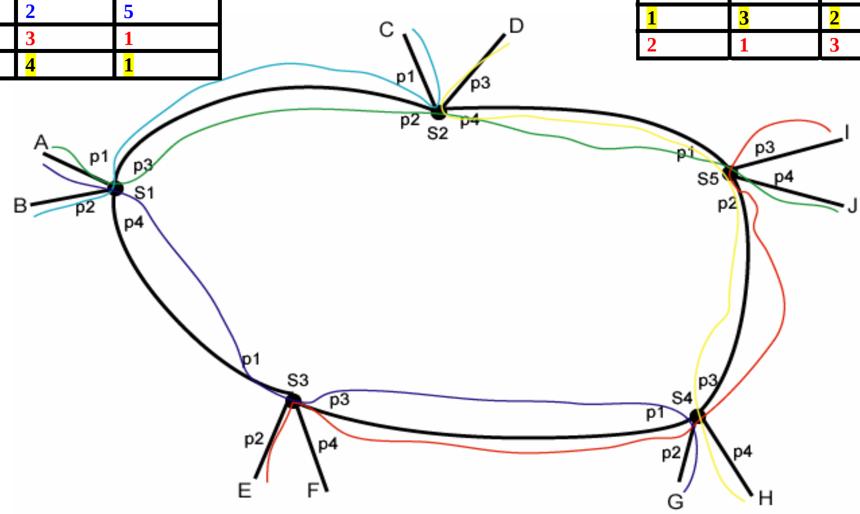
In Port	In Slot	Out Port	Out Slot
1	1	3	3
2	1	3	1

Switch #4

In Port	In Slot	Out Port	Out Slot
1	3	2	5
1	1	3	1
3	1	4	1

Switch #5

In Port	In Slot	Out Port	Out Slot
1	1	4	2
1	3	2	1
2	1	3	1



Real Circuits and Virtual Circuits

- Virtual Circuits
  - Packet based, label (not destination address) in packet header
  - Doesn't always consume bandwidth, i.e., traffic can be bursty
- Real Circuits
  - No packets; raw bit stream, implicit label with either time slot or wavelength
  - Is always consuming a fixed bandwidth, easy to keep track of bandwidth but not necessarily the most efficient utilization of link capacity.

QoS with Real Circuits

- Bandwidth
  - Hard bandwidth guarantees are given by default (even if you don't want them).
- Delay
  - Very little delay variation. Most delay attributable to propagation. Switching delays in most circuit switches is minimal.
- Bit Error Rate
  - Is the primary "signal quality measure"

QoS with Virtual Circuits

- Bandwidth
  - Is by default shared with other users. Effort required to make guarantees. Very good statistical multiplexing gain can be obtained.
- Delay
  - In addition to propagation and switch processing delay we now have queueing induced delays
  - Queueing delays: can be quite large, can be quite variable
  - By default no guarantees made
- Dropped/Errored Packets
  - Packets can be errored (bits errors), or dropped due to buffer overflows.

Cell Switching (ATM)

- Connection-oriented packet-switched network
- Used in both WAN and LAN settings
- Signaling (connection setup) Protocol: Q.2931
- Specified by ATM forum
- Packets are called *cells*
  - 5-byte header + 48-byte payload
- Commonly transmitted over SONET
  - other physical layers possible

Variable vs. Fixed-Length Packets

- No Optimal Length
  - If small: high header-to-data overhead
  - If large: low utilization for small messages
- Fixed-Length Easier to Switch in Hardware
  - Simpler
  - Enables parallelism

Big vs. Small Packets

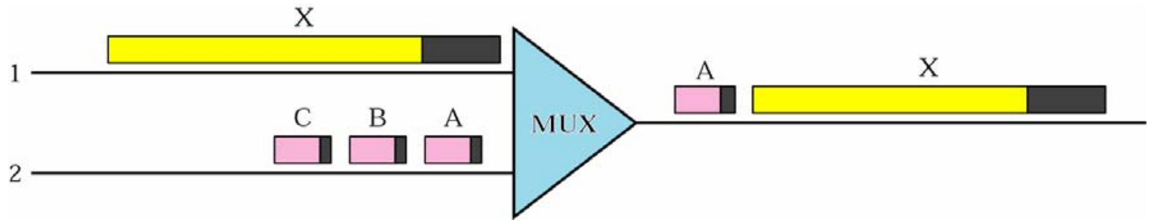
- Small Improves Queue behavior
  - finer-grained preemption point for scheduling link
    - maximum packet = 4KB
    - link speed = 100Mbps
    - transmission time = 4096 x 8/100 = 327.68us
    - high priority packet may sit in the queue 327.68us
    - in contrast, 53 x 8/100 = 4.24us for ATM
  - near cut-through behavior
    - two 4KB packets arrive at same time
    - link idle for 327.68us while both arrive
    - at end of 327.68us, still have 8KB to transmit
    - in contrast, can transmit first cell after 4.24us
    - at end of 327.68us, just over 4KB left in queue

Big vs. Small (cont.)

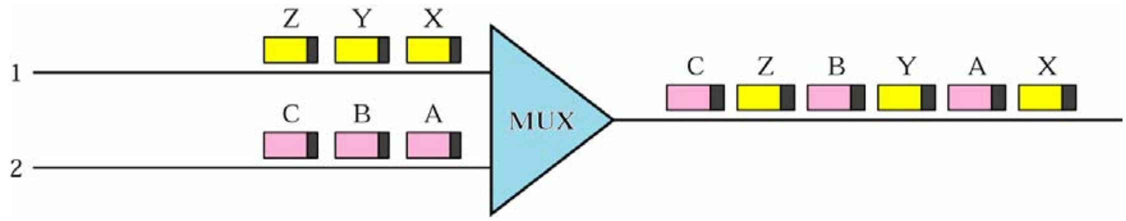
- Small Improves Latency (for voice)
  - Voice digitally encoded at 64Kbps (8-bit samples at 8KHz)
  - Need full cell's worth of samples before sending cell
  - Example: 1000-byte cells implies 125ms per cell (too long)
  - Smaller latency implies no need for echo cancellers

- ATM Compromise: 48 bytes = (32+64)/2

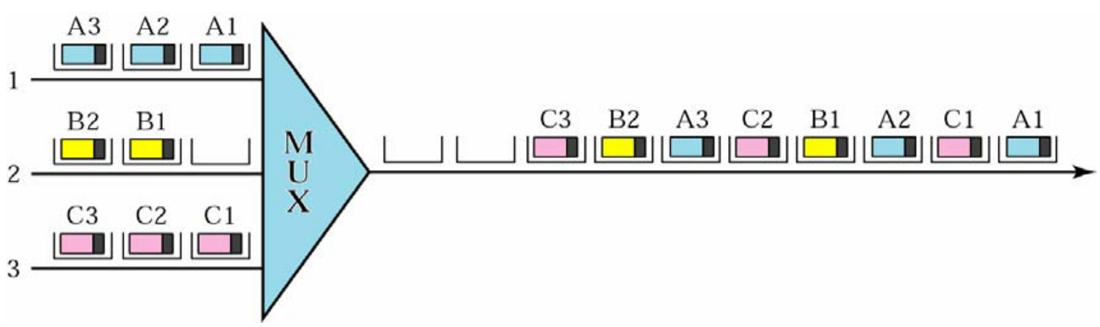
Multiplexing using different frame sizes



Multiplexing using cells



ATM multiplexing



Cell Format

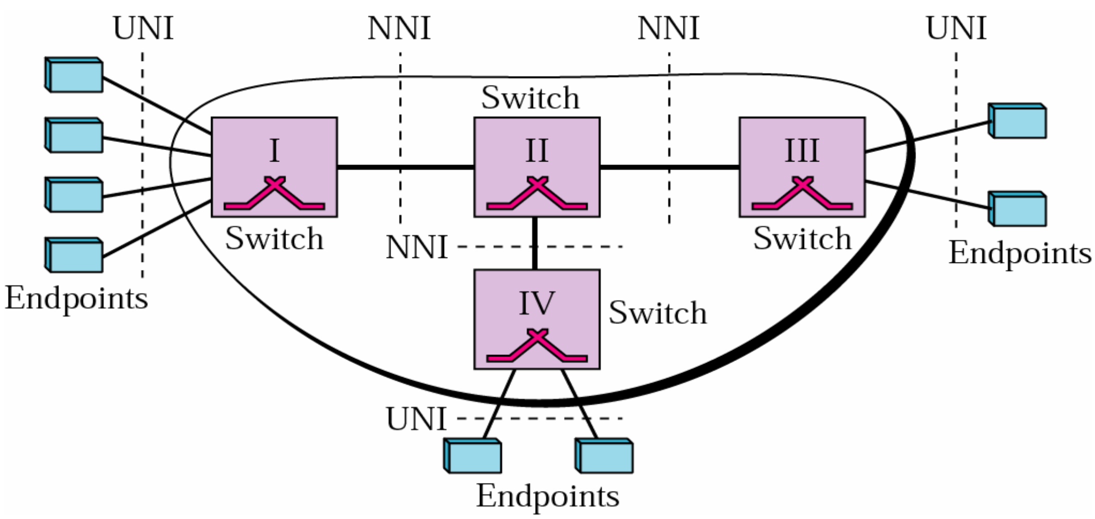
- User-Network Interface (UNI)

4	8	16	3	1	8	384 (48 bytes)
GFC	VPI	VCI	Type	CLP	HEC (CRC-8)	Payload

  - host-to-switch format
  - GFC: Generic Flow Control (still being defined)
  - VCI: Virtual Circuit Identifier
  - VPI: Virtual Path Identifier
  - Type: management, congestion control, AAL5 (later)
  - CLP: Cell Loss Priority
  - HEC: Header Error Check (CRC-8)

- Network-Network Interface (NNI)
  - switch-to-switch format
  - GFC becomes part of VPI field

Architecture of an ATM network



Connection identifiers

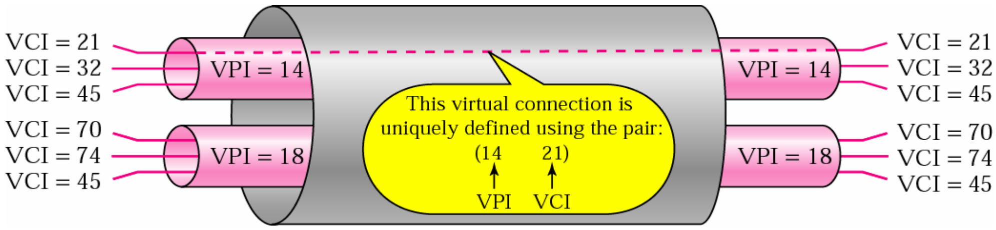
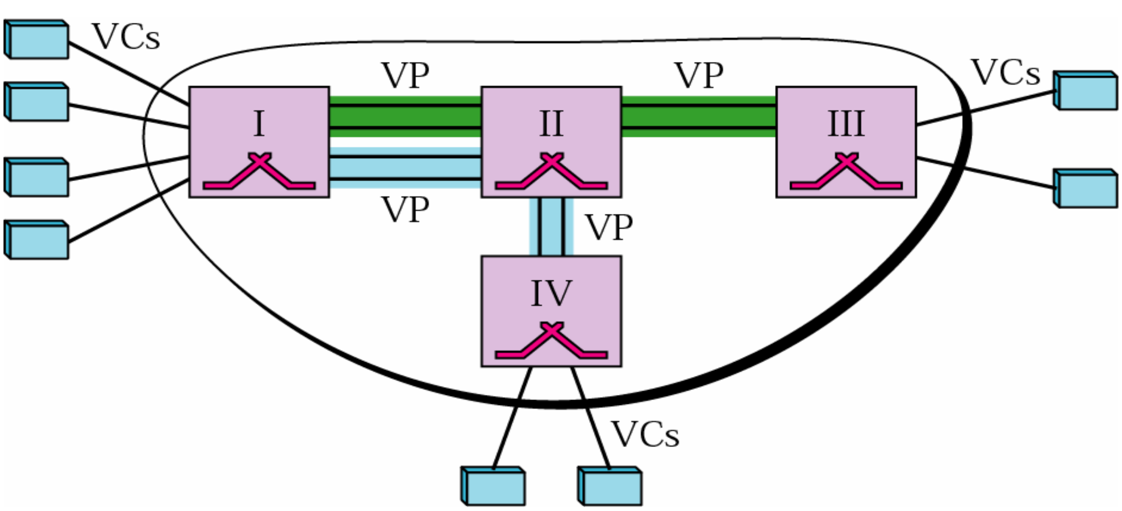
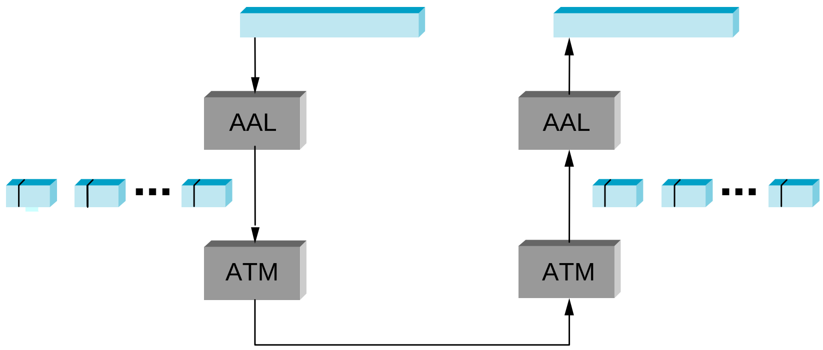


Figure 18.18 Example of VPs and VCs

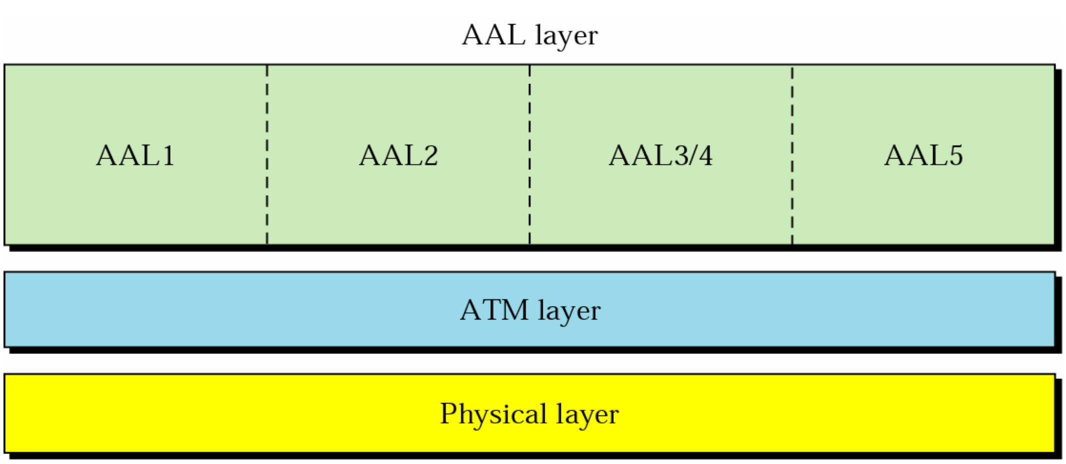


ATM Adaptation Layer (AAL)

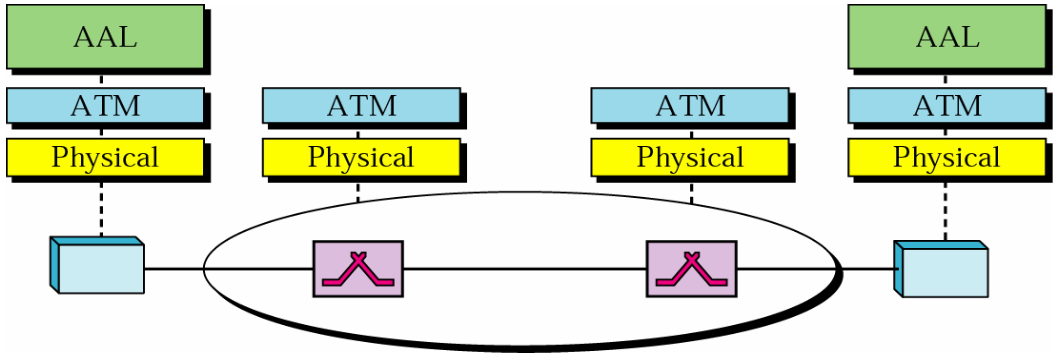
- AAL 1 and 2 designed for applications that need guaranteed rate (e.g., voice, video)
- AAL 3/4 designed for packet data
- AAL 5 is an alternative standard for packet data



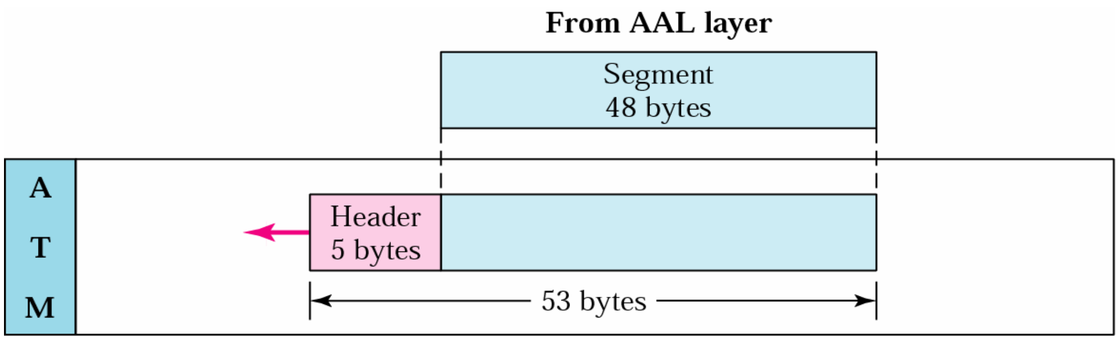
ATM layers



ATM layers in endpoint devices and switches



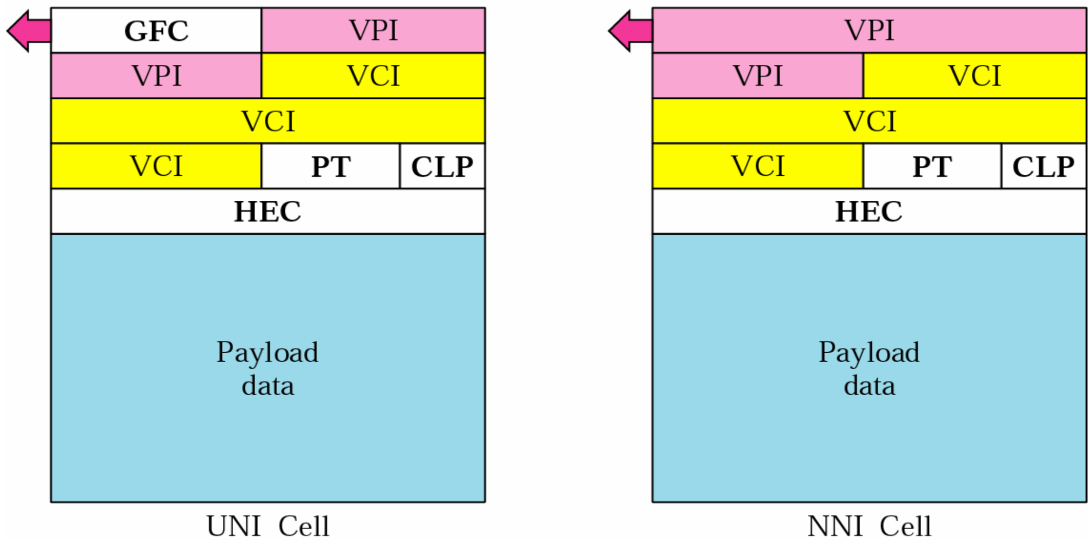
ATM layer



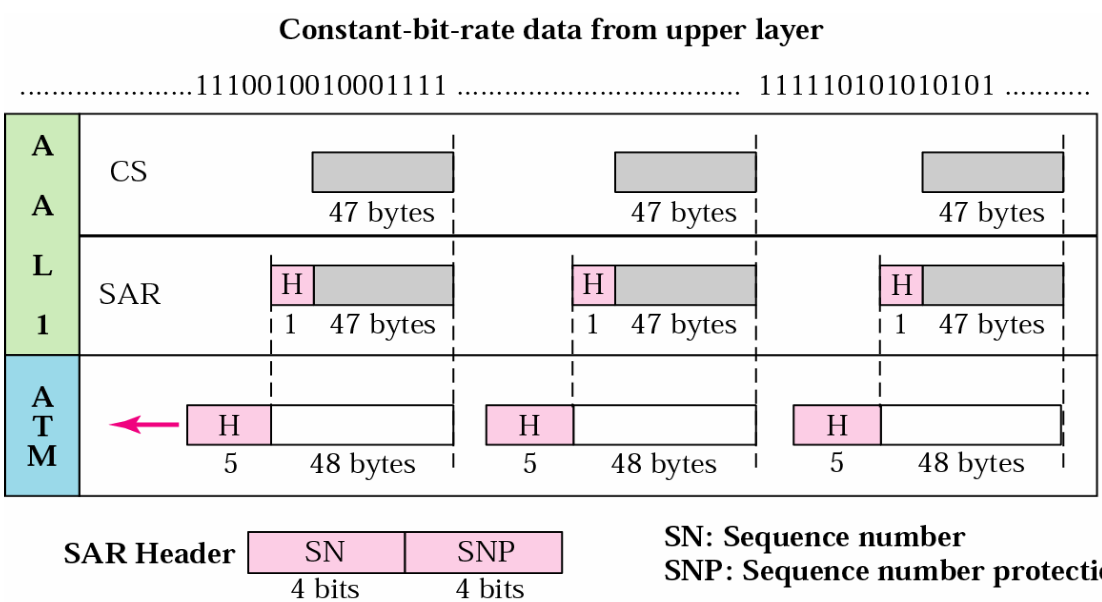
ATM headers

GFC: Generic flow control
VPI: Virtual path identifier
VCI: Virtual channel identifier

PT: Payload type
CLP: Cell loss priority
HEC: Header error control

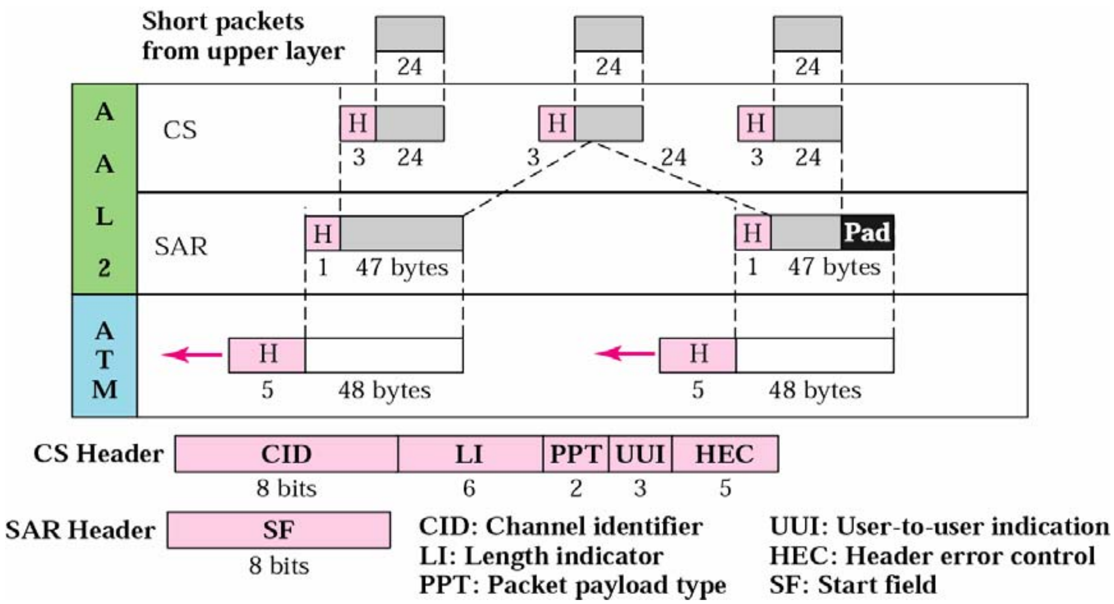


AAL1

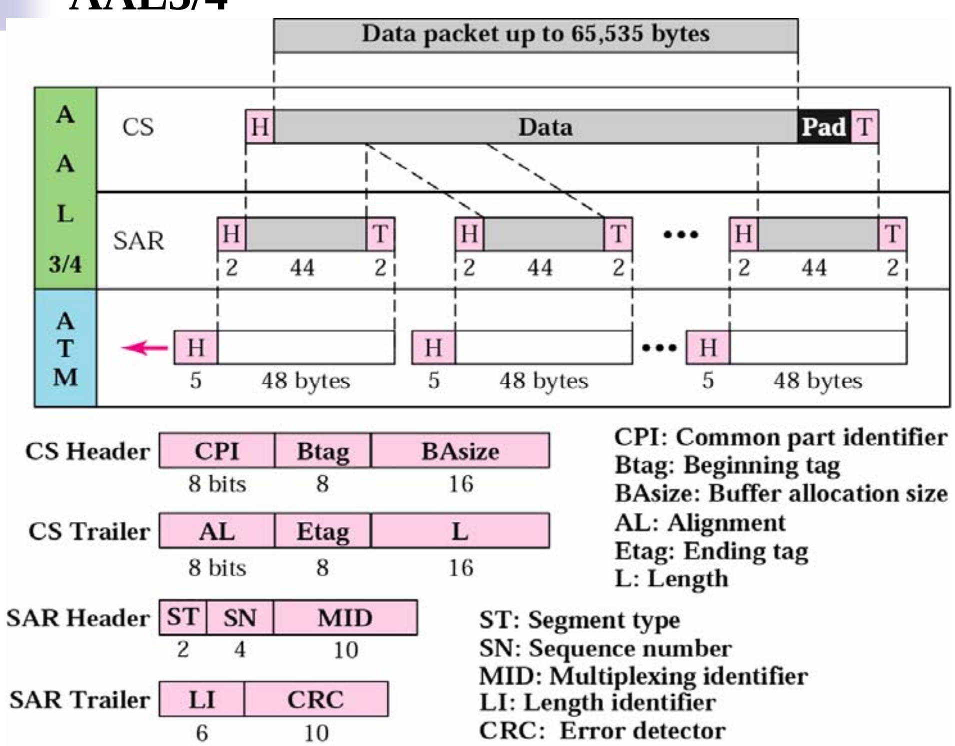


SN: Sequence number
SNP: Sequence number protection

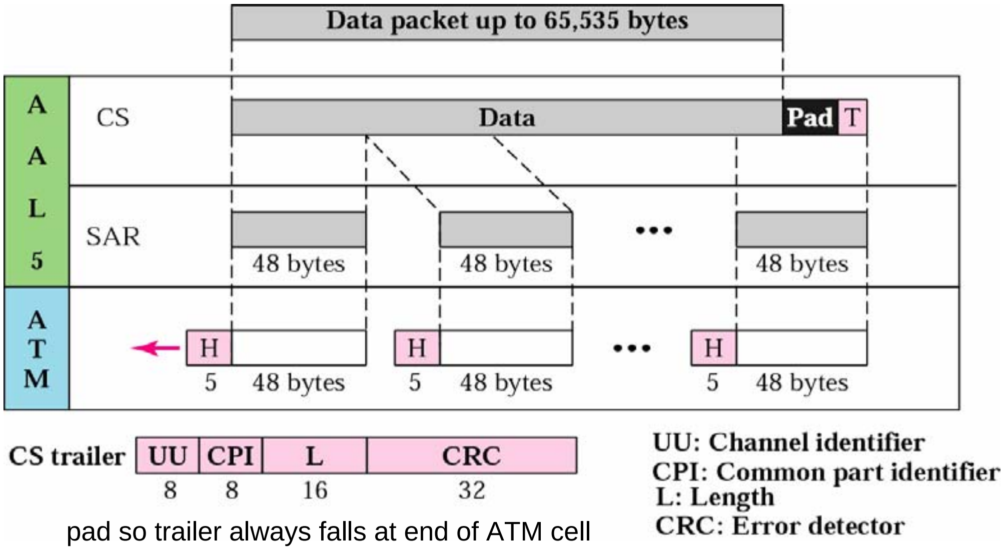
AAL2


Short packets from upper layer
24
24
24
A A L 2
CS
H
3 24
SAR
H Pad
1 47 bytes
ATM
H
5 48 bytes
CS Header
CID LI PPT UUI HEC
8 bits 6 2 3 5
SAR Header
SF
8 bits
CID: Channel identifier
LI: Length indicator
PPT: Packet payload type
UUI: User-to-user indication
HEC: Header error control
SF: Start field

AAL3/4


Data packet up to 65,535 bytes
A A L 3/4
CS
H Data Pad T
SAR
H T
2 44 2
ATM
H
5 48 bytes
CS Header
CPI Btag BAsize
8 bits 8 16
CS Trailer
AL Etag L
8 bits 8 16
SAR Header
ST SN MID
2 4 10
SAR Trailer
LI CRC
6 10
CPI: Common part identifier
Btag: Beginning tag
BAsize: Buffer allocation size
AL: Alignment
Etag: Ending tag
L: Length
ST: Segment type
SN: Sequence number
MID: Multiplexing identifier
LI: Length identifier
CRC: Error detector

AAL5



pad so trailer always falls at end of ATM cell
Length: size of PDU (data only)
CRC-32 (detects missing or misordered cells) Cell Format
end-of-PDU bit in Type field of ATM header

References

- *Data Communications and Networking*, 3rd ed**.**, Behrouz A. Forouzan, McGraw-Hill, 2004.
- GMPLS & Switching, Grotto Networking.
- Rick Graziani, “STP: Spanning Tree Protocol,” Cabrillo College, CCNA