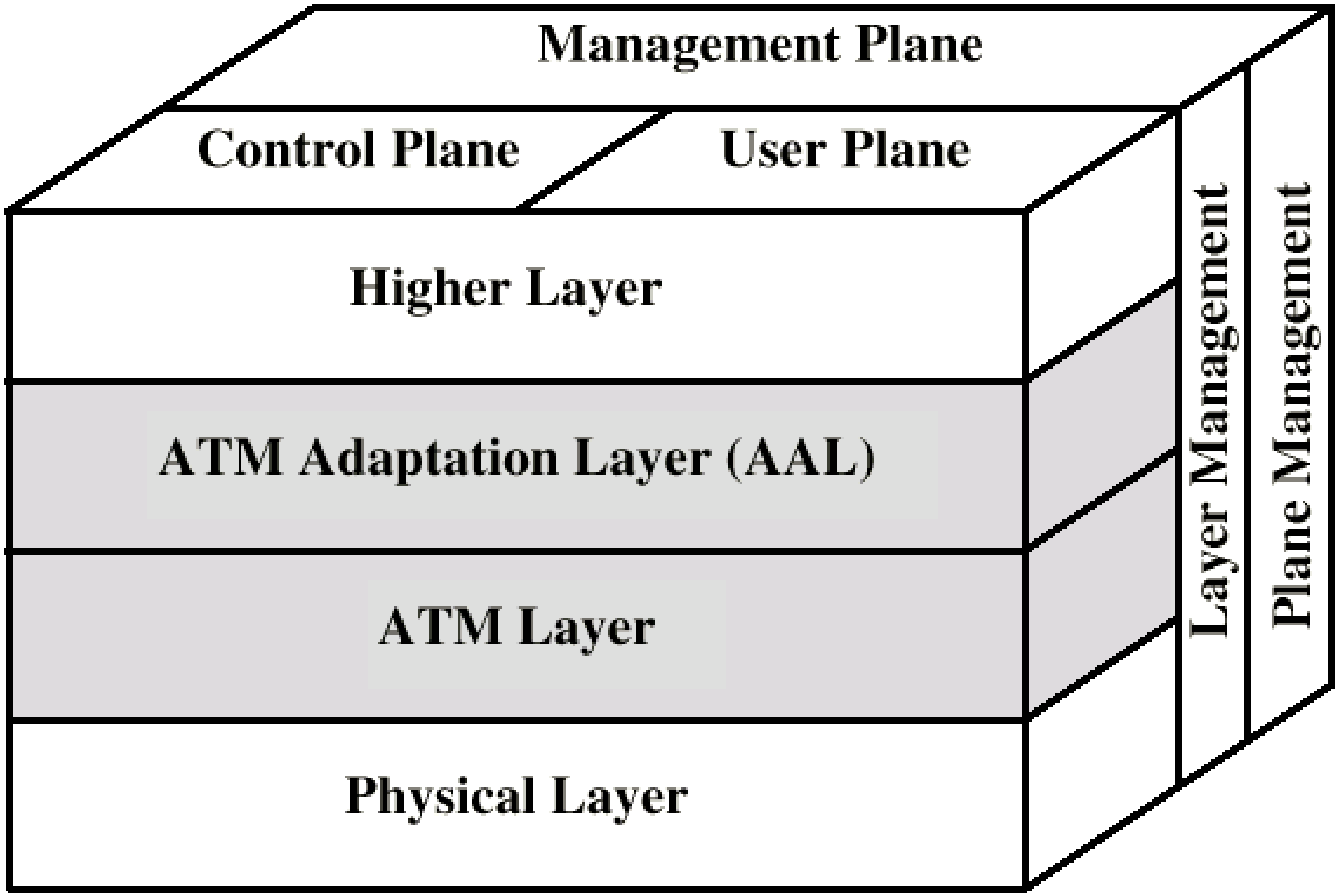
Module V Asynchronous Transfer Model, Routing on switched network.

Protocol Architecture

- Similarities between ATM and packet switching
  - Transfer of data in discrete chunks
  - Multiple logical connections over single physical interface
- In ATM flow on each logical connection is in fixed sized packets called cells
- Minimal error and flow control
  - Reduced overhead
- Data rates (physical layer) 25.6Mbps to 622.08Mbps

Protocol Architecture (diag)



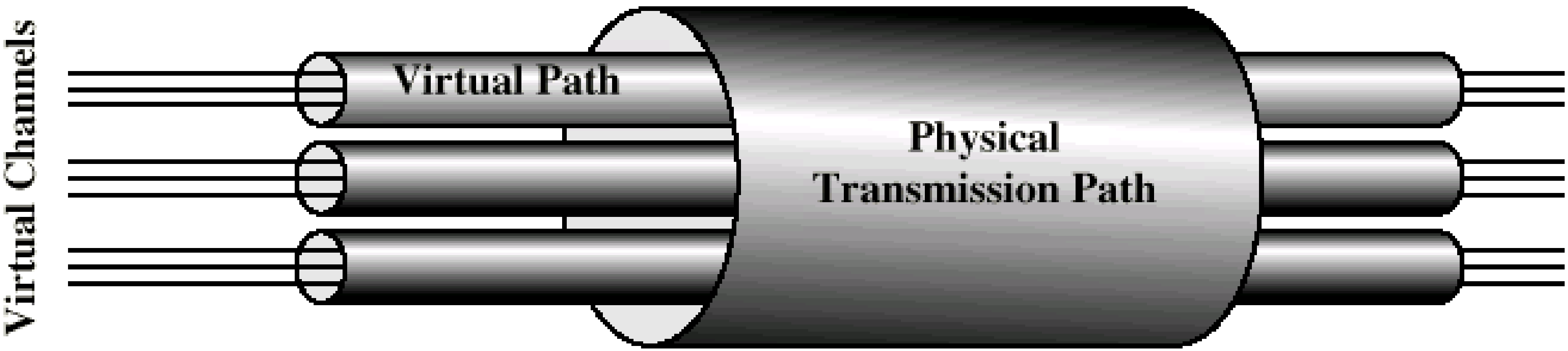
Reference Model Planes

- User plane
  - Provides for user information transfer
- Control plane
  - Call and connection control
- Management plane
  - Plane management
    - whole system functions
  - Layer management
    - Resources and parameters in protocol entities

ATM Logical Connections

- Virtual channel connections (VCC)
- Analogous to virtual circuit in X.25
- Basic unit of switching
- Between two end users
- Full duplex
- Fixed size cells
- Data, user-network exchange (control) and network-network exchange (network management and routing)
- Virtual path connection (VPC)
  - Bundle of VCC with same end points

ATM Connection Relationships



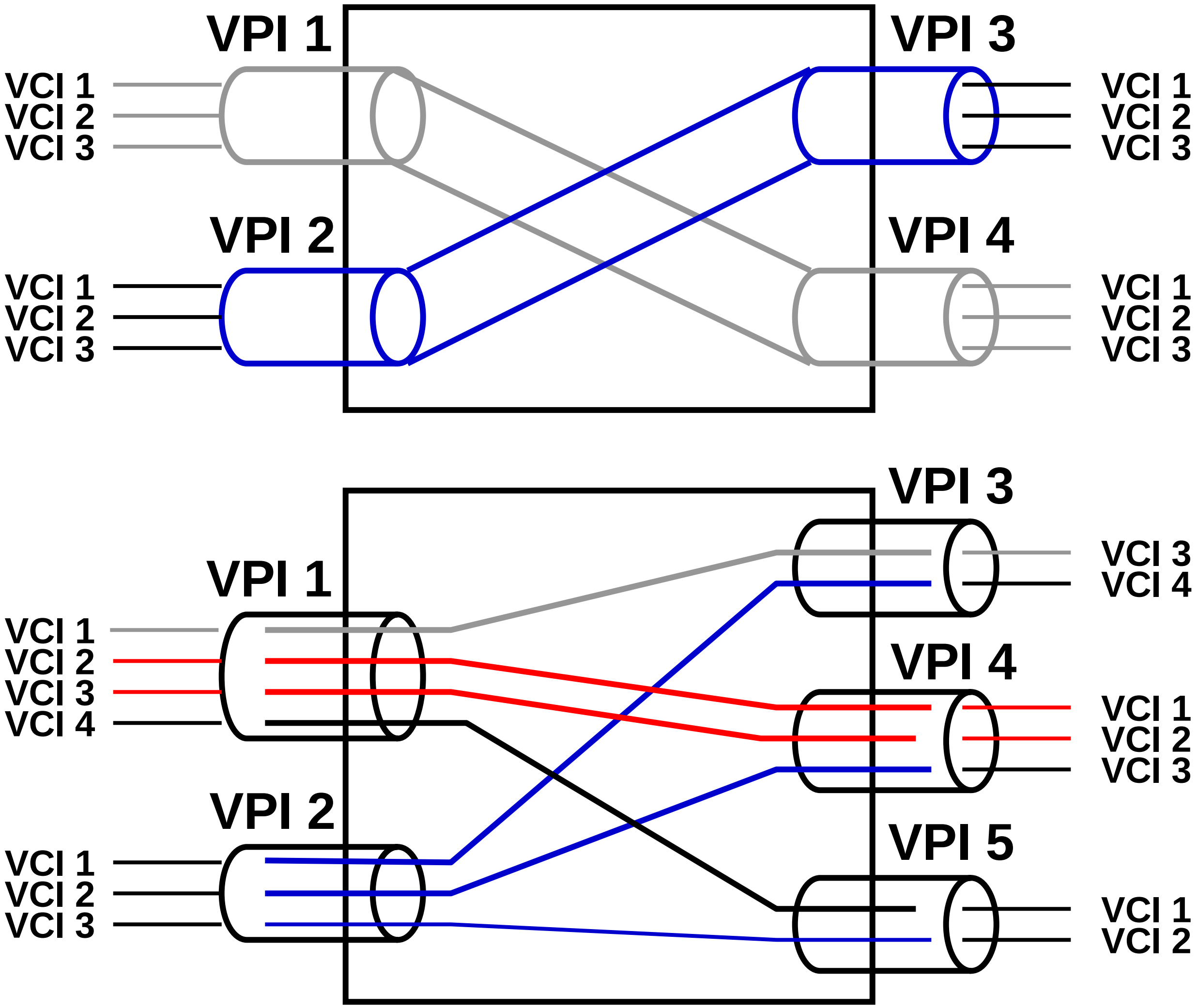
Virtual Channel Terminology

Virtual Channel (VC)	A generic term used to describe unidirectional transport of ATM cells associated by a common unique identifier value.
Virtual Channel Link	A means of unidirectional transport of ATM cells between a point where a VCI value is assigned and the point where that value is translated or terminated.
Virtual Channel Identifier (VCI)	A unique numerical tag that identifies a particular VC link for a given VPC.
Virtual Channel Connection (VCC)	A concatenation of VC links that extends between two points where ATM service users access the ATM layer. VCCs are provided for the purpose of user-user, user-network, or network-network information transfer. Cell sequence integrity is preserved for cells belonging to the same VCC.

Virtual Path Terminology

Virtual Path	A generic term used to describe unidirectional transport of ATM cells belonging to virtual channels that are associated by a common unique identifier value.
Virtual Path Link	A group of VC links, identified by a common value of VPI, between a point where a VPI value is assigned and the point where that value is translated or terminated.
Virtual Path Identifier (VPI)	Identifies a particular VP link.
Virtual Path Connection (VPC)	A concatenation of VP links that extends between the point where the ~~VCI~~ VPI values are assigned and the point where those values are translated or removed, i.e., extending the length of a bundle of VC links that share the same VPI. VPCs are provided for the purpose of user-user, user-network, or network-network information transfer.

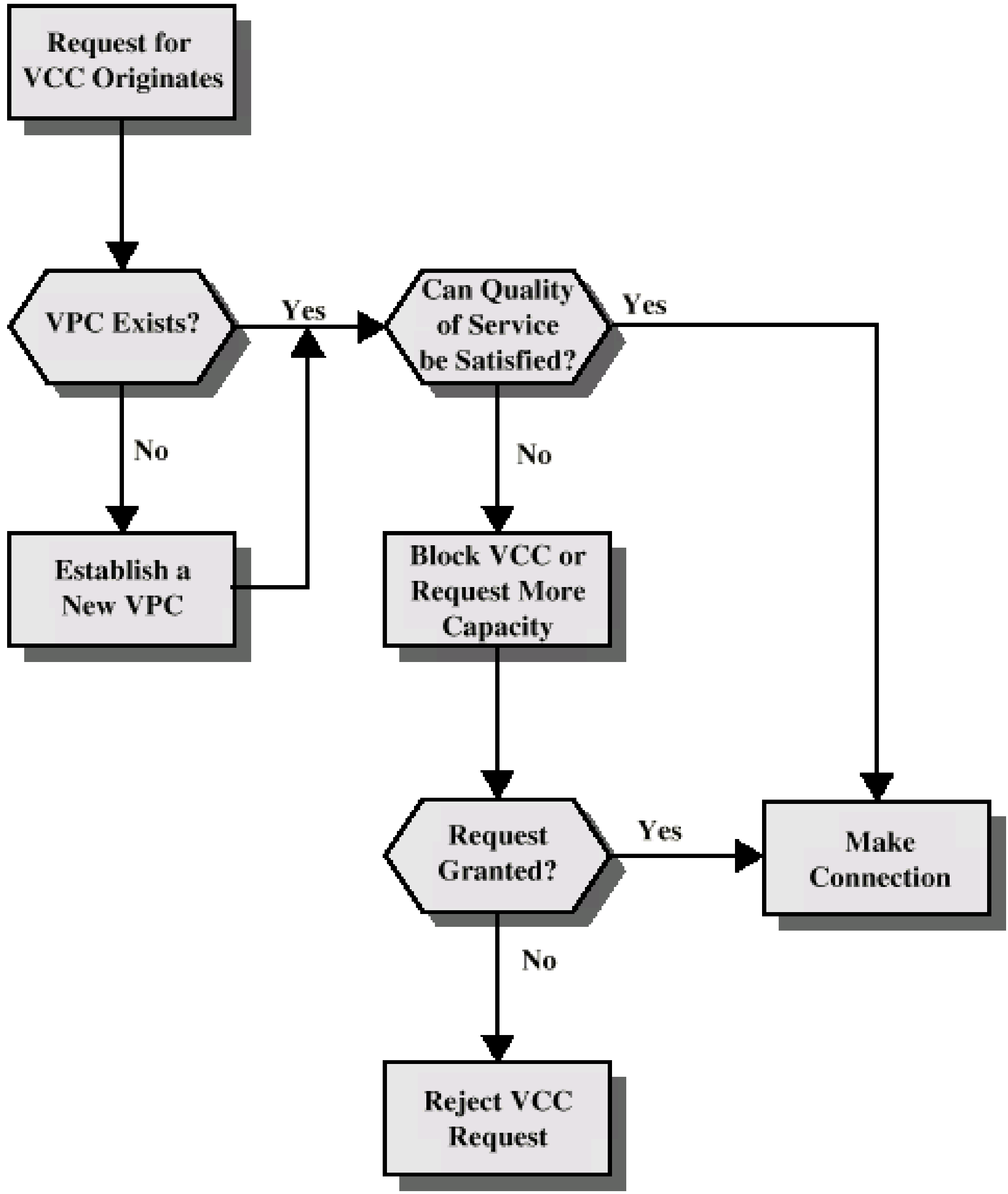
VP/VC Switching



Advantages of Virtual Paths

- Simplified network architecture
- Increased network performance and reliability
- Reduced processing
- Short connection setup time
- Enhanced network services

Call Establishment Using VPs



Virtual Channel Connection Uses

- Between end users
  - End to end user data
  - Control signals
  - VPC provides overall capacity
    - VCC organization done by users
- Between end user and network
  - Control signaling
- Between network entities
  - Network traffic management
  - Routing

VP/VC Characteristics

- Quality of service
- Switched and semi-permanent channel connections
- Call sequence integrity
- Traffic parameter negotiation and usage monitoring

- VPC only
  - Virtual channel identifier restriction within VPC

Control Signaling - VCC

- Done on separate connection
- Semi-permanent VCC
- Meta-signaling channel
  - Used as permanent control signal channel
- User to network signaling virtual channel
  - For control signaling
  - Used to set up VCCs to carry user data
- User to user signaling virtual channel
  - Within pre-established VPC
  - Used by two end users without network intervention to establish and release user to user VCC

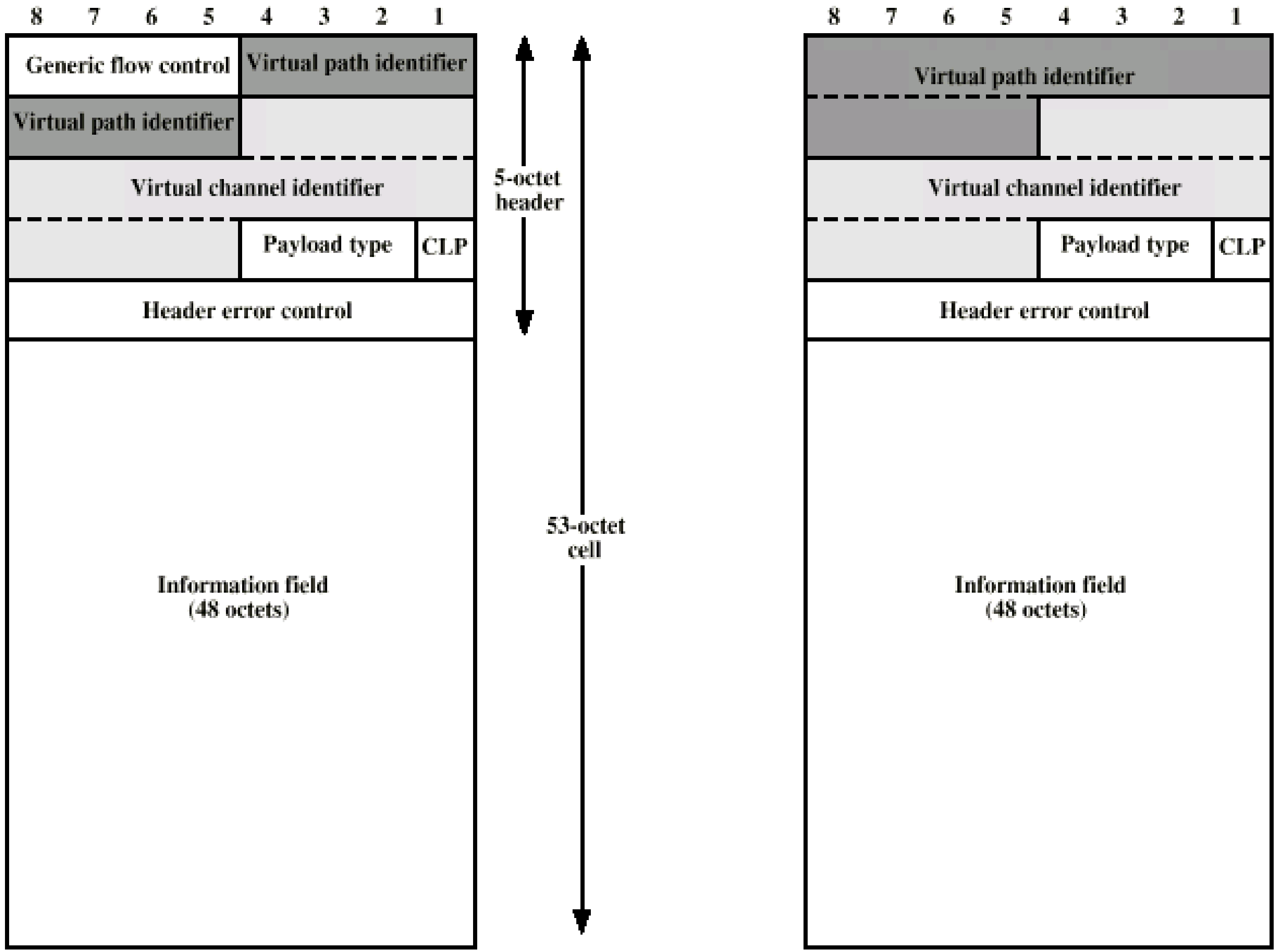
Control Signaling - VPC

- Semi-permanent
- Customer controlled
- Network controlled

ATM Cells

- Fixed size
- 5 octet header
- 48 octet information field
- Small cells reduce queuing delay for high priority cells
- Small cells can be switched more efficiently
- Easier to implement switching of small cells in hardware

ATM Cell Format



(a) User-Network Interface

(b) Network-Network Interface

Header Format

- Generic flow control
  - Only at user to network interface
  - Controls flow only at this point
- Virtual path identifier
- Virtual channel identifier
- Payload type
  - e.g. user info or network management
- Cell loss priority
- Header error control

Generic Flow Control (GFC)

- Control traffic flow at user to network interface (UNI) to alleviate short term overload
- Two sets of procedures
  - Uncontrolled transmission
  - Controlled transmission
- Every connection either subject to flow control or not Subject to flow control
  - May be one group (A) default
  - May be two groups (A and B)
- Flow control is from subscriber to network
  - Controlled by network side

GFC Field Coding

Table 11.3 Generic Flow Control (GFC) Field Coding

	Uncontrolled	Controlling → controlled		Controlled → controlling	
		1-queue model	2-queue model	1-queue model	2-queue model
First bit	0	HALT(0)/NO_HALT(1)	HALT(0)/NO_HALT(1)	0	0
Second bit	0	SET(1)/NULL(0)	SET(1)/NULL(0) for Group A	cell belongs to controlled(1) /uncontrolled(0)	cell belongs to Group A(1)/ or not (0)
Third bit	0	0	SET(1)/NULL(0) for Group B	0	cell belongs to Group B(1)/ or not (0)
Fourth bit	0	0	0	equipment is uncontrolled(0)/ controlled(1)	equipment is uncontrolled(0)/ controlled(1)

Single Group of Connections (1)

- Terminal equipment (TE) initializes two variables
  - TRANSMIT flag to 1
  - GO_CNTR (credit counter) to 0
- If TRANSMIT=1 cells on uncontrolled connection may be sent any time
- If TRANSMIT=0 no cells may be sent (on controlled or uncontrolled connections)
- If HALT received, TRANSMIT set to 0 and remains until NO_HALT

Single Group of Connections (2)

- If TRANSMIT=1 and no cell to transmit on any uncontrolled connection:
  - If GO_CNTR>0, TE may send cell on controlled connection
    - Cell marked as being on controlled connection
    - GO_CNTR decremented
  - If GO_CNTR=0, TE may not send on controlled connection
- TE sets GO_CNTR to GO_VALUE upon receiving SET signal
  - Null signal has no effect

Use of HALT

- To limit effective data rate on ATM
- Should be cyclic
- To reduce data rate by half, HALT issued to be in effect 50% of time
- Done on regular pattern over lifetime of connection

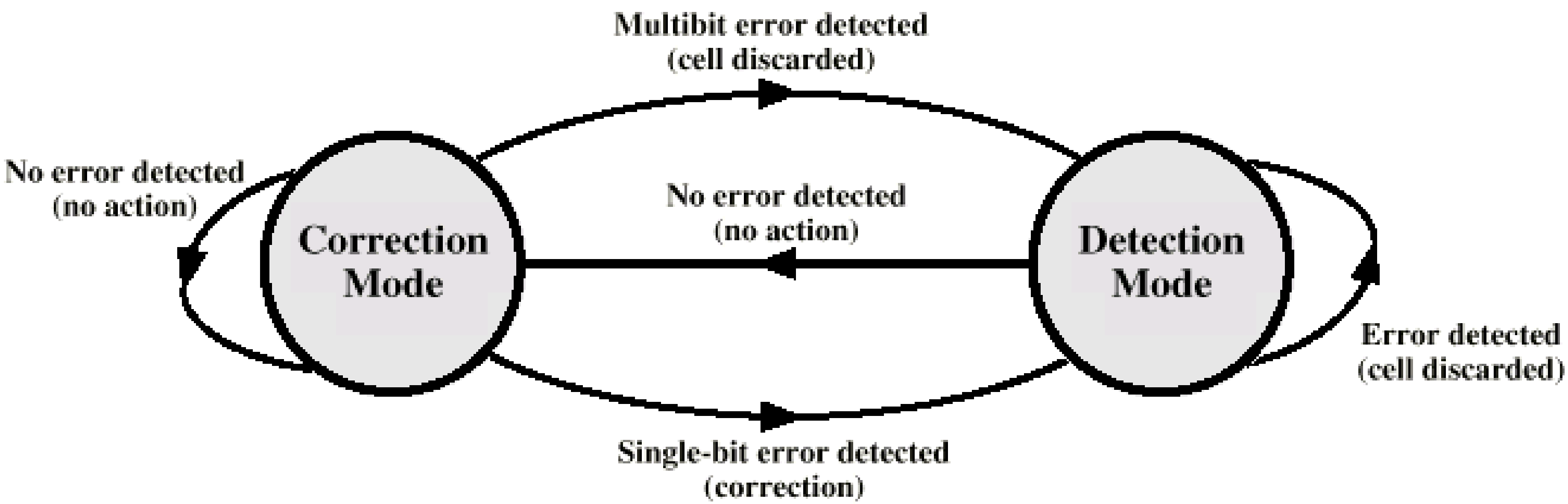
Two Queue Model

- Two counters
  - GO_CNTR_A, GO_VALUE_A,GO_CNTR_B, GO_VALUE_B

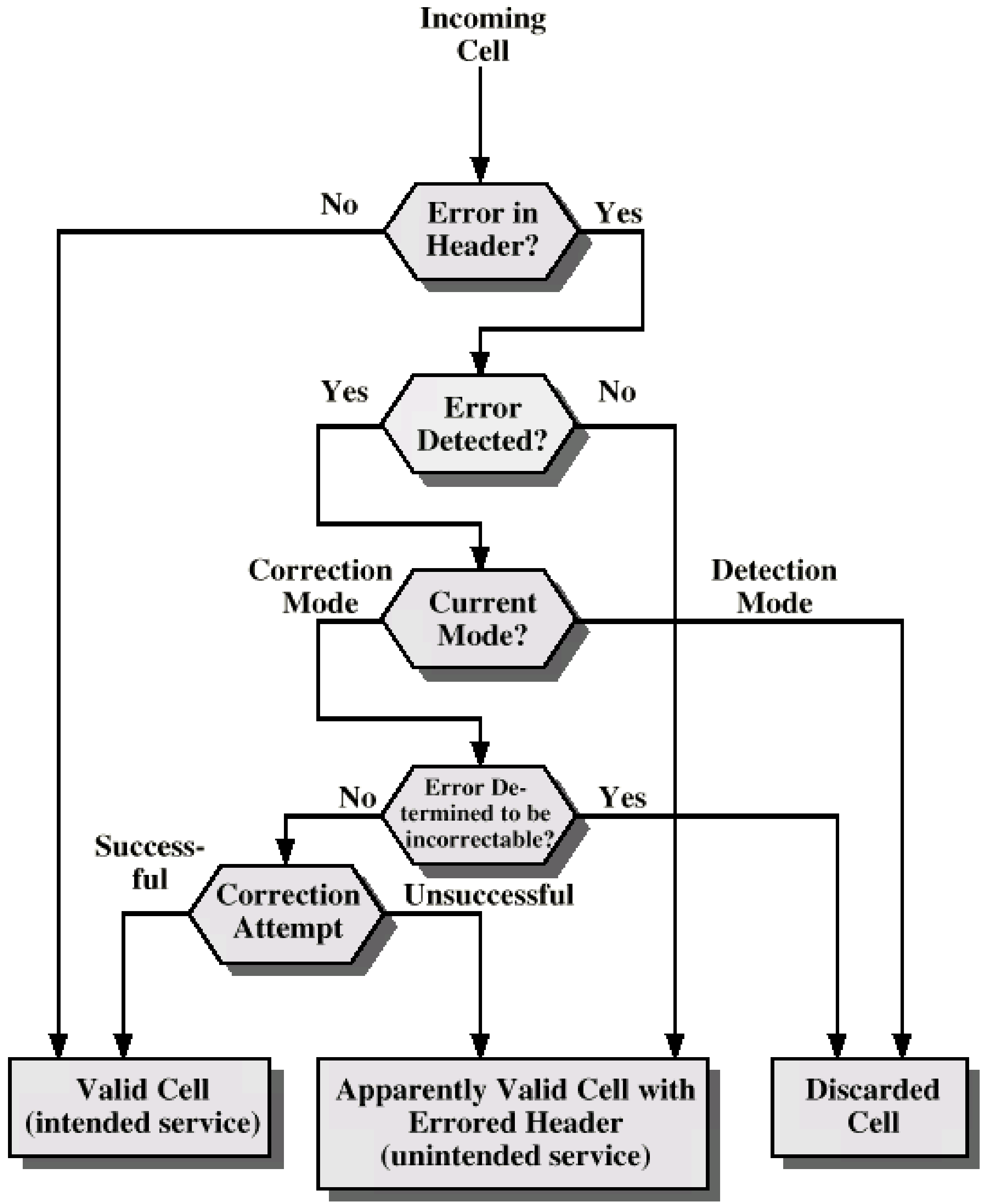
Header Error Control

- 8 bit error control field
- Calculated on remaining 32 bits of header
- Allows some error correction

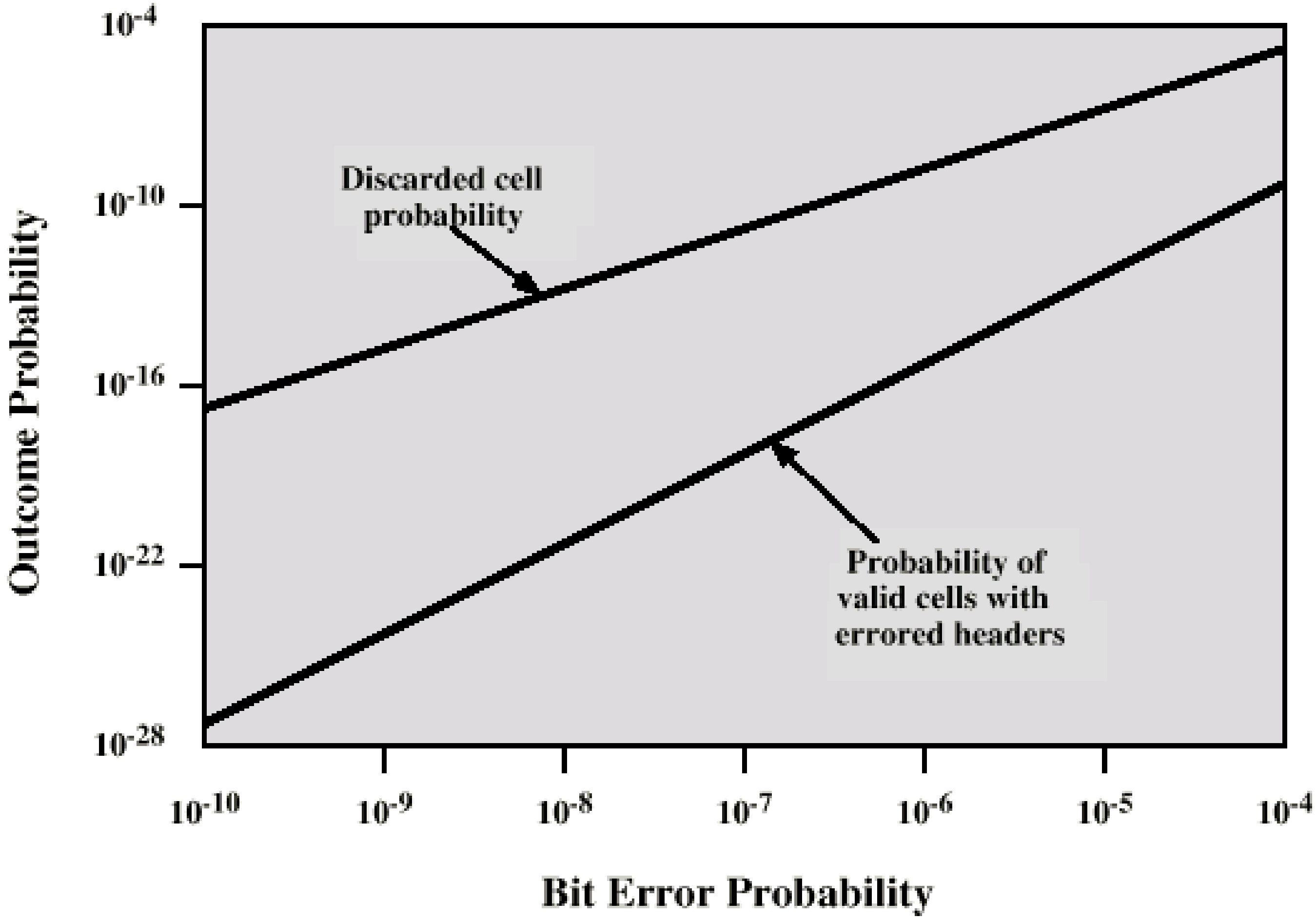
HEC (Header error control)Operation at Receiver



Effect of Error in Cell Header



Impact of Random Bit Errors on HEC Performance



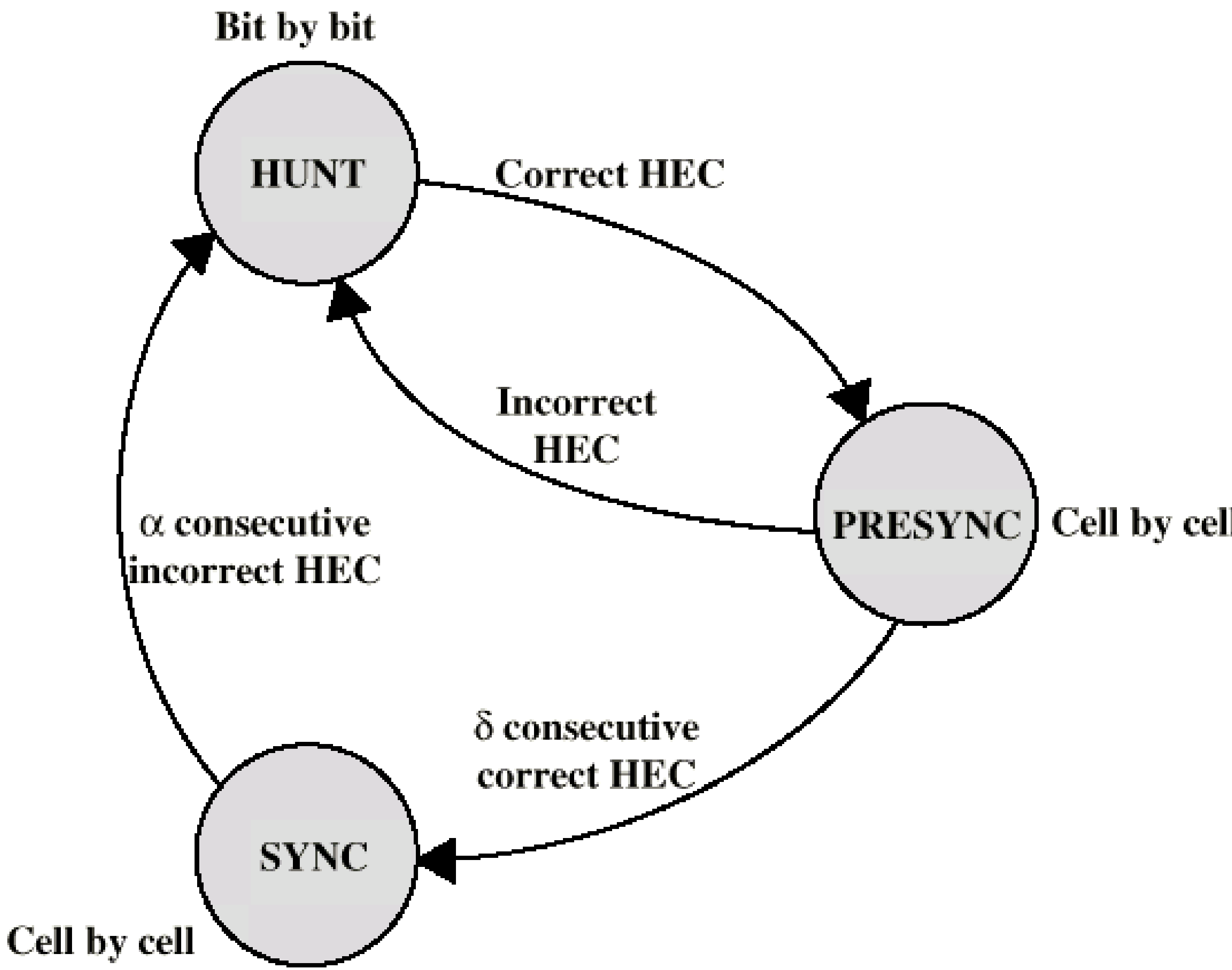
Transmission of ATM Cells

- 622.08Mbps
- 155.52Mbps
- 51.84Mbps
- 25.6Mbps
- Cell Based physical layer
- SDH based physical layer

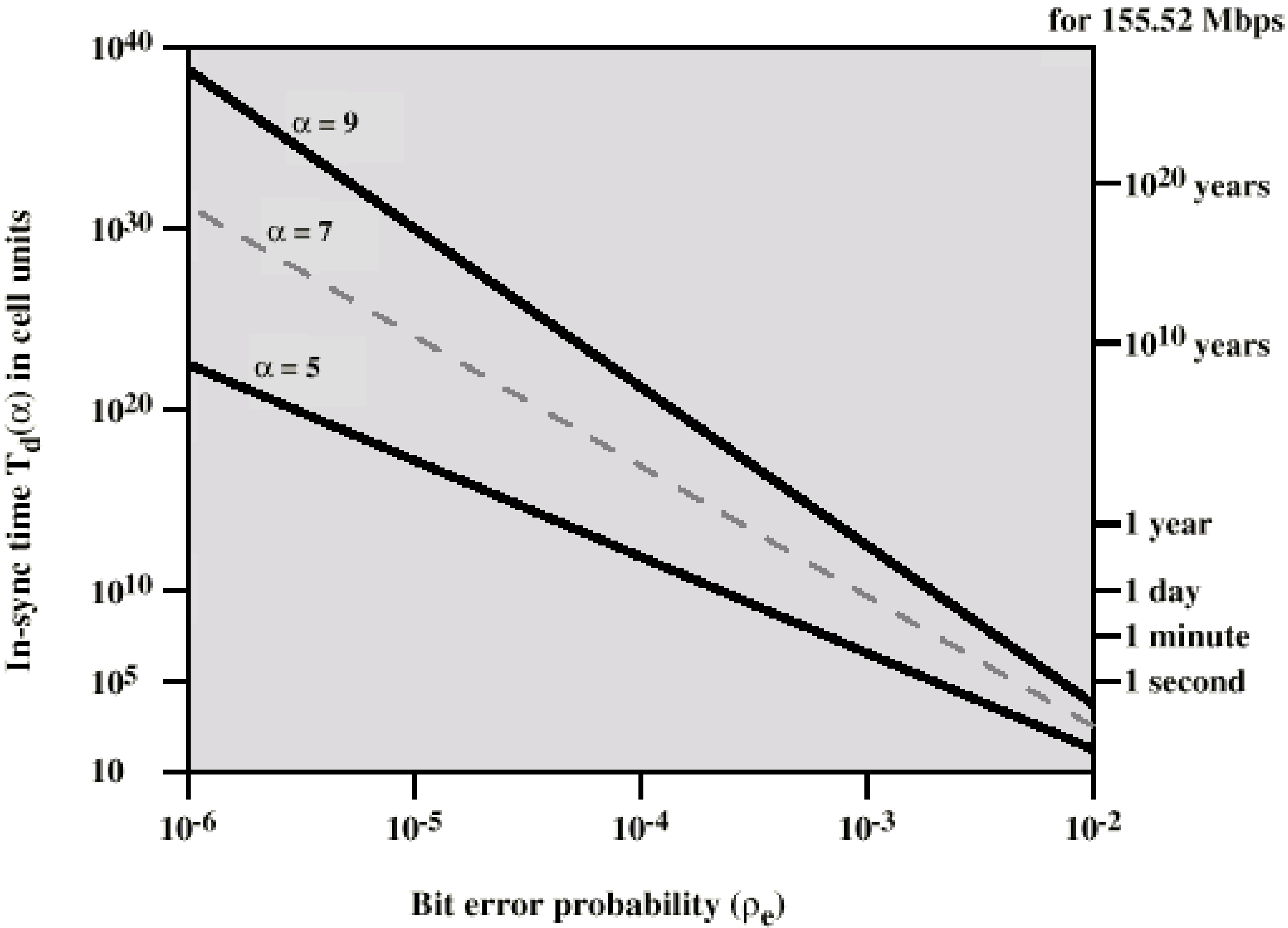
Cell Based Physical Layer

- No framing imposed
- Continuous stream of 53 octet cells
- Cell delineation based on header error control field

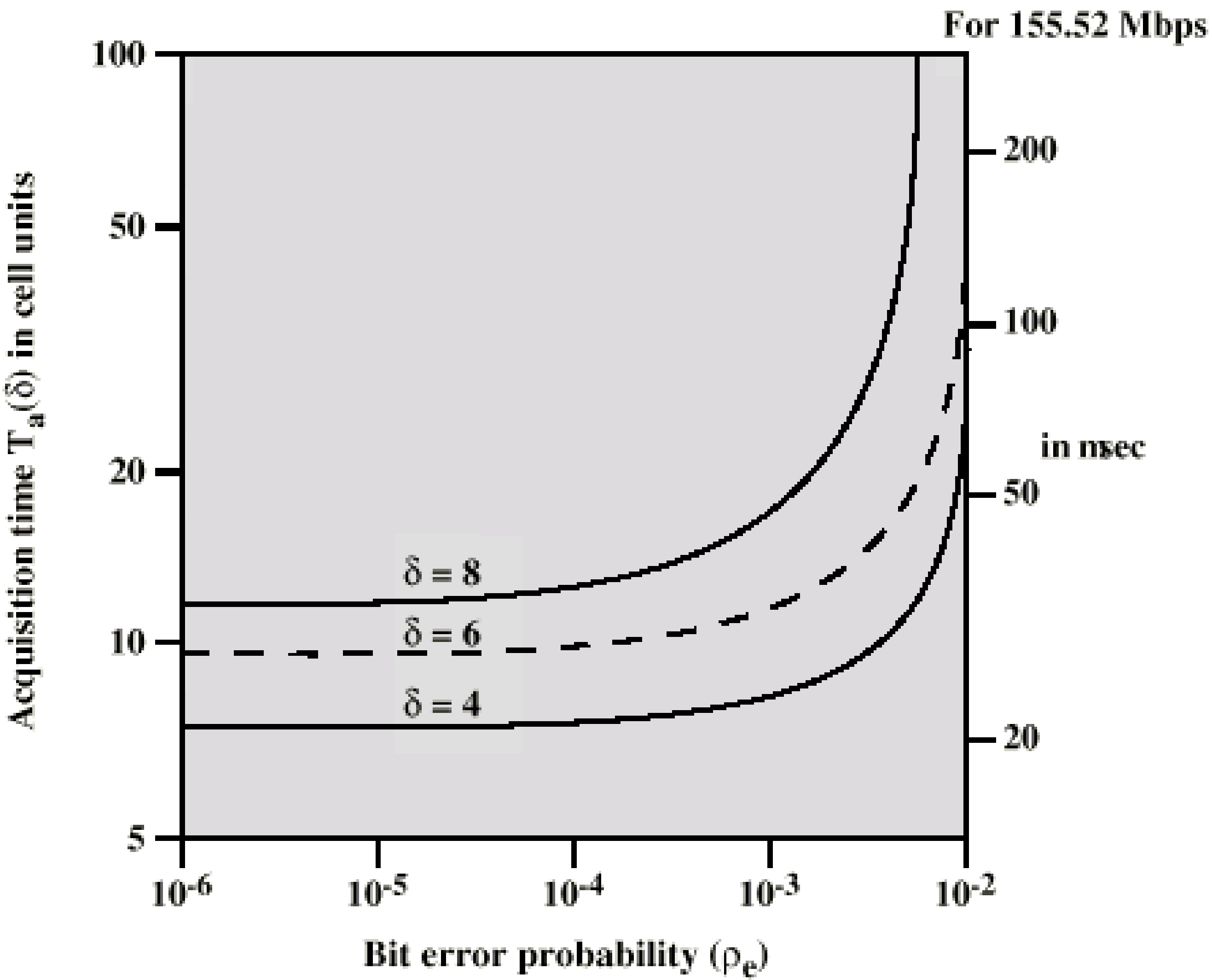
Cell Delineation State Diagram



Impact of Random Bit Errors on Cell Delineation Performance



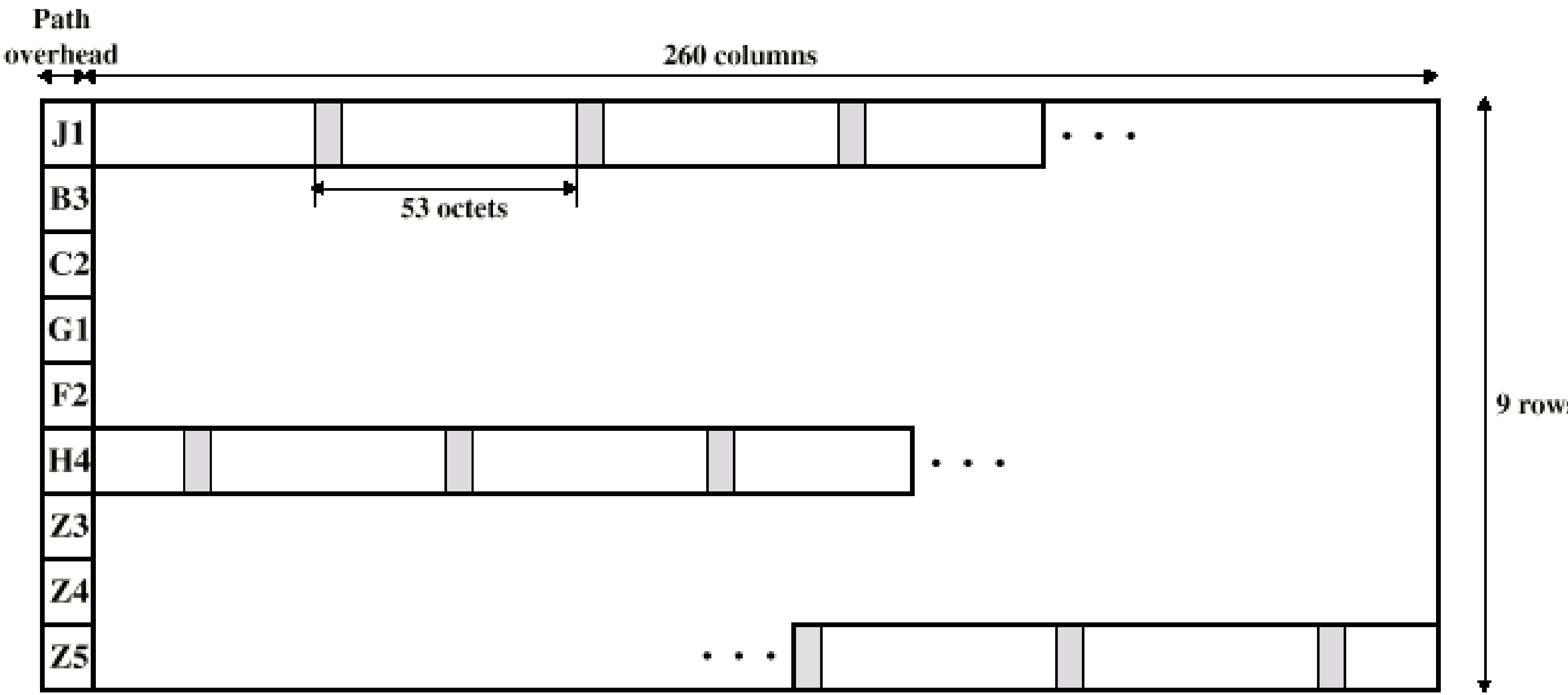
Acquisition Time v Bit Error Rate



SDH Based Physical Layer

- Imposes structure on ATM stream
- e.g. for 155.52Mbps
- Use STM-1 (STS-3) frame
- Can carry ATM and STM payloads
- Specific connections can be circuit switched using SDH channel
- SDH multiplexing techniques can combine several ATM streams

STM-1 Payload for SDH-Based ATM Cell Transmission



ATM Service Categories

- Real time
  - Constant bit rate (CBR)
  - Real time variable bit rate (rt-VBR)
- Non-real time
  - Non-real time variable bit rate (nrt-VBR)
  - Available bit rate (ABR)
  - Unspecified bit rate (UBR)
  - Guaranteed frame rate (GFR)

Real Time Services

- Amount of delay
- Variation of delay (jitter)

CBR

- Fixed data rate continuously available
- Tight upper bound on delay
- Uncompressed audio and video
  - Video conferencing
  - Interactive audio
  - A/V distribution and retrieval

rt-VBR

- Time sensitive application
  - Tightly constrained delay and delay variation
- rt-VBR applications transmit at a rate that varies with time
- e.g. compressed video
  - Produces varying sized image frames
  - Original (uncompressed) frame rate constant
  - So compressed data rate varies
- Can statistically multiplex connections

nrt-VBR

- May be able to characterize expected traffic flow
- Improve QoS in loss and delay
- End system specifies:
  - Peak cell rate
  - Sustainable or average rate
  - Measure of how bursty traffic is
- e.g. Airline reservations, banking transactions

UBR

- May be additional capacity over and above that used by CBR and VBR traffic
  - Not all resources dedicated
  - Bursty nature of VBR
- For application that can tolerate some cell loss or variable delays
  - e.g. TCP based traffic
- Cells forwarded on FIFO basis
- Best efforts service

ABR

- Application specifies peak cell rate (PCR) and minimum cell rate (MCR)
- Resources allocated to give at least MCR
- Spare capacity shared among all ARB sources
- e.g. LAN interconnection

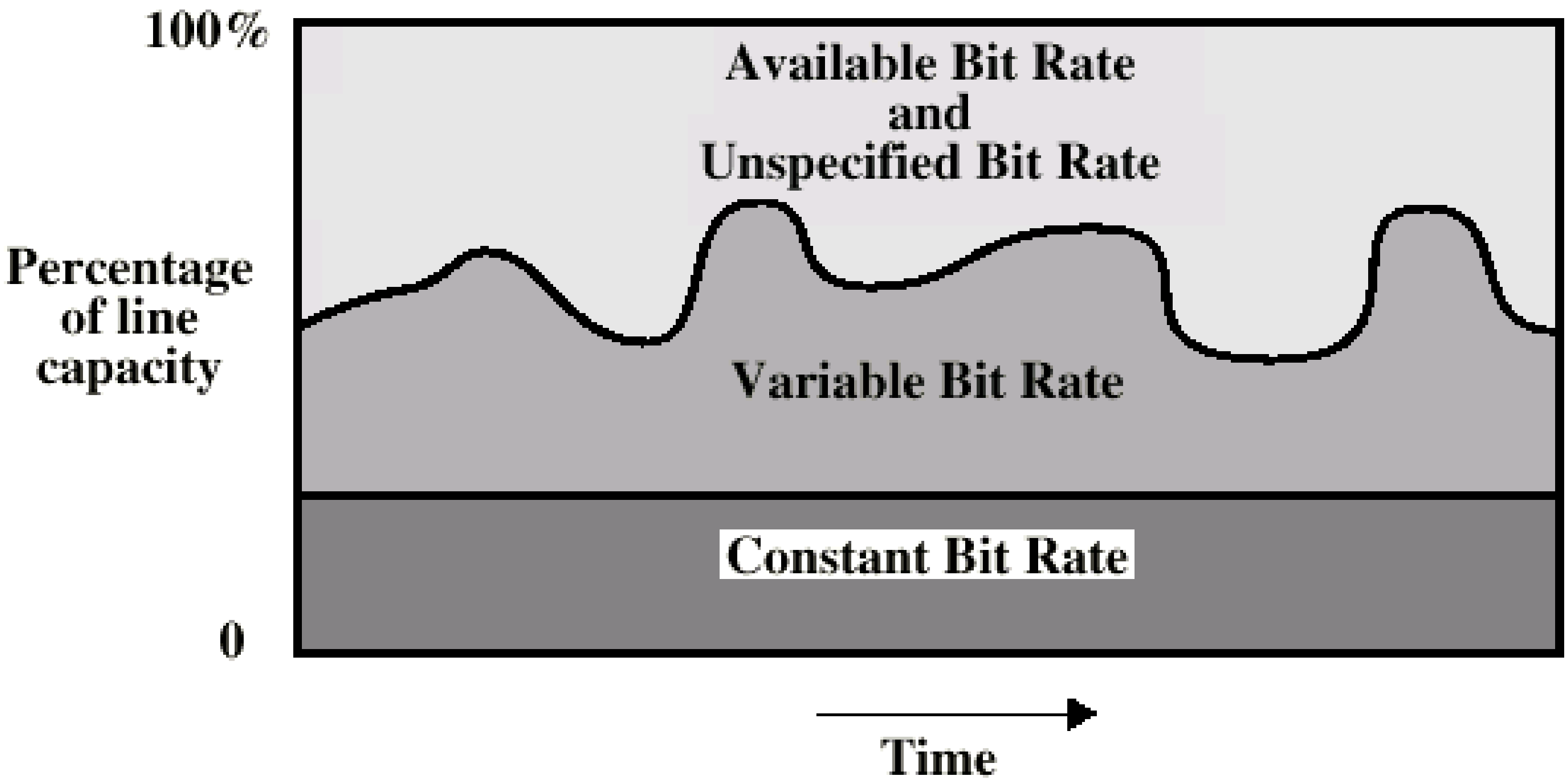
Guaranteed Frame Rate (GFR)

- Designed to support IP backbone subnetworks
- Better service than UBR for frame based traffic
  - Including IP and Ethernet
- Optimize handling of frame based traffic passing from LAN through router to ATM backbone
  - Used by enterprise, carrier and ISP networks
  - Consolidation and extension of IP over WAN
- ABR difficult to implement between routers over ATM network
- GFR better alternative for traffic originating on Ethernet
  - Network aware of frame/packet boundaries
  - When congested, all cells from frame discarded
  - Guaranteed minimum capacity
  - Additional frames carried if not congested

ATM Adaptation Layer

- Support for information transfer protocol not based on ATM
- PCM (voice)
  - Assemble bits into cells
  - Re-assemble into constant flow
- IP
  - Map IP packets onto ATM cells
  - Fragment IP packets
  - Use LAPF over ATM to retain all IP infrastructure

ATM Bit Rate Services



Adaptation Layer Services

- Handle transmission errors
- Segmentation and re-assembly
- Handle lost and misinserted cells
- Flow control and timing

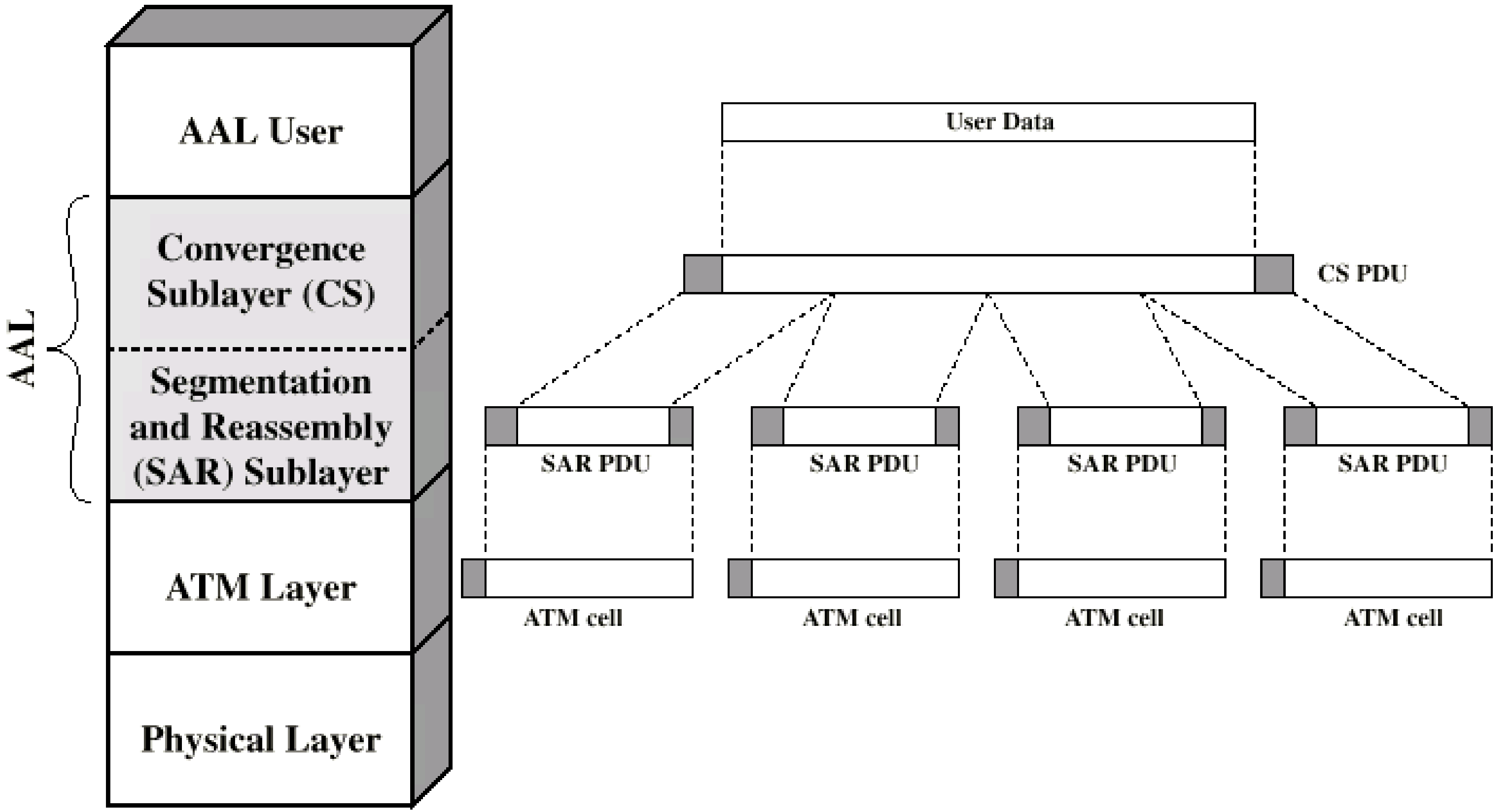
Supported Application types

- Circuit emulation
- VBR voice and video
- General data service
- IP over ATM
- Multiprotocol encapsulation over ATM (MPOA)
  - IPX, AppleTalk, DECNET)
- LAN emulation

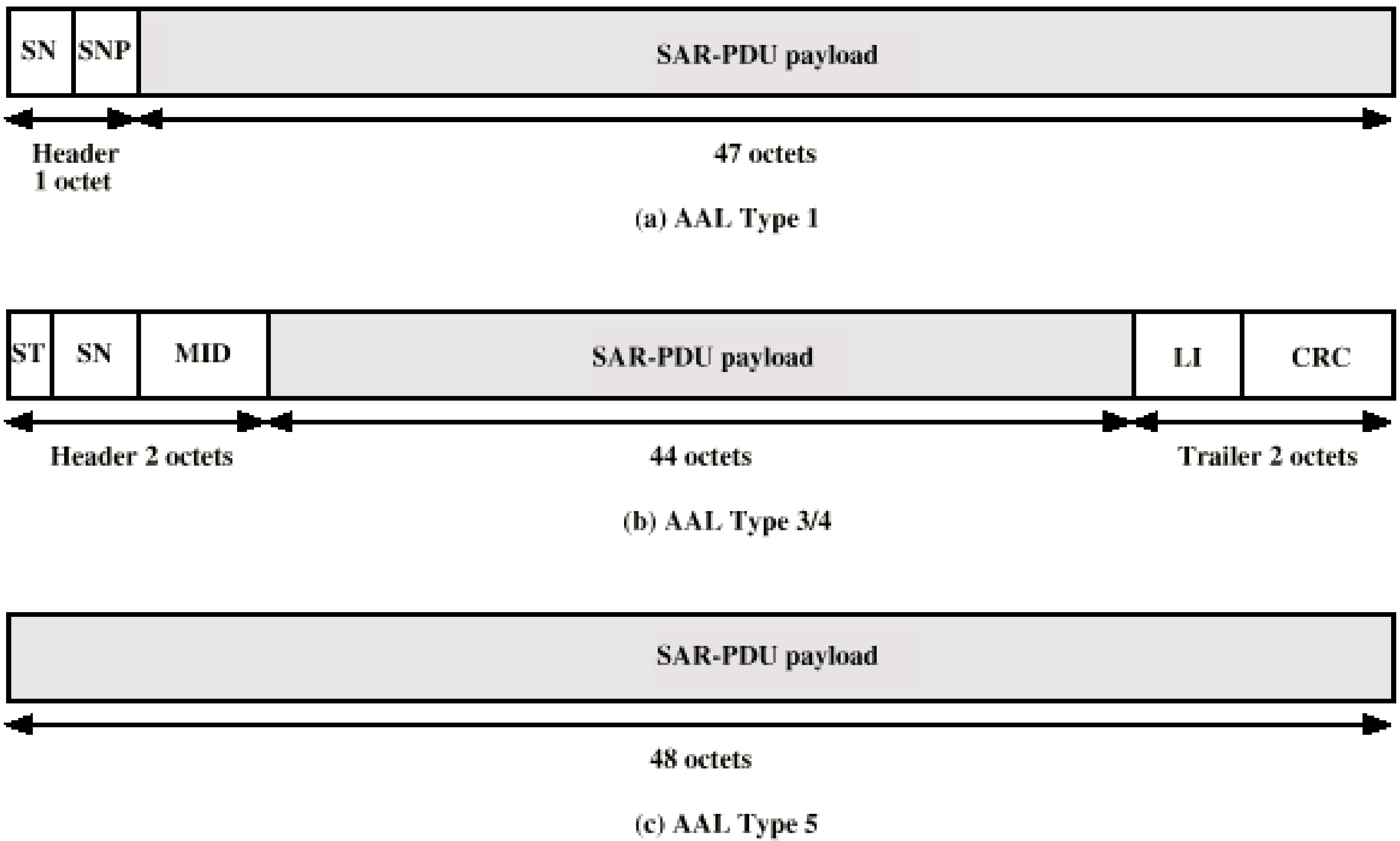
AAL Protocols

- Convergence sublayer (CS)
  - Support for specific applications
  - AAL user attaches at SAP
- Segmentation and re-assembly sublayer (SAR)
  - Packages and unpacks info received from CS into cells
- Four types
  - Type 1
  - Type 2
  - Type 3/4
  - Type 5

AAL Protocols



Segmentation and Reassembly PDU



SN = sequence number (4 bits)
SNP = sequence number protection (4 bits)
ST = segment type (2 bits)
MID = multiplexing identification (10 bits)
LI = length indication (6 bits)
CRC = cyclic redundancy check (10 bits)

AAL Type 1

- CBR source
- SAR packs and unpacks bits
- Block accompanied by sequence number

AAL Type 2

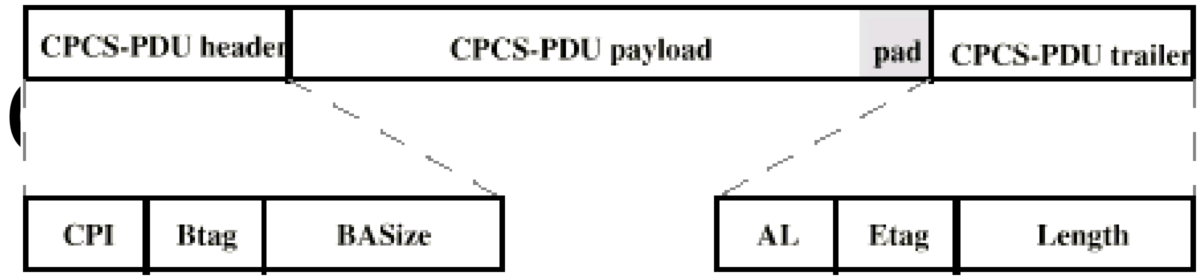
- VBR
- Analog applications

AAL Type 3/4

- Connectionless or connected
- Message mode or stream mode

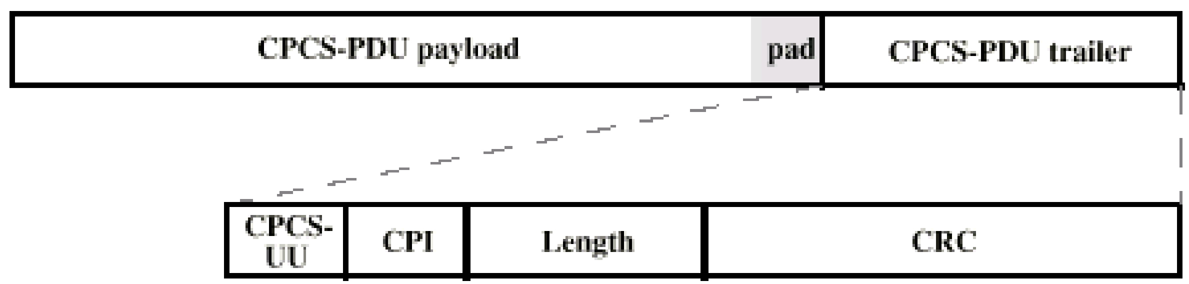
AAL Type 5

- Streamlined transport for connection oriented higher layer protocols



CPI = common part indicator (1 octet)
Btag = beginning tag (1 octet)
BASize = buffer allocation size (2 octets)
AL = alignment (1 octet)
Etag = end tag (1 octet)
Length = length of CPCS-PDU payload (2 octets)

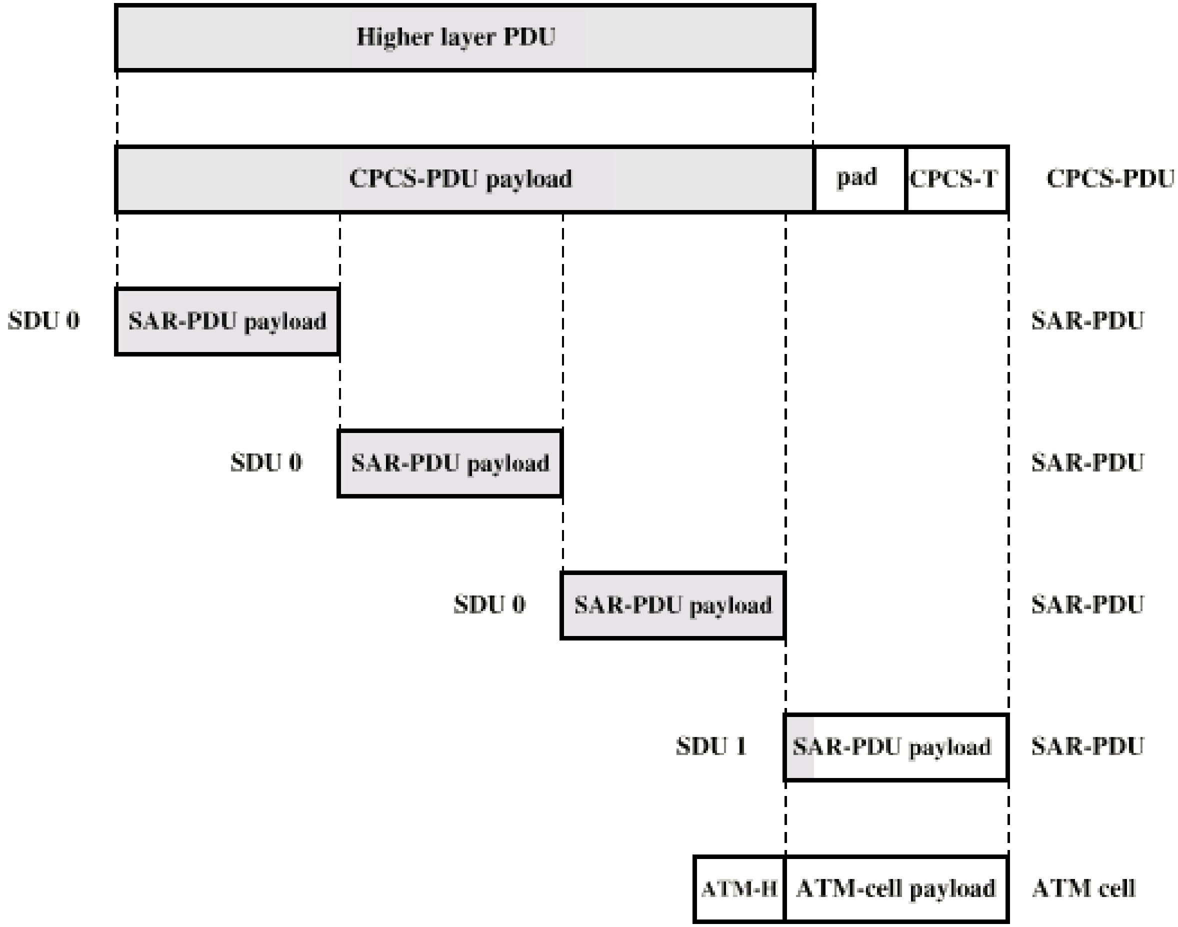
(a) AAL Type 3/4



CPCS-UU = CPCS user-to-user indication (1 octet)
CPI = common part indicator (1 octet)
Length = length of CPCS-PDU payload (2 octets)
CRC = cyclic redundancy check (4 octets)

(b) AAL Type 5

Example AAL 5 Transmission



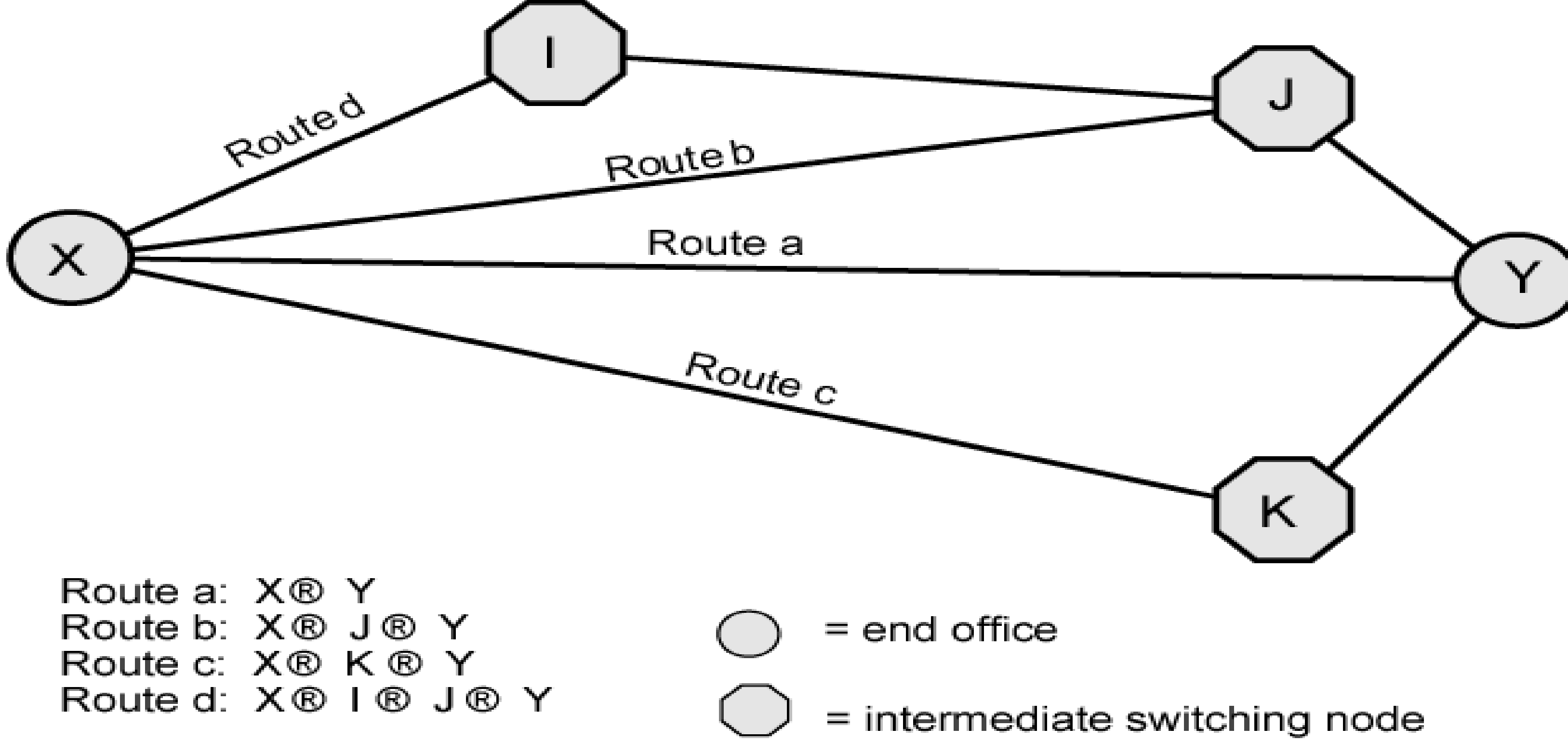
CPCS = common part convergence sublayer
SAR = segmentation and reassembly
PDU = protocol data unit
CPCS-T = CPCS trailer
ATM-H = ATM header
SDU = Service Data Unit type bit

Routing in Circuit Switched Network

- Many connections will need paths through more than one switch
- Need to find a route
  - Efficiency
  - Resilience
- Public telephone switches are a tree structure
  - Static routing uses the same approach all the time
- Dynamic routing allows for changes in routing depending on traffic
  - Uses a peer structure for nodes

Alternate Routing

- Possible routes between end offices predefined
- Originating switch selects appropriate route
- Routes listed in preference order
- Different sets of routes may be used at different times



(a) Topology

Time Period	First route	Second route	Third route	Fourth and final route
Morning	a	b	c	d
Afternoon	a	d	b	c
Evening	a	d	c	b
Weekend	a	c	b	d

(b) Routing table

Routing in Packet Switched Network

- Complex, crucial aspect of packet switched networks
- Characteristics required
  - Correctness
  - Simplicity
  - Robustness
  - Stability
  - Fairness
  - Optimality
  - Efficiency

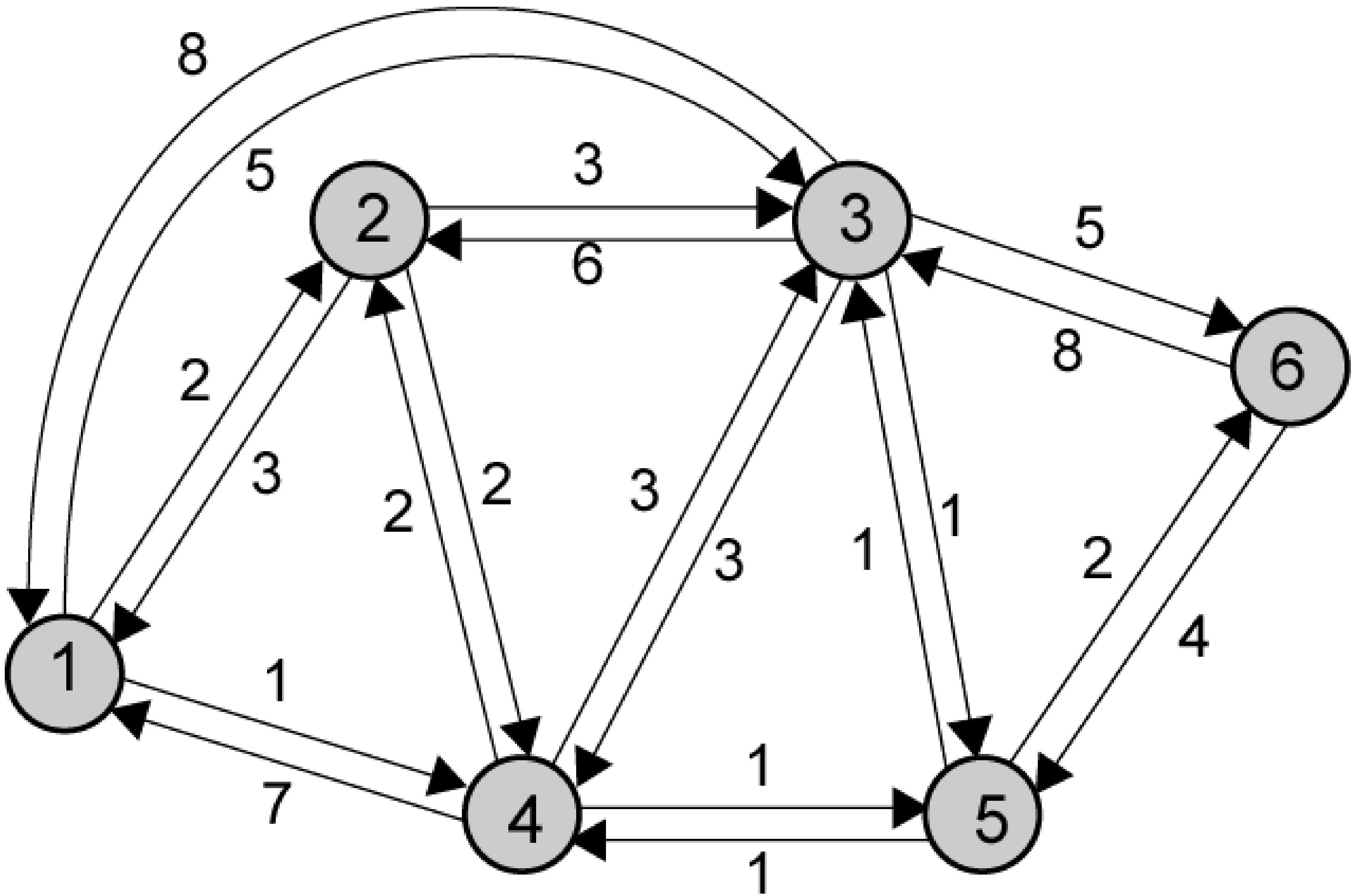
Performance Criteria

- Used for selection of route

- Minimum hop

- Least cost

Example Packet Switched Network



Decision Time and Place

- Time
  - Packet or virtual circuit basis
- Place
  - Distributed
    - Made by each node
  - Centralized
  - Source

Network Information Source and Update Timing

- Routing decisions usually based on knowledge of network (not always)
- Distributed routing
  - Nodes use local knowledge
  - May collect info from adjacent nodes
  - May collect info from all nodes on a potential route
- Central routing
  - Collect info from all nodes
- Update timing
  - When is network info held by nodes updated
  - Fixed - never updated
  - Adaptive - regular updates

Routing Strategies

- Fixed
- Flooding
- Random
- Adaptive

Fixed Routing

- Single permanent route for each source to destination pair
- Determine routes using a least cost algorithm
- Route fixed, at least until a change in network topology

Fixed Routing Tables

CENTRAL ROUTING DIRECTORY

To Node \ From Node	1	2	3	4	5	6
1	—	1	5	2	4	5
2	2	—	5	2	4	5
3	4	3	—	5	3	5
4	4	4	5	—	4	5
5	4	4	5	5	—	5
6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

Destination	Next Node
1	2
2	2
3	5
5	5
6	5

Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	6

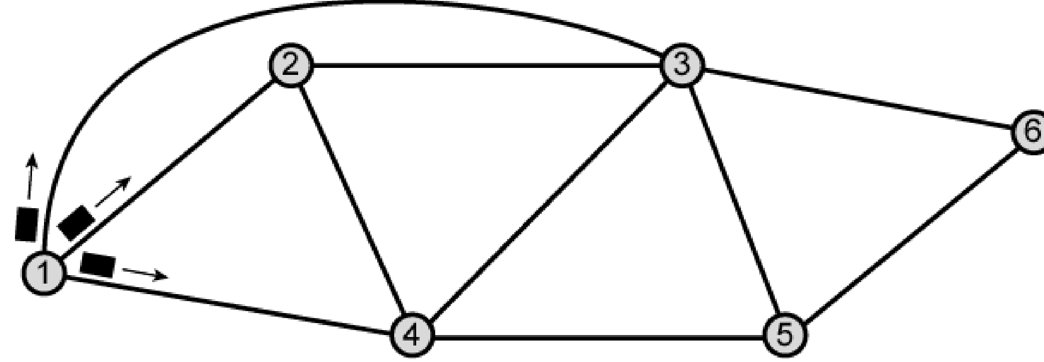
Node 6 Directory

Destination	Next Node
1	5
2	5
3	5
4	5
5	5

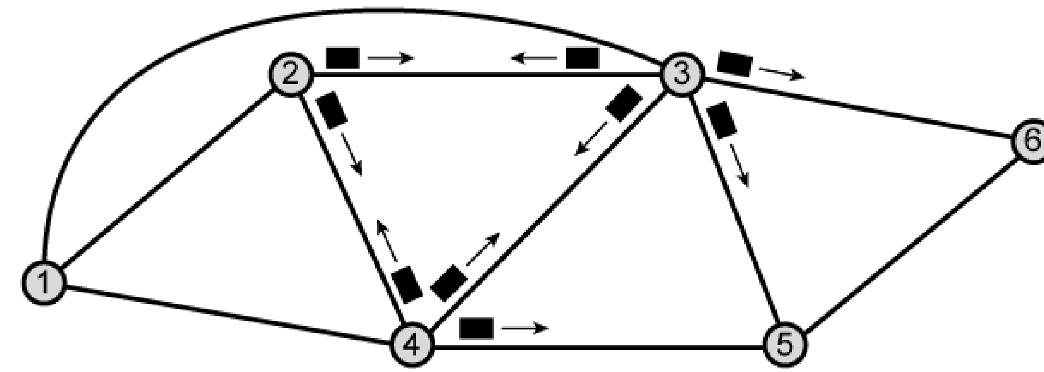
Flooding

- No network info required
- Packet sent by node to every neighbor
- Incoming packets retransmitted on every link except incoming link
- Eventually a number of copies will arrive at destination
- Each packet is uniquely numbered so duplicates can be discarded
- Nodes can remember packets already forwarded to keep network load in bounds
- Can include a hop count in packets

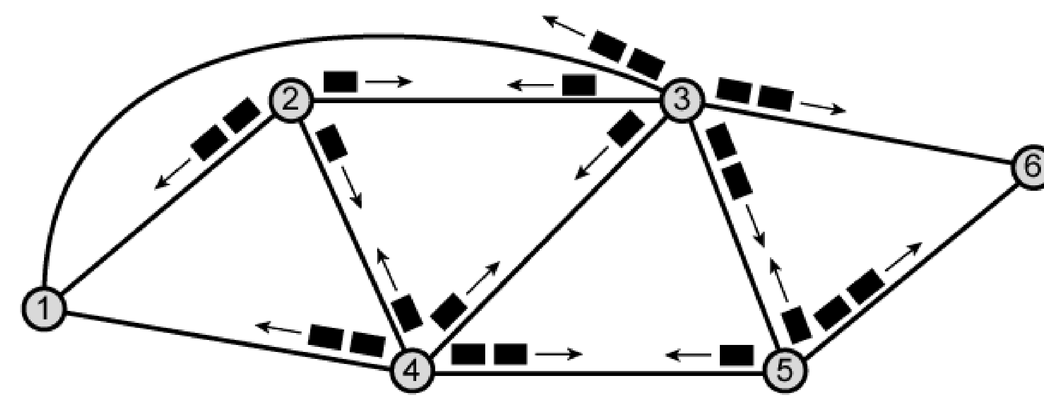
Flooding Example



(a) First hop



(b) Second hop



(c) Third hop

Properties of Flooding

- All possible routes are tried
  – Very robust
- At least one packet will have taken minimum hop count route
  – Can be used to set up virtual circuit
- All nodes are visited
  – Useful to distribute information (e.g. routing)

Random Routing

- Node selects one outgoing path for retransmission of incoming packet
- Selection can be random or round robin
- Can select outgoing path based on probability calculation
- No network info needed
- Route is typically not least cost nor minimum hop

Adaptive Routing

- Used by almost all packet switching networks
- Routing decisions change as conditions on the network change
  - Failure
  - Congestion
- Requires info about network
- Decisions more complex
- Tradeoff between quality of network info and overhead
- Reacting too quickly can cause oscillation
- Too slowly to be relevant

Adaptive Routing - Advantages

- Improved performance
- Aid congestion control
- Complex system
  - May not realize theoretical benefits

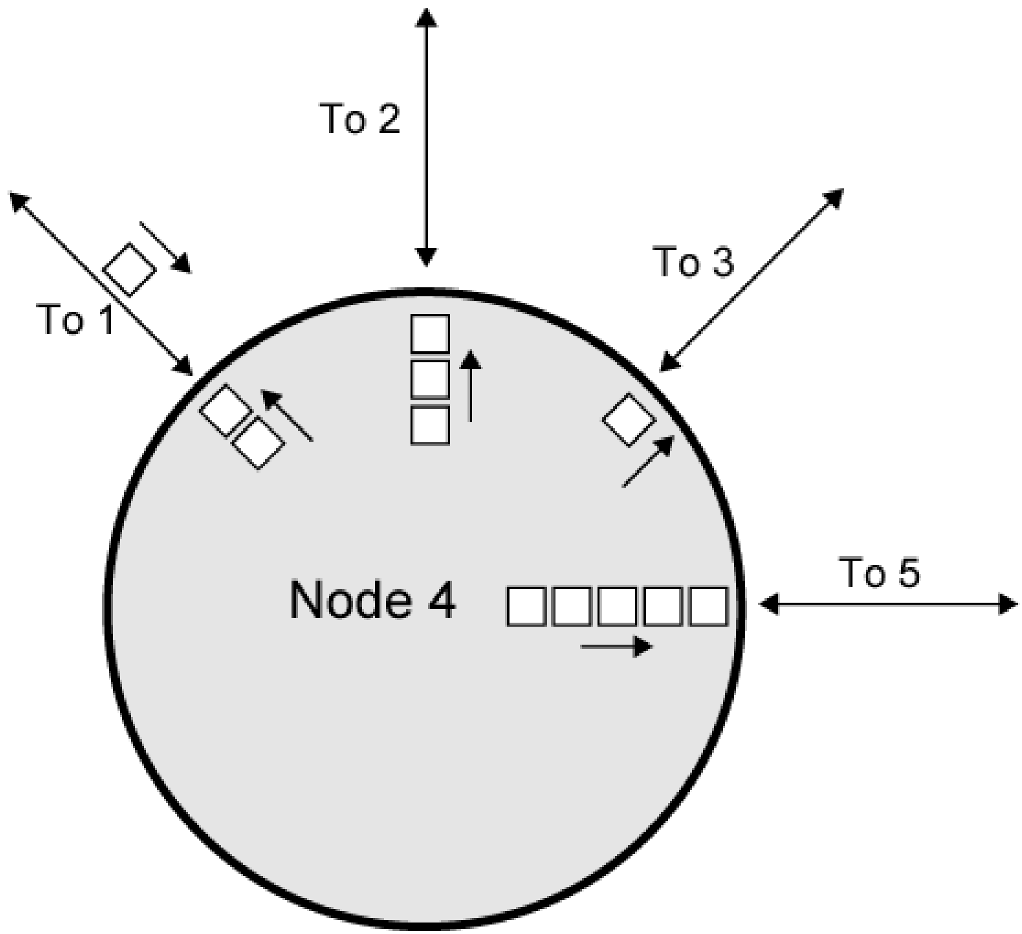
Classification

- Based on information sources
  - Local (isolated)
    - Route to outgoing link with shortest queue
    - Can include bias for each destination
    - Rarely used - do not make use of easily available info
  - Adjacent nodes
  - All nodes

Isolated Adaptive Routing

Node 4's Bias
Table for
Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0



Least Cost Algorithms

- Basis for routing decisions
  - Can minimize hop with each link cost 1
  - Can have link value inversely proportional to capacity
- Given network of nodes connected by bi-directional links
- Each link has a cost in each direction
- Define cost of path between two nodes as sum of costs of links traversed
- For each pair of nodes, find a path with the least cost
- Link costs in different directions may be different
  - E.g. length of packet queue

Dijkstra's Algorithm Definitions

- Find shortest paths from given source node to all other nodes, by developing paths in order of increasing path length
- N = set of nodes in the network
- s = source node
- T = set of nodes so far incorporated by the algorithm
- w(i, j) = link cost from node i to node j
  - w(i, i) = 0
  - w(i, j) = ∞ if the two nodes are not directly connected
  - w(i, j) $\geq$ 0 if the two nodes are directly connected
- L(n) = cost of least-cost path from node s to node n currently known
  - At termination, L(n) is cost of least-cost path from s to n

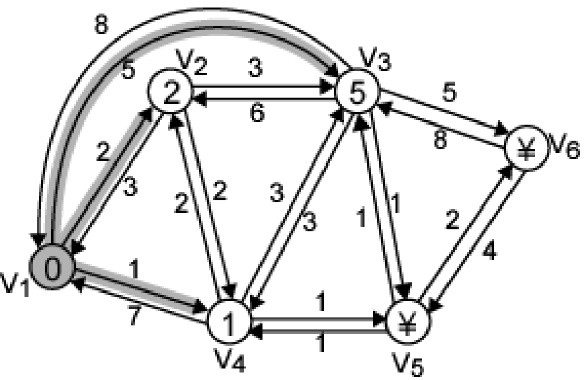
Dijkstra's Algorithm Method

- Step 1 [Initialization]
  - $T = \{s\}$ Set of nodes so far incorporated consists of only source node
  - $L(n) = w(s, n)$ for $n \neq s$
  - Initial path costs to neighboring nodes are simply link costs
- Step 2 [Get Next Node]
  - Find neighboring node not in T with least-cost path from s
  - Incorporate node into T
  - Also incorporate the edge that is incident on that node and a node in T that contributes to the path
- Step 3 [Update Least-Cost Paths]
  - $L(n) = \min[L(n), L(x) + w(x, n)]$ for all $n \notin T$
  - If latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n
- Algorithm terminates when all nodes have been added to T

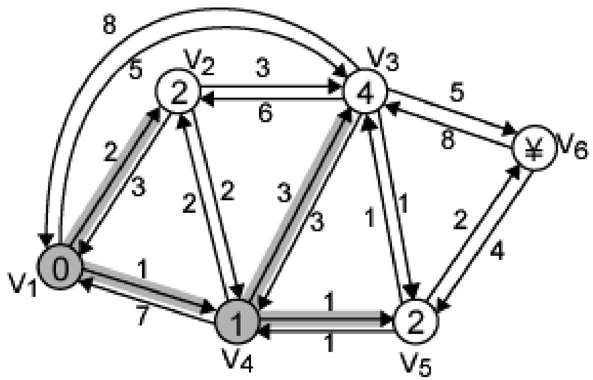
Dijkstra's Algorithm Notes

- At termination, value L(x) associated with each node x is cost (length) of least-cost path from s to x.
- In addition, T defines least-cost path from s to each other node
- One iteration of steps 2 and 3 adds one new node to T
  - Defines least cost path from s tothat node

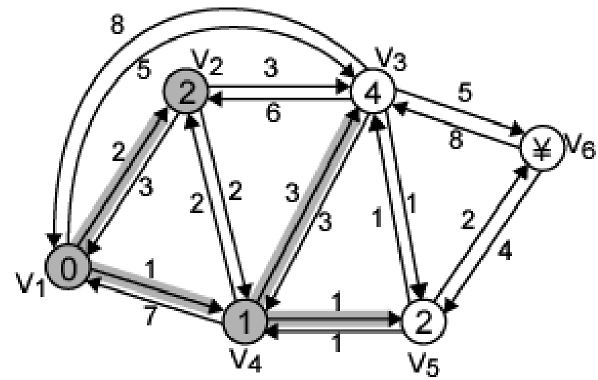
Example of Dijkstra’s Algorithm



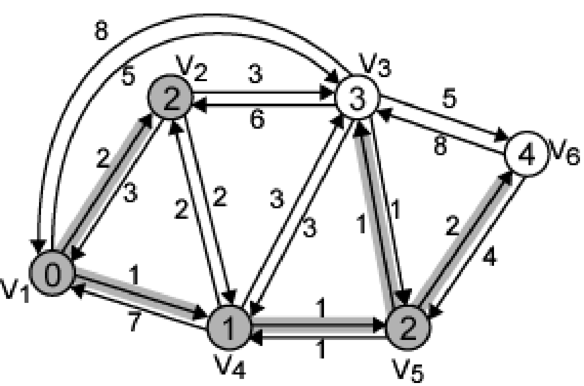
T = {1}



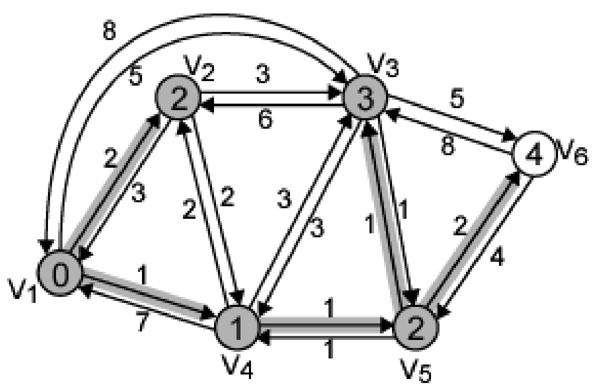
T = {1, 4}



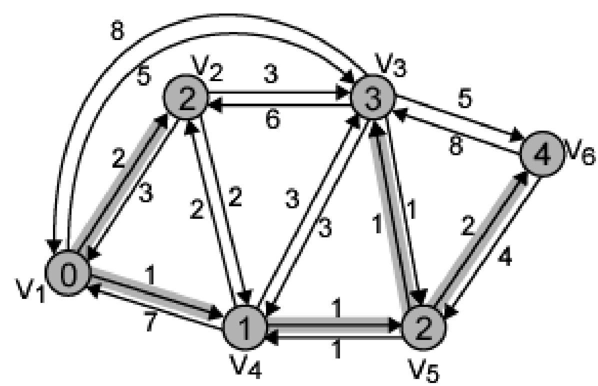
T = {1, 2, 4}



T = {1, 2, 4, 5}



T = {1, 2, 3, 4, 5}



T = {1, 2, 3, 4, 5, 6}

Results of Example Dijkstra's Algorithm

Iteration	T	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{1}	2	1–2	5	1-3	1	1–4	∞	-	∞	-
2	{1,4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
3	{1, 2, 4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
4	{1, 2, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
5	{1, 2, 3, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4–5	4	1-4-5-6

Bellman-Ford Algorithm Definitions

- Find shortest paths from given node subject to constraint that paths contain at most one link
- Find the shortest paths with a constraint of paths of at most two links
- And so on
- s = source node
- w(i, j) = link cost from node i to node j
  - w(i, i) = 0
  - w(i, j) = ∞ if the two nodes are not directly connected
  - w(i, j) $\geq$ 0 if the two nodes are directly connected
- h = maximum number of links in path at current stage of the algorithm
- $L_h(n)$ = cost of least-cost path from s to n under constraint of no more than h links

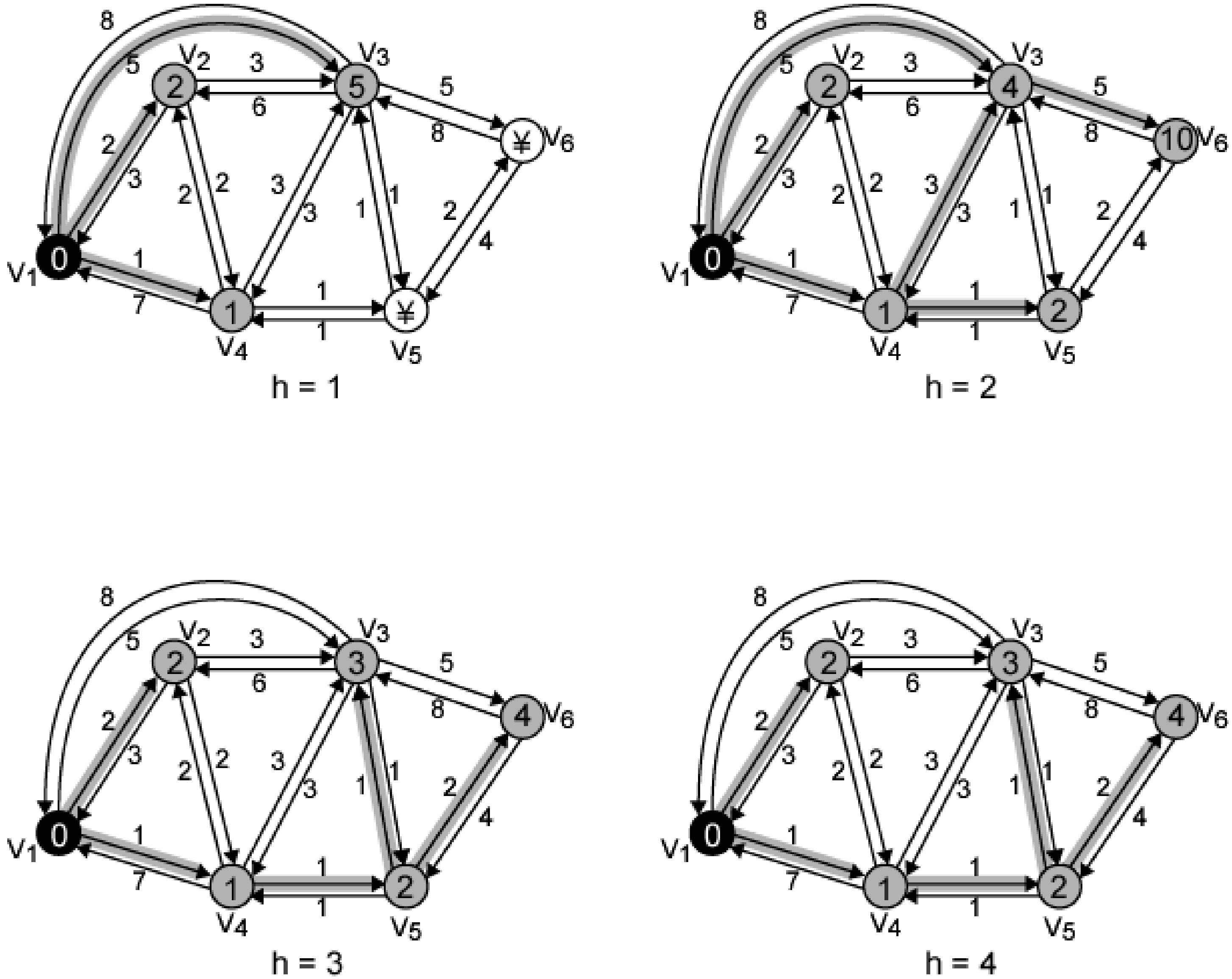
Bellman-Ford Algorithm Method

- Step 1 [Initialization]
  - $L_0(n) = \infty$, for all $n \neq s$
  - $L_h(s) = 0$, for all h
- Step 2 [Update]
- For each successive $h \geq 0$
  - For each $n \neq s$, compute
  - $L_{h+1}(n) = \min_j [L_h(j) + w(j,n)]$
- Connect n with predecessor node j that achieves minimum
- Eliminate any connection of n with different predecessor node formed during an earlier iteration
- Path from s to n terminates with link from j to n

Bellman-Ford Algorithm Notes

- For each iteration of step 2 with h=K and for each destination node n, algorithm compares paths from s to n of length K=1 with path from previous iteration
- If previous path shorter it is retained
- Otherwise new path is defined

Example of Bellman-Ford Algorithm



Results of Bellman-Ford Example

h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Comparison

- Results from two algorithms agree
- Information gathered
  - Bellman-Ford
    - Calculation for node n involves knowledge of link cost to all neighboring nodes plus total cost to each neighbor from s
    - Each node can maintain set of costs and paths for every other node
    - Can exchange information with direct neighbors
    - Can update costs and paths based on information from neighbors and knowledge of link costs
  - Dijkstra
    - Each node needs complete topology
    - Must know link costs of all links in network
    - Must exchange information with all other nodes

Evaluation

- Dependent on processing time of algorithms
- Dependent on amount of information required from other nodes
- Implementation specific
- Both converge under static topology and costs
- Converge to same solution
- If link costs change, algorithms will attempt to catch up
- If link costs depend on traffic, which depends on routes chosen, then feedback
  - May result in instability