

Buffering

Relevant sections of Keshav:
§ 9.7: Packet dropping

Copyright © 2003, Tim Moors

Forwarding modes

Switches receive frames/packets and forward them
Forwarding *could* start as soon as packet has been classified;
DA in header + fast classification \Rightarrow could forward before receive last bit

Switch modes:

- **Store-and-forward**: store complete frame before forwarding
 - **Cut-through**: start forwarding before frame has been completely received
 - Aims to reduce the delay across the switch
 - $\text{delay}_{\min}(\text{store-and-forward}) = \text{frame length} / \text{transmission rate}$,
e.g. 1.2ms for 10Mb/s Ethernet
 - May become important when path traverses multiple switches
 - Alternative technique is to reduce frame size, e.g. ATM
 - Subject to MAC, e.g. CSMA/CD access delays
 - Possible only when there is no queuing at the output port
- “cut-through is not an *alternative* to store-and-forward operation, it is *in addition* to it.”

Copyright © 2003, Tim Moors

Cut-through

“Switches can cut-through, but routers can’t. So switches are faster.”

Theory: Layers are separate, and higher layer doesn’t start processing incoming packet until lower layer has finished with it.
Interpretation of network layer (IP) fields can’t start until frame has been fully received.

Practice:

- Switches at low layers may interpret fields from higher layers.
e.g. frame switch may inspect IP address to determine outgoing port.
- Violates insulation benefits of layering, but provides marketing advantage.

Cut-through is difficult when:

- input rate < output rate
- transmission on multiple output ports (multicast & unicast but don’t know which port to use)

Copyright © 2003, Tim Moors

Store-and-forward advantages

- Confining errored frames
- Confining runt frames

Bottom line: **Store-and-forward and cut-through have essentially the same performance**

Copyright © 2003, Tim Moors

Confining errored frames

(i.e. not forwarding them)

- Need to receive complete frame before can verify integrity.
- Strictly, need to verify integrity before basing forwarding decisions on DA.
- Bridges shouldn’t learn from errored frames; SA could be erroneous, leading to latter mis-directing of frames.
- Some switches offer a hybrid mode: Observe frame error rates. If low, use cut-through, else use store-and-forward.

One measurement: 9032273 frames, 4694 CRC mismatch (5E-4), 72 runts (8E-6), 2059 giants (2E-4)

Copyright © 2003, Tim Moors

Confining “runt frames”

“runt frames”: frames that are too small

- Ethernet imposes minimum frame length (64B, excluding preamble) – source may be unaware of collisions for shorter frames
 - Most protocols have frame overheads (e.g. 802.3: 26B, 802.11: 28B, FDDI: 32B)
 \Rightarrow frames shorter than minimum overhead are invalid
- “fragment-free cut-through” ensures frame exceeds minimum length, but cuts-through once this condition is met.

Store-and-forward can also confine “giant frames”.

Copyright © 2003, Tim Moors

Outline of discard strategies

Tagging to indicate loss priority

Units of transmission vs control

Partial Packet Discard

Early Packet Discard

Random Early Discard

} when transmission unit =
buffering unit < control unit
try to manage congestion

Copyright © 2003, Tim Moors

Source tagging of loss priority

Some applications exchange packets that vary in their loss sensitivity.

e.g. streaming media can often be coded hierarchically:

one level of low-fidelity baseband information

+ one or more levels of enhancement information (less loss sensitive)

e.g. normal definition TV + HDTV supplement.

lecture: slides + audio + video of lecturer

Such applications benefit from the network discarding

unimportant packets in order to transfer important packets.

(Priority is only relative to other packets from this application)

- Sources have incentive to “tag” packets, indicating their loss sensitivity

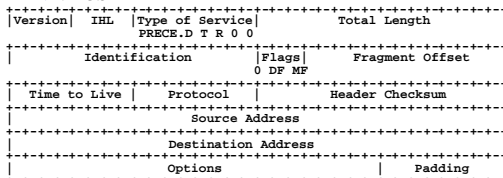
Copyright © 2003, Tim Moors

Implementing tagging

ATM: Cell Loss Priority bit in cell header

IPv4: TOS

R = reliability (1=lower loss, 0=higher loss)



IPv6: set the Traffic Class/Differentiated Services field to indicate the required per hop behaviour (drop priority)

Copyright © 2003, Tim Moors

Network tagging of loss priority

To guarantee service: Applications negotiate a contract with the network.

- Network agrees to provide service guarantees.
- Application agrees to abide by certain traffic profile (e.g. as described by a Leaky Bucket).

Network may “police” source traffic to ensure that it conforms by the agreed profile.

- Traffic that does not conform to the profile may be summarily discarded, or may be *tagged* for preferential discard within the network.

... end of tagging discussion.

Copyright © 2003, Tim Moors

Simplistic buffer management

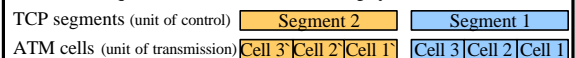
- First-In First-Out + Drop-Tail
- FIFO:
 - Simple
 - Inefficient (e.g. head of line blocking)
 - Service is proportional to rate \Rightarrow doesn't penalise fast transmitters.
- Drop-Tail: Incoming packets are discarded when the queue is full, irrespective of priority of incoming/queued packets.
 - May be better to drop from front, since then the source will learn of congestion earlier.

Copyright © 2003, Tim Moors

Transmission unit < control unit

Partial & Early Packet discard are used when the unit of transmission < unit of “control” (what can be retransmitted)

e.g. TCP segment (KB) \gg ATM cell (53B: too small for cell sequence numbers; \leq 48B of payload)



Reservation of buffer space for packets may not be:

- possible: packet length in AAL 3/4 but not AAL 5
- desirable: reservation is conservative: may block packets that could have passed

Copyright © 2003, Tim Moors

Partial Packet Discard

Applications respond differently to partial loss.

Crude dichotomy:

- “data” – complete delivery is important – program with missing code will likely crash (sometime)
- “streaming media” (voice/video) – omission of one sample isn’t significant (provided the media isn’t heavily compressed)
- If one cell is lost, data application will retransmit *entire* larger control unit (packet/segment/file).
- If one cell is lost due to congestion, might as well discard remaining cells of control unit; they’ll be retransmitted anyhow. Switch continues discarding remaining transmission units of same control unit (assuming it can identify control units – layer violations)

⇒ Partial Packet Discard

Downside: Switch may behave inappropriately for other higher layers (e.g. voice application)

Copyright © 2003, Tim Moors

Early Packet Discard

Policy: If buffer fill exceeds a threshold then don’t accept new packets. Start discarding cells before buffer fills (!)

Motivation: Focus on existing packets for which resources have already been invested. Packet loss may also act as implicit signal of imminent congestion.

Copyright © 2003, Tim Moors

Random Early Discard/“Detection”

(moving away from cells, and into packets...)

Congestion occurs when:

- Routers discard packets (obvious)
- Delay/throughput is high; may sacrifice some throughput in order to reduce delay

Congestion is caused by sources transmitting too fast; *TCP* interprets loss as indicating congestion, and reduces rate

Router may discard *packets* to signal imminent congestion; sources will *hopefully* slow down and so avoid congestion (e.g. high delay)

[Explicit Congestion Notification may be better, since it won’t cause packet discard for apps that don’t slow down in response to loss.]

Copyright © 2003, Tim Moors

Have fun in seminar week!

Copyright © 2003, Tim Moors