6. Media access control data path

This clause describes per ringlet behavior, unless explicitly mentioned otherwise.

6.1 Flow-control overview

Editors' Notes (DVJ): *To be removed prior to final publication.*

The following acronyms should be updated in the central clause:

PTQ: primary transit queue.
STQ: secondary transit queue.

6.1.1 Traffic classes

Client data is classified into 3 traffic classes, as listed in table 3. Class-A traffic is provisioned with a committed information rate (CIR), and provides the lowest MAC delay and jitter[1] bounds. Class-B traffic is provisioned with a CIR, and provides bounded MAC delay and jitter for that traffic within the profile of the CIR. Class-B traffic beyond the provisioned CIR is referred to as excess information rate (EIR) class-B traffic. Class-C traffic is not provisioned. EIR class-B and class-C traffic is marked as out of profile by the MAC, and provide no MAC delay or jitter bounds.

Editors' Notes (jl): *To be removed prior to final publication.*

The exact jitter bounds have not been calculated. Class-A jitter is on the order of N * MTU, where N is the number of stations on the ring. Class-B jitter is on the order of RTT. More precise calculations are needed.

Table 3—Service Classes

Name	Bandwdith	Jitter	Row	Example of use
Class-A	provisioned	small	1	Real time
Class-B	provisioned	bounded	2	Near real time
Class-C	opportunistic	unbounded	3	Best effort network traffic

The class-C traffic is opportunistic rather than provisioned, in that only unprovisioned or unused reclaimable provisioned class-A/B bandwidths are available. As such, the class-C traffic provides no minimum-bandwidth or maximum-jitter guarantees. A weighted fairness algorithm is used to partition class-C traffic among contending stations.

Internal to the MAC, the class-A bandwidth is partitioned into two subclasses: subclass-A0 and subclass-A1. The MAC's client requests claas-A traffic, not one of the internal subclasses. The MAC is provisioned for a total class-A amount, from which it determines how much is subclass-A0 and how much is subclass-A1, based on ring circumference and STQ size. The MAC advertises a class-A provisioning equal to its internal subclass-A0 amount.

[1]MAC delay and jitter are defined to be measured from the point where a packet arrives at the head of a source client add queue until it arrives at the tail of a destination client receive queue.

6.1.2 MAC data paths

Each attach point has one or two transit queues, for saving pass-through traffic that arrives during this stations's transmissions, as shown in Figure 8. There are two types of MAC transit queueing designs: mono-queue and dual-queue (see 6.4 and 6.5). The mono-queue design places all pass-through into a primary transit queue (PTQ); the dual-queue design places class-A traffic into a higher-precedence primary transit queue (PTQ) and other traffic into a lower-precedence secondary transit queue (STQ).

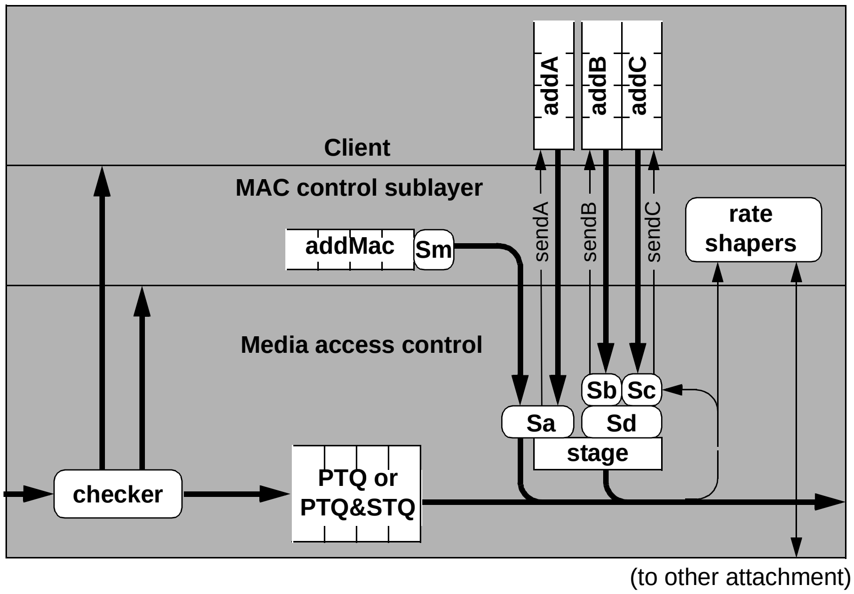

Figure 8—MAC data paths

The client has the option of labeling its frames as class-A, class-B, or class-C. The class-A, class-B, and class-C traffic flow through the Sa, Sb, and Sc shapers, respectively. Additionally, excess class-B traffic is marked out of profile by the MAC and also uses the Sc shaper.

Accepted client traffic is expected to be placed into a stage queue. Sufficient stage-queue storage ensures full-rate transmissions, despite the latencies inherent in signaling rate-limiting flow-control information across the MAC-to-client interface.

Packets from the MAC-control send queue are rate-limited by the ***Sm*** (shaper of MAC), to avoid disruption of provisioned class-A traffic due to unexpected bursts of control traffic. All class-A traffic is rate-limited by the ***Sa*** (shaper of class-A), to avoid having the client exceed its class-A provisioned rates. MAC-control and client traffic both flow through the same shaper, although the shaper effectively only throttles the client's cumulative class-A traffic.

The client's class-B traffic is rate-limited by the ***Sb*** (shaper of class-B), to constrain the client within its class-B provisioned rates. Similarly, the client's class-C traffic is rate-limited by the ***Sc*** (shaper of class-C), to constrain the client within its weighted fair-share use of the residual (unprovisioned or unused provisioned) bandwidth.

The client's class-B and class-C traffic are both shaped by the ***Sd*** (shaper for downstream), to constrain the client for sustaining downstream provisioned class-A rates. Thus, in profile class-B flows are only possible when allowed by shaper *Sb* and *Sd*; similarly, excess class-B and class-C flows are only possible when allowed by shapers *Sc* and *Sd*.

Flow control indications are generated by the MAC, to selectively restrict the tranffic flows from the client. Simple ***sendA*** and ***sendB*** go/no-go indications provide for throttling class-A and class-B traffic, but ***sendC*** communicates a maximum hop-count distance so that class-C traffic can be selectively throttled based on the distance to its target.

NOTE—sendX is logically equivalent to NOT stopX.

6.1.3 Reclamation

Provisioned bandwidth can be reused, or reclaimed, by a lower priority class whenever the reclamation does not effect the jitter bounds of the higher priority class(es) on the local station or on any other station on the ring.

NOTE—Traffic can be sent one hop when there is no traffic in the PTQ, regardless of the provisioning on the link. This is because the maximum delay for any PTQ traffic coming into the local station is 1 MTU, and because the added frame gets stripped at the neighboring station, creating an idle space for any PTQ traffic that needs to enter at the neighboring station.

6.2 Bandwidth provisioning

Editors' Notes (DVJ): *To be removed prior to final publication.*

This subclause contains introductory material and may therefore be moved to the introduction clause when this document is revised in the future.

Editors' Notes (jl): *To be removed prior to final publication.*

The means of provisioning the class-A and class-B traffic, assuring the provisioning is consistent across the ring, and communicating the provisioned amounts to the stations controlling the links over which the traffic transits is left to the OAM&P section drafters.

Note that the communication of provisioning information must be phased so as to not to create a case where transitioning provisioning levels temporarily cause cumulative provisions to exceed the link capacity.

Whether bandwidth is provisioned globally uniform or differently for each link based on spatial awareness of the traffic being provisioned is an implementation decision.

Each station shall have default values for provisioned bandwidths to enable it to plug and play without management configuration.

There are two forms of traffic provisioning, uniform and spatial. Uniform provisioning uses a single rate per class for the entire ring. Spatial provisioning uses independent rates per class for each link. Uniform provisioning makes no distinction between traffic flows based on their hop-count distance. Spatial provisioning differentiates traffic amounts on a per hop-count basis.

Class-A traffic is divided into subclasses, subclass-A0 and subclass-A1. The subclass-A1 is more efficient, but relies on sufficient secondary transit queue depth to buffer class-B and class-C traffic while a congested station is signaling upstream stations to decrease excess traffic. The amount of supportable subclass-A1 traffic in any station is proportional to the size of the secondary transit queue in that station.

All flow-control protocols assume that each station is aware of its class profile as well as the ringlet's cumulative class profile. The simple flow-control protocols (that assume flat class profiles) and the sophisticated flow-control protocols (that allow hop-count dependent class profiles) are fully interoperable.

6.2.1 Mono-queue uniform provisioning

A simple mono-queue station has provisioned levels of class-A and class-B traffic, as illustrated in Figure 9. Within each figure cell, a bandwidth profile is illustrated above the physical ring illustration. In this simplest of provisioning examples, provisioning levels are uniform across the ring, not hop-count dependent. Uniform provisioning makes no distinction between class-A traffic (cell 1) sent from W-to-X and class-A traffic sent from W-to-Z. Similarly, uniform provisioning makes no distinction between class-B traffic (cell 2) traffic sent from W-to-Y and class-B traffic sent from W-to-Z.

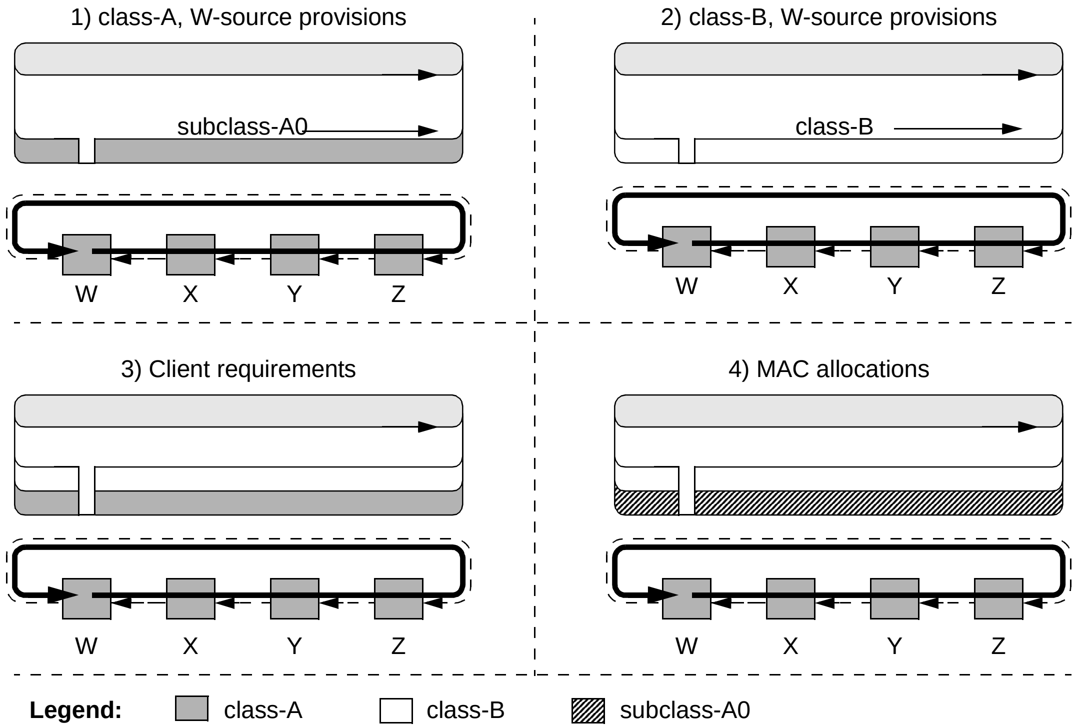

Figure 9—Mono-queue uniform provisioning

For each station, the cumulative profiles form a cumulative bandwidth profile (cell 3) of client-visible bandwidth allocations. Within a mono-queue MAC, the bandwidth advertised for the class-A service (cell 4) consists of the entire class-A provisioned bandwidth, since a second transit queue is necessary to support advertisement of a more efficient subclass-A1 advertisement.

6.2.2 Dual-queue uniform provisioning

A more storage-rich dual-queue station has advertised provisioned levels of subclass-A0, subclass-A1, and class-B traffic, as illustrated in Figure 10. Uniform provisioning makes no distinction between class-A traffic (cell 1) sent from W-to-X and class-A traffic sent from W-to-Z. Similarly, the provisioning (cell 2) of class-B traffic makes no distinction between traffic sent from W-to-Y and traffic sent from W-to-Z.

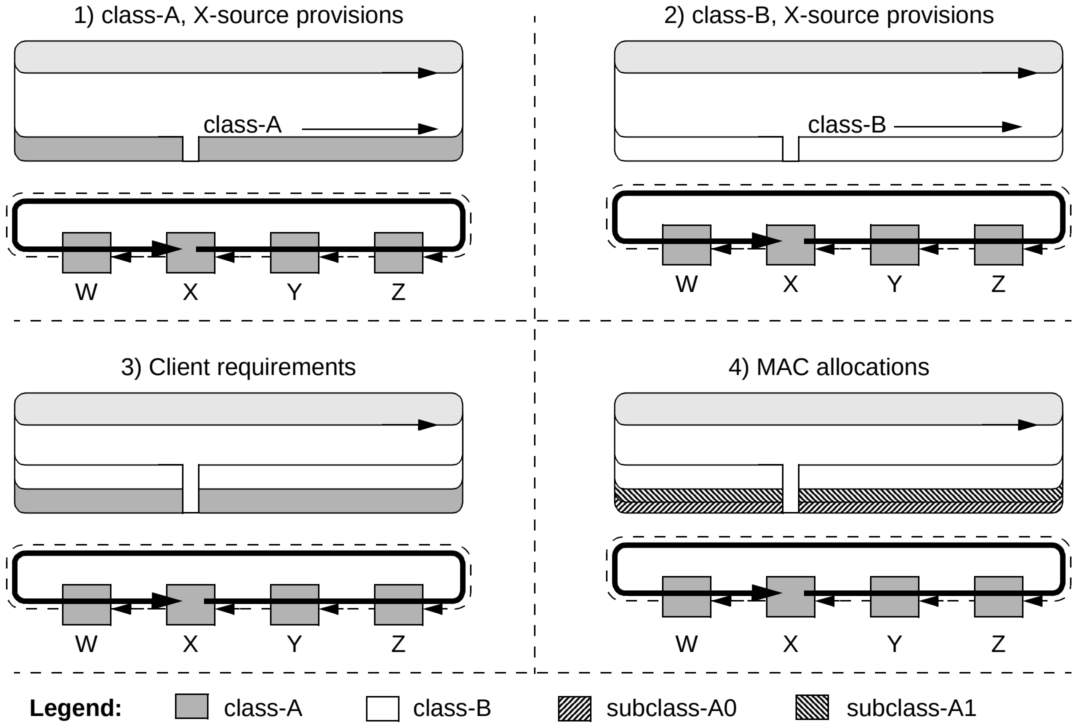

Figure 10—Dual-queue uniform provisioning

> ***Editors' Notes (JL):*** *To be removed prior to final publication.*
>
> These figures need to be reviewed to assure they match the text, or vice versa.

For each station, the sum of class profiles forms a cumulative bandwidth profile (cell 3) of client-visible class-A and class-B bandwidth allocations. The class-A service (cell 4) consists of the allocation of subclass-A0 and class-A1 bandwidths, where the levels of supportable subclass-A1 bandwidths are proportional to the size of the secondary transit queue.

6.2.3 Mono-queue spatial provisioning (informative)

Each station has provisioned levels of class-A and class-B traffic, as illustrated in Figure 11. Spatial provisioning makes a distinction between class-A traffic sent (cell 1) from Y-to-X and class-A traffic (cell 2) sent from Y-to-W. Similarly, spatial provisioning makes a distinction between class-B traffic (cell 3) sent from Y-to-W and class-B traffic (cell 4) sent from Y-to-Z.

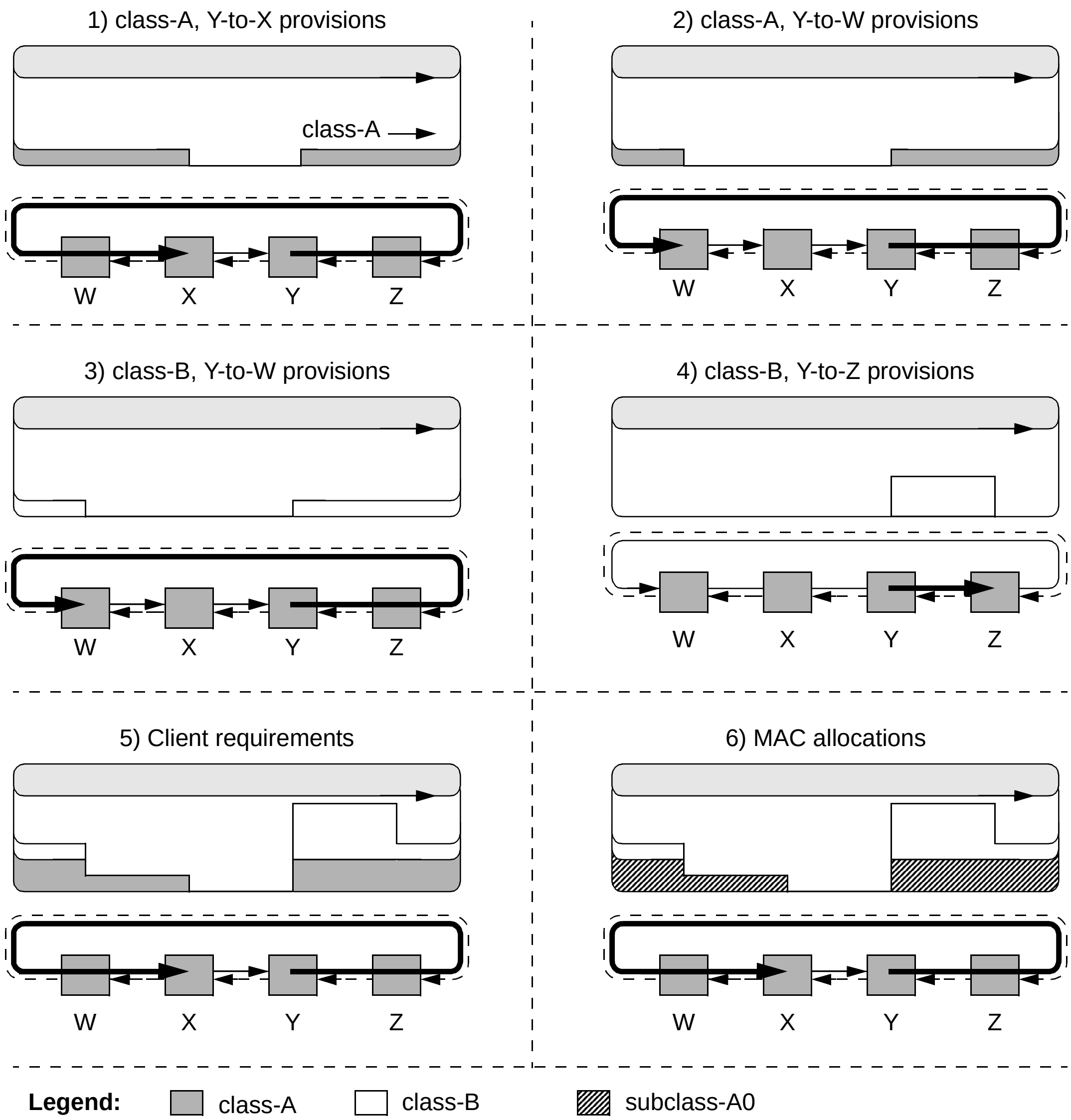

Figure 11—Mono-queue spatial provisioning

For each station, the sum of class profiles forms a cumulative bandwidth profile (cell 3) of client-visible class-A and class-B bandwidth allocations. The class-A service (cell 4) consists of only subclass-A0 bandwidth, since a second transit queue is necessary to support advertisement of a more efficient subclass-A1 advertisement.

6.2.4 Dual-queue spatial provisioning (informative)

Each station has provisioned levels of class-A and class-B traffic, as illustrated in Figure 12. Spatial provisioning of class-A traffic makes a distinction between (cell 1) traffic sent from Z-to-Y and (cell 2) traffic sent from Z-to-X. Similarly, spatial provisioning makes a distinction between class-B traffic (cell 3) sent from Z-to-X and class-B traffic (cell 4) sent from Z-to-W

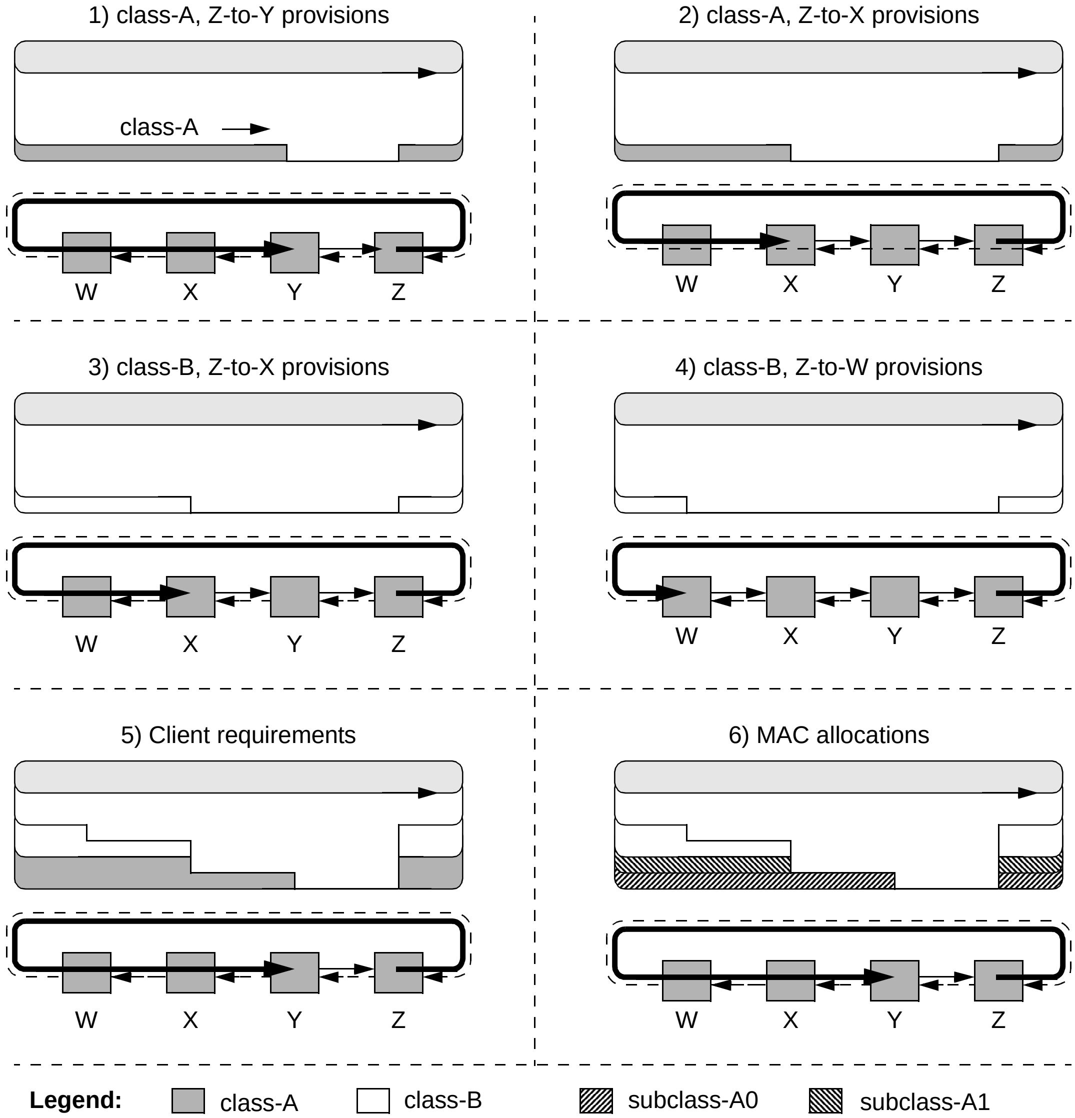

Figure 12—Dual-queue spatial provisioning

For each station, the sum of class profiles forms a cumulative bandwidth profile (cell 3) of client-visible class-A and class-B bandwidth allocations. The class-A service (cell 4) consists of subclass-A0 and subclass-A1 bandwidths, where the level of supportable subclass-A1 bandwidth is proportional to the size of the secondary transit queue.

6.2.5 Cumulative ringlet provisioning (informative)

Each station has its own (cells 1, 2, 3, and 4) cumulative class profile, as illustrated in Figure 13. These can be combined (cell 5) into a ringlet class profile. Within consistent ringlet profiles, the sum of subclass-A0, subclass-A1, and class-B profiles (*provisioned* in cell 5) shall be less than any individual link capacity.

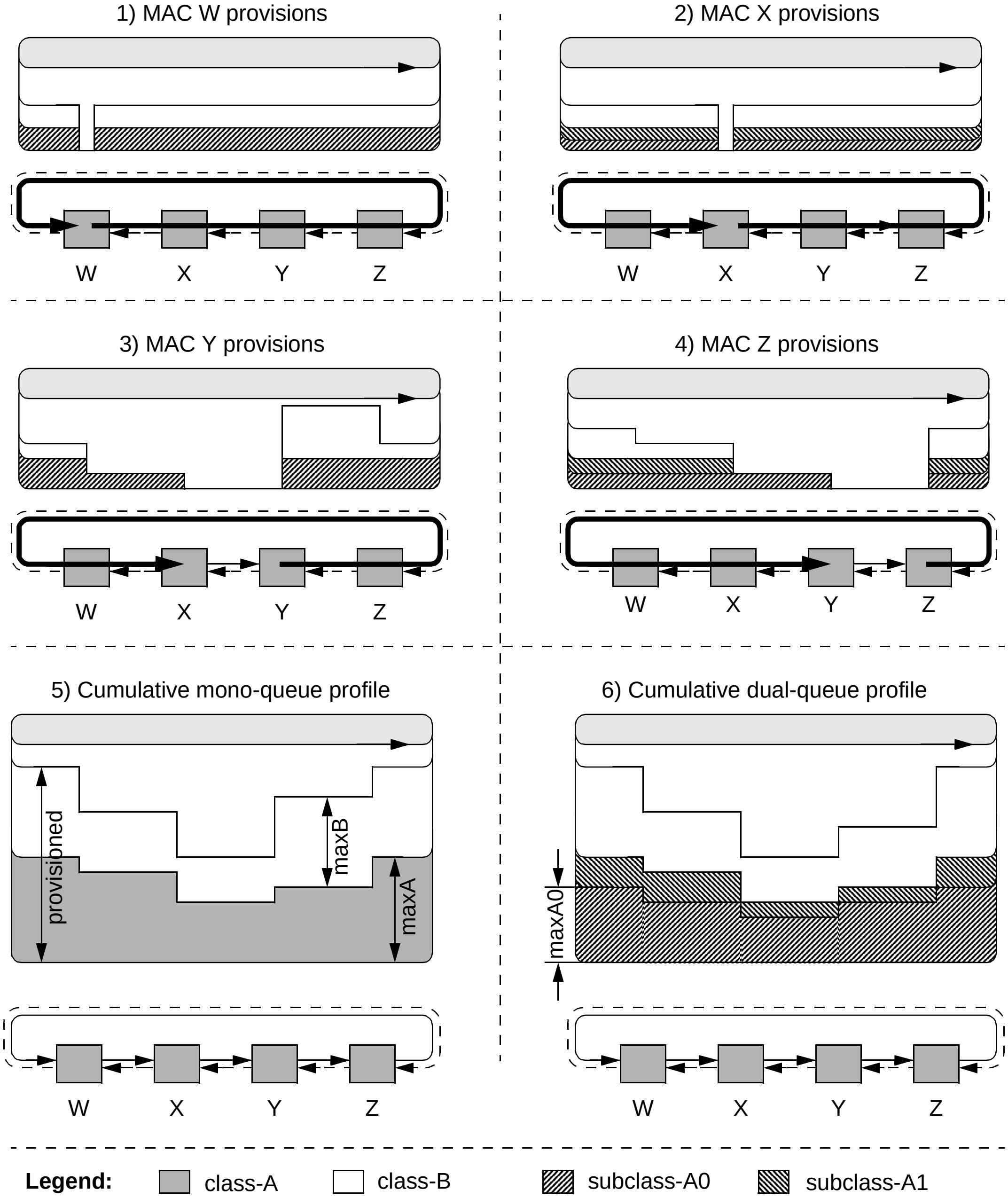

Figure 13—Cumulative station provisions

For scalar policing, the cumulative class profile yields *maxA* and *maxA0*, the worst-case provisioned class-A and subclass-A0 segments respectively. For vector policing, the cumulative class profile provides *rateA[n]* and *rateA0[n]*, the provisioned class-A and subclass-A0 levels on segment *n*, respectively. This information is used to rate-limit each station's class-B and class-C transmissions, with the intent of sustaining downstream class-A transmissions.

During provisioning, the cumulative class profile also yields *maxB*, the worst-case provisioned class-B segment. To ensure interoperability, the sum of *maxA* and *maxB* shall be less than the link capacity.

NOTE—The value of *rateA*+*rateB* equals the maximum value of *rateA[n]*+*rateB[m]*, measured on all links *n* and *m*. The requirement for this sum to be less than the link capacity is more restrictive than the physical mandated restriction that *rate[n]*+*rateB[n]* be less than the link capacity on any link *n*.

6.3 Rate control

6.3.1 Add queue rate policing

Packets from the MAC Control add queue, class-A add queue, and class-B add queue are rate controlled, via the ***sendA*** and ***sendB*** signals, to the class-A and class-B rates provisioned for the station. Fairness eligbile add traffic (Class-C and EIR class-B traffic) is rate controlled, via the ***sendC*** signal, to 2 rates. It is always controlled to a static rate configured for the station. In addition, any class-C traffic passing the congestion point reported in the Type 1 fairness message is also rate controlled to the lesser of the static rate and the dynamic rate provided by the RPR-fa algorithm. The ***sendC*** signal can be used to indicate which of these rate limitations is in effect by including the distance to the congestion point. No congestion point is indicated by a distance of the ring size. In addition to the above, all classes of traffic are effectively rate limited by the transmission selection algorithms described in 6.4.2 and 6.5.2.

6.3.2 Mono-queue rate-shaping

In a mono-queue implementation, the PTQ output is not shaped.

6.3.3 Dual-queue rate-shaping

In a dual-queue implementation, the PTQ output does not need to be shaped. The total outgoing (add plus transit) sum of (subclass-A1 + class-B + class-C) traffic output shall be shaped to meet the class-A requirements of downstream stations.

6.3.4 MAC shapers

Although multiple shapers are used within this standard, the behaviors of all shapers can be characterized by a common algorithm with application-specific parameters. The shaper's *dSize*, *uSize*, and *rate* parameters effect the credit adjustments and limits, as illustrated in Figure 14. The *dSize* and *uSize* values typically represent sizes of a transmitted frame and update intervals respectively.

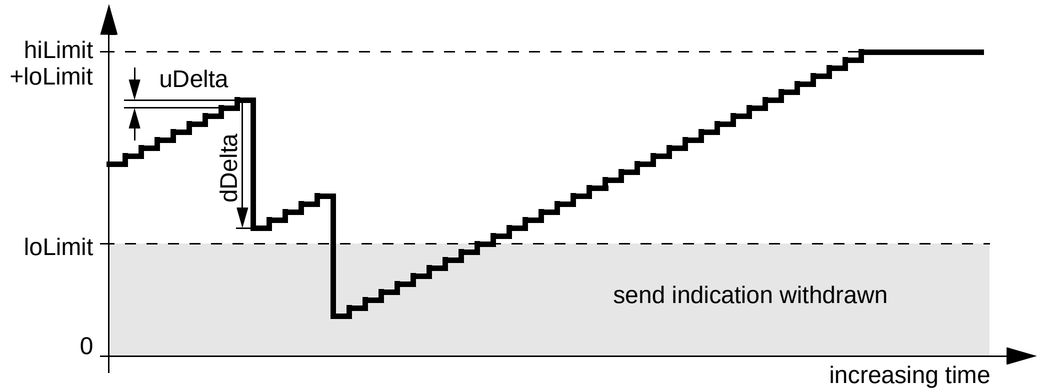

Figure 14—Credit adjustments over time

Crossing below the *loLimit* threshold typically generates a rate-limiting indication, so that offered traffic can stop before reaching the lesser MTU limit, where excessive transmissions are rejected.

The *hiLimit* value limits the positive credits, to avoid overflow and to bound the burst traffic after inactivity intervals. The term *hiLimits* is used to represent the sum of *hiLimit* and *loLimit* values.

6.4 Mono-queue MAC design

A mono-queue MAC uses one queue, the primary transit queue (PTQ), for all transit traffic. The PTQ is at least 2 MTUs (in order to allow almost immediate access for control frames).

6.4.1 Mono-queue MAC data paths

To be able to detect when to transmit and receive packets from the ring, a mono-queue MAC makes use of only one transit queue, as shown in Figure 15. The PTQ has the behavior of a small FIFO.

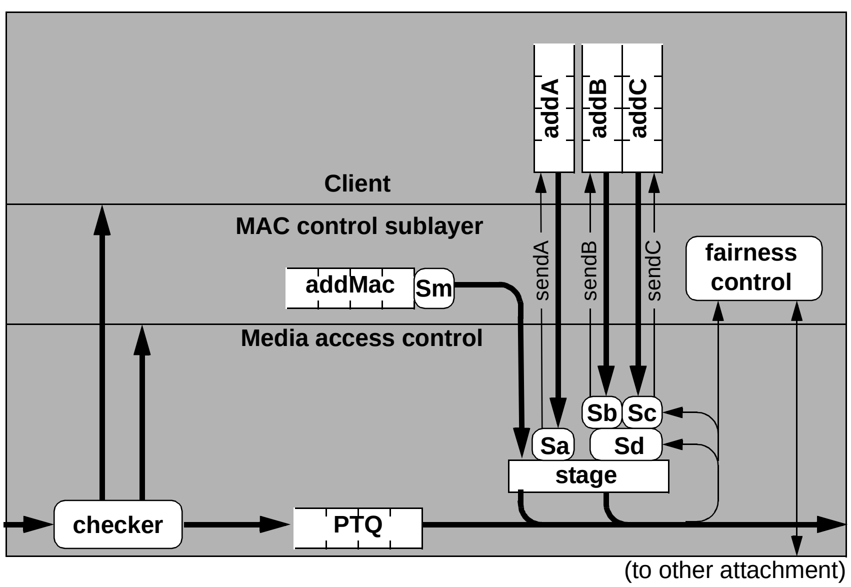

Figure 15—Mono-queue MAC data path

Packets from the MAC-control send queue are rate-limited by shaper ***Sm*** (shaper of MAC). All class-A traffic is rate-limited by shaper ***Sa*** (shaper of class-A).

This standard does not define how to implement the PTQ. However, to meet ordering expectations, a FIFO ordering shall be maintained when entries pass through the PTQ.

6.4.2 Mono-queue transmit selections

The behavior of a mono-queue MAC is described by its transmit-selection protocols (described in this subclause) and shaping functions. The transmit-selection protocol and shaping functions are largely independent, but some coupling (via the internal shaper provided *sendM* and *sendA* signals) is required to ensure conformance of MAC-supplied class-A control frame transmissions.

A mono-queue MAC can transmit data packets from three possible internal queues, as illustrated in Figure 16. An exact definition of the mono-queue transmit-frame selection sequence is specified in table 4, where the rows are evaluated in top-to-bottom order. The intent is to always empty the primary transit queue (PTQ) before frame transmissions, to avoid queue overflow conditions associated with enqueeing additional incoming frames during frame transmissions. .

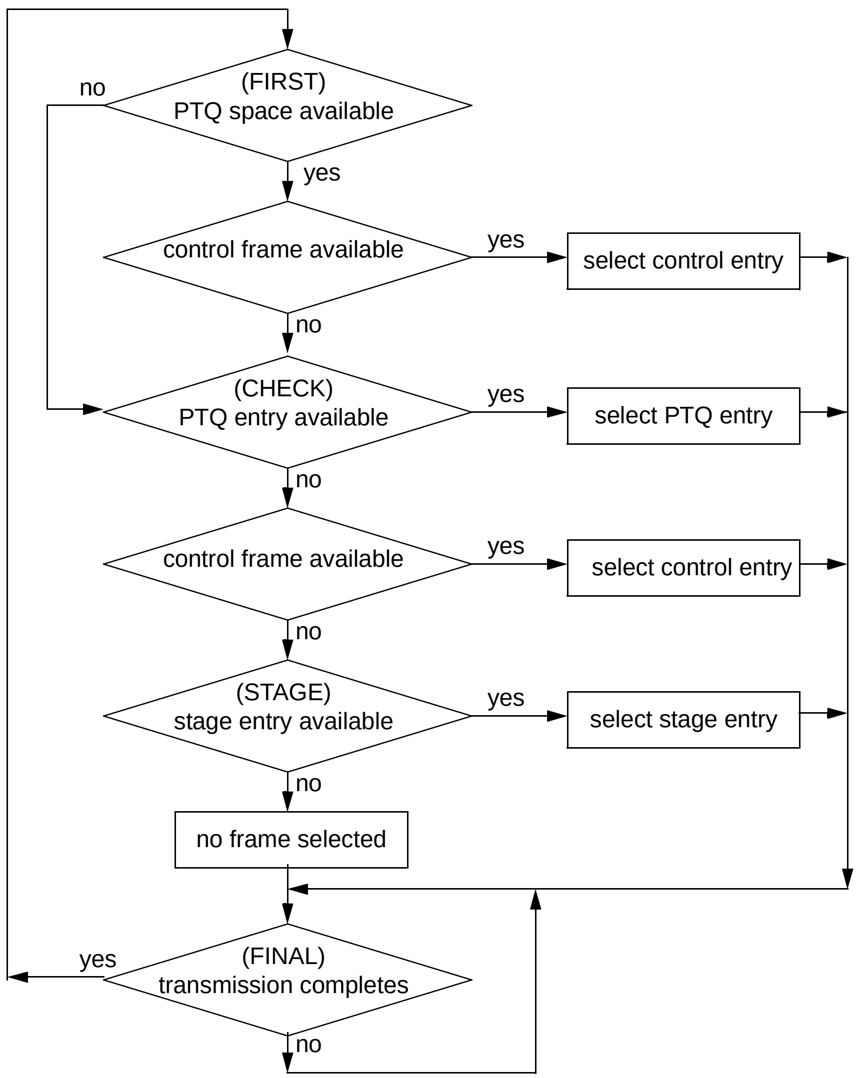

Figure 16—Mono-queue transmit-frame selection

Table 4—Mono-queue transmit-frame selection

Last state		Row	Next state	
state	**condition**		**selection**	**state**
FIRST	sizeOfMacControl > spaceInPTQ	1	—	CHECK
	passM == 0	2		
	—	3	MAC control queue	FINAL
CHECK	CUT_THOUGH && headerInPTQ	4	primary transit queue	FINAL
	STORE_FORWARD && frameInPTQ	5		
	frameInMacControl == 0	6	—	STAGE
	passM == 0	7	—	
	passA == 0	8	—	
	—	9	MAC control queue	FINAL
STAGE	CUT_THOUGH && headerInStage	10	stage queue	FINAL
	STORE_FORWARD && frameInStage	11		
	—	12	no frame selected	
FINAL	selected transfer completes	13	—	FIRST
	—	14	—	FINAL

Row 1: The size of the queued MAC control frame shall be less than the available PTQ space.
Row 2: In the absence of MAC-control transmission credits, other transmission sources are checked.
Row 3: The small MAC-control transmissions have precedence over client transmissions.

Row 4: (Representing the cut-through option)
The primary transit queue is selected when a header is available, to avoid queue overflows.
Row 5: (Representing the store-and-forward option)
The primary transit queue is selected when a frame is available, to avoid queue overflows.

Row 6: In the absence of MAC-control frames, other transmission sources are checked.
Row 7: In the absence of MAC-control transmission credits, other transmission sources are checked.
Row 8: In the absence of class-A transmission credits, other transmission sources are checked.
Row 9: The MAC-control queue transmissions have precedence over client transmissions.

Row 10: (Representing is the cut-through option)
The stage queue header pre-empts STQ transmissions, since provision checks were done previously.

> ***Editors' Notes (JL):*** *To be removed prior to final publication.*
>
> These discussions of STQ are out of place. Need discussion of PTQ instead, matching the figure.

Row 11: (Representing is the store-and-forward option)
A stage-queue frame pre-empts STQ transmissions, since provision checks were done previously.
Row 12: No frame is selected when no frame transmissions are possible.

Row 13: The next selection occurs when the transmission completes.
Row 14: The selected transmission/retransmission continues until completed.

6.5 Dual-queue MAC design

The behavior of a dual-queue MAC is described by its transmit-selection protocols (described in this subclause) and shaping functions. The transmit-selection protocol and shaping functions are largely independent, but coupling (via the internal shaper provided *sendM* and *sendA* signals) is required to ensure conformance of MAC-supplied class-A control frame transmissions. Additional coupling (via the internal shaper provided *sendD* signal) is required to properly sustain downstream class-A transmissions.

A dual-queue MAC uses two transit queues, the primary transit queue (PTQ) for class-A traffic, and the secondary transit queue (STQ) for class-B and class-C traffic. The size of the secondary transit queue is left to the implementations. The dual-queue design is described in following subclauses.

6.5.1 Dual-queue MAC data paths

A dual-queue MAC makes use of two transit queues, as shown in Figure 17. The sizes of the transit queues are left to the implementations, although the primary transit queue has a minimum size of at least 2 MTUs. The size of the secondary transit queue determines its flow-control threshold values.

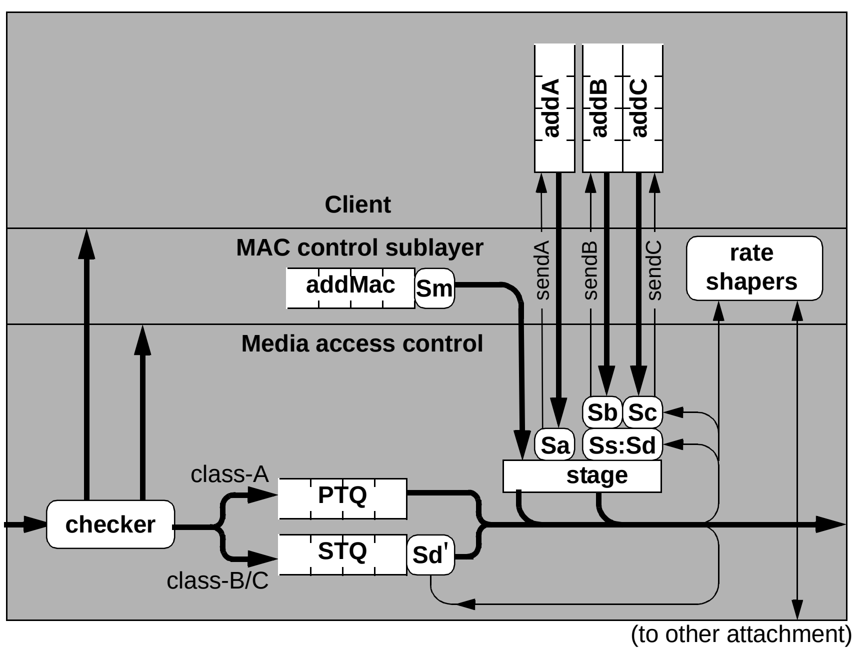

Figure 17—Dual-queue data-path model

Packets from the MAC-control send queue are rate-limited by shaper ***Sm*** (shaper of MAC). All class-A traffic is rate-limited by shaper ***Sa*** (shaper of class-A).

This standard does not define how to implement the PTQ and STQ queues. However, to meet ordering expectations, the following externally visible behaviors shall be supported:

a) PTQ ordering. A FIFO ordering shall be maintained when entries pass through the PTQ.
b) STQ ordering. A FIFO ordering shall be maintained when entries pass through the STQ.
c) Cross ordering. An entry from the STQ shall not be output before a previously received PTQ entry.

6.5.2 Dual-queue transmit selections

A dual-queue MAC can transmit data packets from three possible internal queues, as illustrated in Figure 18 An exact definition of the dual-queue transmit-frame selection sequence is specified in table 5, where the rows are evaluated in top-to-bottom order.

.

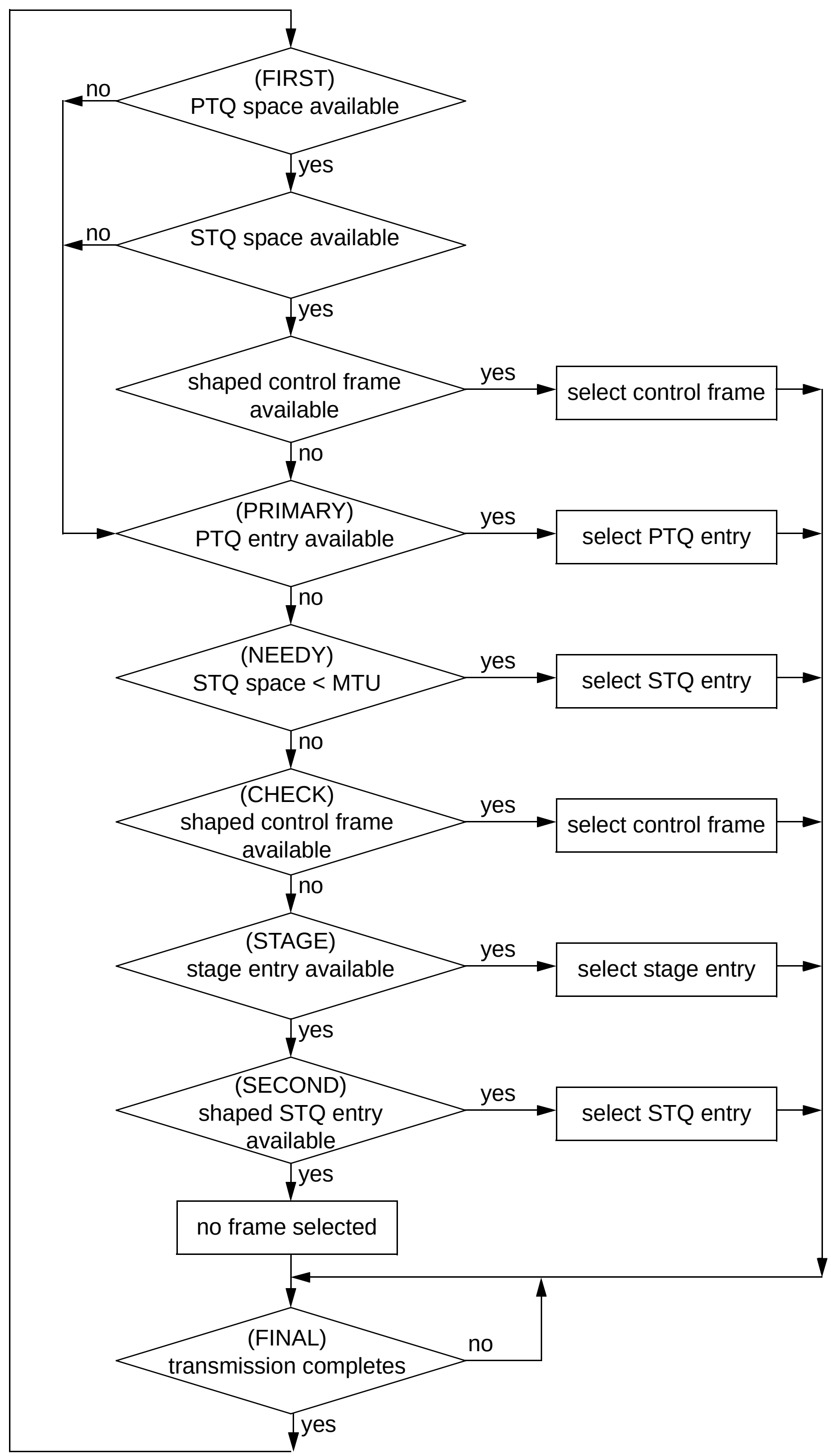

Figure 18—Dual-queue transmit selection

Acceptance into the stage entry is governed by the per class shapers and the fairness algorithm. Fairness eligible traffic is added to the stage only when the station is not congested (as defined in clause 9).

Table 5—Dual-queue transmit-frame selection

Last state		Row	Next state	
state	**condition**		**selection**	**state**
FIRST	sizeOfMacControl > spaceInPTQ	1	—	PRIMARY
	sizeOfMacControl > spaceInSTQ	2		
	passM == 0	3		
	—	4	MAC control queue	FINAL
PRIMARY	CUT_THOUGH && headerInPTQ	5	primary transit queue	FINAL
	STORE_FORWARD && frameInPTQ	6		
	—	7	—	NEEDY
NEEDY	depthSTQ > sizeSTQ–MTU	8	secondary transit queue	FINAL
	—	9	—	CHECK
CHECK	NOT(entryInMacControl)	10	—	STAGE
	passM == 0	11	—	
	passA == 0	12	—	
	—	13	MAC control queue	FINAL
STAGE	CUT_THOUGH && headerInStage	14	stage buffer	FINAL
	STORE_FORWARD && frameInStage	15		
	—	16	—	SECOND
SECOND	passD==0	17	no transmit selection	FINAL
	CUT_THOUGH && headerInSTQ	18	secondary transit queue	
	STORE_FORWARD && frameInSTQ	19		
	—	20	no transmit selection	
FINAL	selected transfer completes	21	—	FIRST
	—	22	—	FINAL

Row 1: Validate that the MAC control frame is less than the available PTQ space.
Row 2: Validate that the MAC control frame is less than the available STQ space.
Row 3: Validate that the MAC control frame remains within its provisioned rate.
Row 4: The small MAC-control transmissions have precedence over client transmissions.

Row 5: (Representing the cut-through option)
The primary transit queue is selected when a header is available, to avoid queue overflows.
Row 6: (Representing the store-and-forward option)
The primary transit queue is selected when a frame is available, to avoid queue overflows.
Row 7: In the absence of primary-transit-queue frames, other transmission sources are checked.

Row 8: The secondary transit queue is emptied when less than one MTU remains free, to avoid overflows.
Row 9: In the absence of secondary-transit-queue overflow threats, other transmission sources are checked.

Row 10: In the absence of MAC-control frames, other transmission sources are checked.
Row 11: In the absence of MAC-control transmission credits, other transmission sources are checked.
Row 12: In the absence of class-A transmission credits, other transmission sources are checked.
Row 13: The MAC-control queue transmissions have precedence over client transmissions.

Row 14: (Representing is the cut-through option)
The stage queue header pre-empts STQ transmissions, since provision checks were done previously.
Row 15: (Representing is the store-and-forward option)
A stage-queue frame pre-empts STQ transmissions, since provision checks were done previously.
Row 16: In the absence of stage-queue frames, other transmission sources are checked.

Row 17: When necessary to sustain downstream class-A traffic, STQ traffic is ignored.
Row 18: (Representing the cut-through option)
A secondary transit-queue header is serviced when available.
Row 19: (Representing the store-and-forward option)
A secondary transit-queue frame is serviced when available.
Row 20: The secondary transit queue is ignored when no frame is available.

Row 21: The next selection occurs after the transmission completes.
Row 22: The selected transmission/retransmission continues until completed.

6.6 Receive operation

The receive operation is shown in Figure 29. A corrupted frame is immediately rejected, a remaining accept frame is copied to the client or MAC control, the *timeToLive* field is decremented, and a passing-through frame is saved into the primary or secondary transmit queue. The mono-queue and dual-queue operations are both defined; the queue-design dependent portions are shaded in grey.

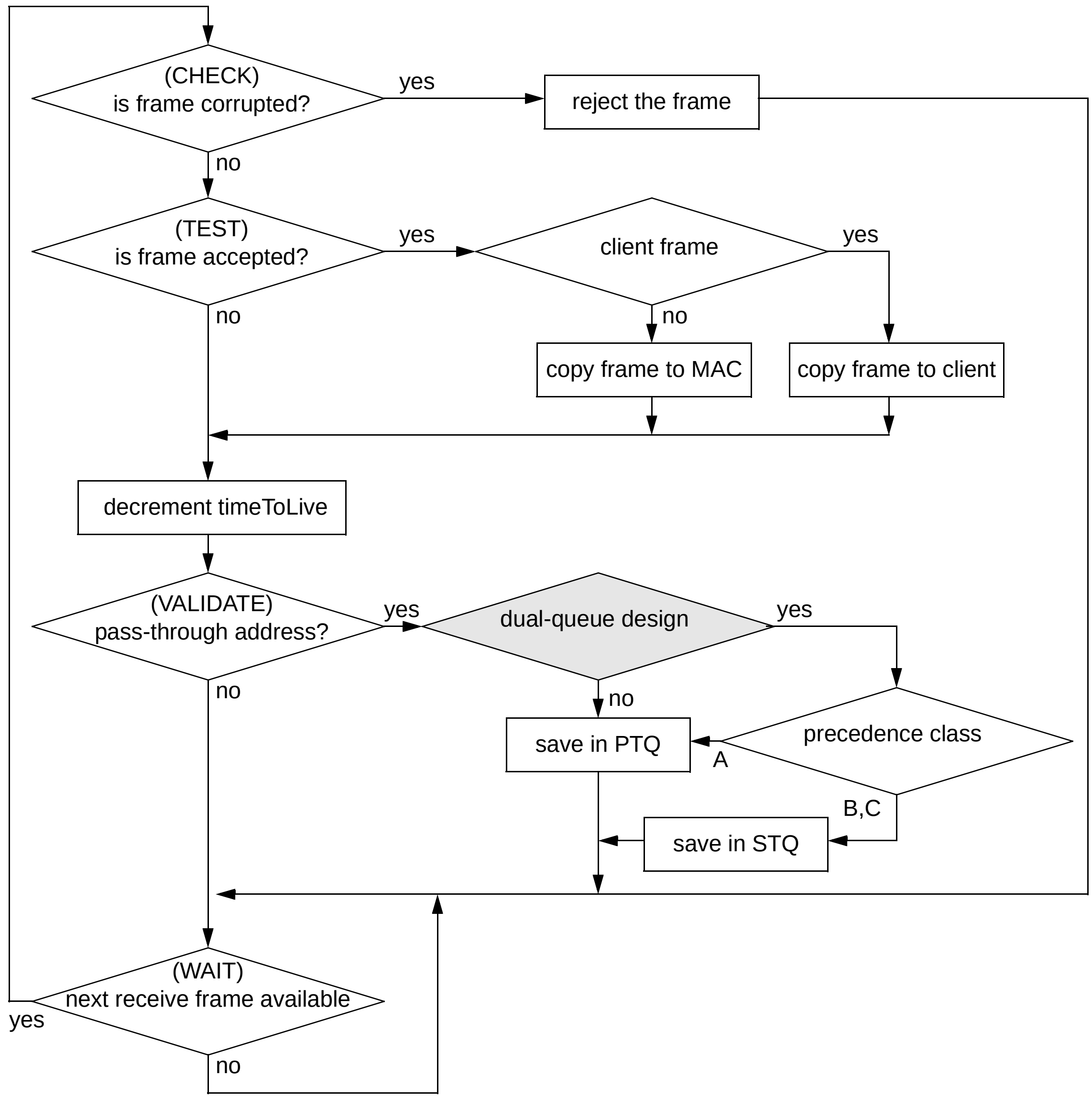

Figure 29—Receive processing flowchart

The specific frame processing steps are specified in Table 6. The mono-queue and dual-queue operations are both defined; the queue-design dependent portions are shaded in grey.

Table 6—Receive processing states

Last state		Row	Next state	
state	**condition**		**action**	**state**
CHECK1	headerCRC == CRC(header)	1	—	CHECK2
	—	2	—	CHECK1
CHECK2	dataCRC==CRC(data)	3	—	CHECK3
	dataCRC== ~CRC(data)	4	—	
	—	5	dataCRC= ~CRC(data) errorCount+= 1	
CHECK3	timeToLive==0	6	—	START
	MULTICAST(sourceMacAddress)	7		
	header.wrap==0&&header.ringID != myRingID	8		
	—	9	—	TEST
TEST	mode.permiscous==1	10	—	COPY
	header.ringID != myRingID	11	—	SENDING
	destinationMacAddress==myMacAddress	12	—	COPY
	sourceMacAddress==myMacAddress	13	—	SENDING
	MULTICAST(destinationMacAddress)	14	—	COPY
	frameHeader.flooding==1	15		
	—	16	—	SENDING
COPY	DataFrame(frameHeader)	17	CopyToClient()	SENDING
	—	18	CopyToMac()	
SENDING	—	19	timeToLive–= 1	VALIDATE
VALIDATE	timeToLive==0	20	—	START
	header.wrap==0&&myState.wrapped	21		
	header.ringID != myState.ringID	22	RecomputeHeaderCrc()	SAVE
	destinationMacAddress=myMacAddress	23	—	START
	sourceMacAddress=myMacAddress	24		
	mono transit-queue implementation	25	CopyToPTQ()	START
	frameHeader.class==CLASS_A	26		
	—	27	CopyToSTQ()	
WAIT	next receive frame available	28	—	CHECK1
	—	29	—	WAIT

Row 1: If the header CRC differs from its expected value, the potentially corrupt frame is rejected.
Row 2: Otherwise, frame processing continues in the normal fashion.

Row 3: A good data CRC is processed normally.
Row 4: A stomped data CRC is flagged and processed normally.
Row 5: A stomped data CRC is counted, flagged, and processed normally.

Row 6: Expired frames are immediately rejected.
Row 7: Multicast source addresses are illegal and therefore rejected.
Row 8: Steer-only frames are rejected when apparently wrapped.
Row 9: Otherwise, normal frame processing continues.

Row 11: Wrapped frames are not copied when returning on their opposing run.
Row 14: Multicast frames are accepted by all but their final destination station.
Row 12: Unicast frames are accepted at their destination station.
Row 15: Flooded frames are accepted by intermediate stations.
Row 16: Otherwise, frames are not accepted.

Row 17: Accepted data frames are passed to the client.
Row 18: Otherwise, accepted frames are passed to the control portion of the MAC.

Row 19: The *timeToLive* field is always updated on passing-through frames.

Row 20: Expired frames are rejected.
Row 21: Steer-only frames are rejected when the wrapping station is reached.
Row 22: Wrapped frames are not stripped when returning on their opposing run.
Row 23: Frames are rejected after reaching their destination.
Row 24: Frames are rejected after returning to their source.

Row 25: (This row is only applicable to the dual-queue design option.)
In mono-queue designs, all remaining frames are placed into the primary transit queue.
Row 26: The remaining class-A frames are placed into the primary transit queue.
Row 27: The remaining class-B and class-C frames are placed into the secondary transit queue.

Editors' Notes (JL): *To be removed prior to final publication.* These row values need to be checked to make sure they align correctly with the figure.

6.7 Wrappable data paths

There are 2 methods of implementing wrapping, each with different effects.

> ***Editors' Notes (JL):*** *To be removed prior to final publication.*
>
> Which of the wrapping methods is standardized, or if both are standardized and either is optional is still under discussion. The implications on the client need to be described.

6.7.1 Center wrap

Each wrapping capable station has wrappable data paths that allow frames to loop-back to the opposing ringlet after link failures are detected, as shown in Figure 20. The wrapping mode has no effect on the behavior of the attachment points, but steering-only frames are rejected when passing into a wrong-run attachment point. This will allow a station to be connected to the ring when there is a single attachment failure.

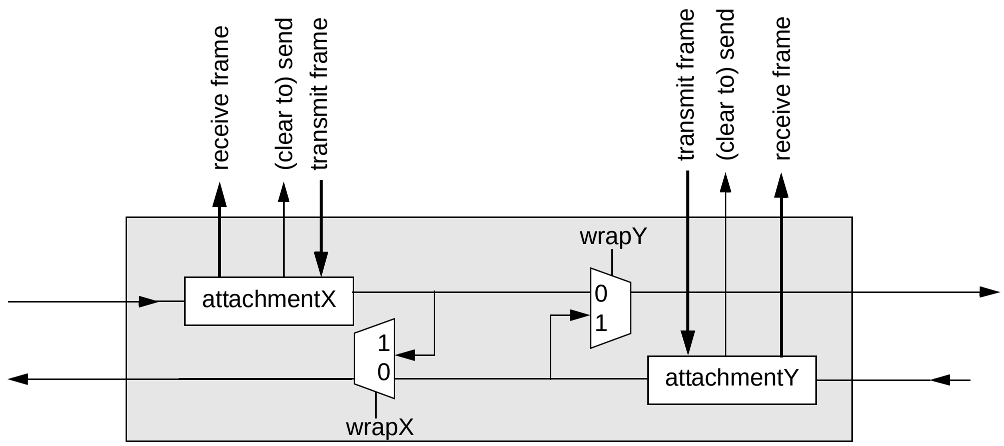

Figure 19—Wrappable data paths

6.7.2 Edge wrap

Each wrapping capable station has wrappable data paths that allow frames to loop-back to the opposing ringlet after link failures are detected, as shown in Figure 20. The wrapping mode has no effect on the behavior of the attachment points, but steering-only frames are rejected when passing into a wrong-run attachment point.

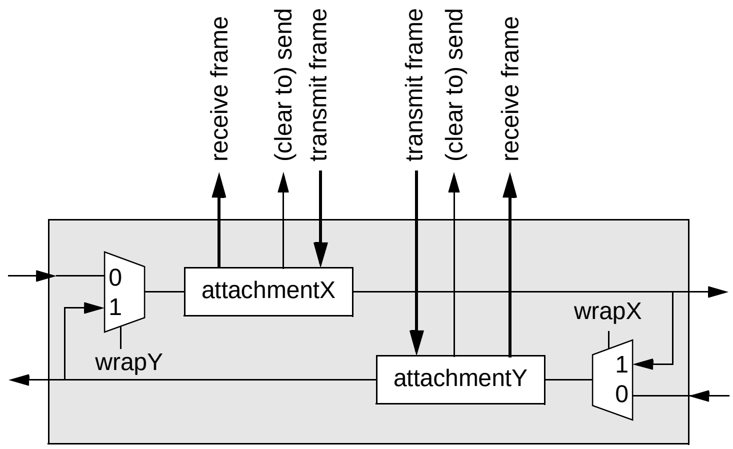

Figure 20—Wrappable data paths

> ***Editors' Notes (JL):*** *To be removed prior to final publication.*
>
> Whether the wrapped traffic goes through the opposite attachment or bypasses it is under discussion.

Annex A: Spatially aware class-A/class-B shaping (informative)

Editors' Notes (DVJ): *To be removed prior to final publication.*

This vector shapers information is preliminary and is intended to be moved to an informative annex.

A.1 Reclamation

Provisioned bandwidth can be reused, or reclaimed, by a lower priority class when the reclamation does not effect the delay and jitter bounds of the higher priority class(es) on the local station or on any other station on the ring.

Editors' Notes (jl): *To be removed prior to final publication.*

The following are informative notes that explain possible methods a station can use to make a local decision on how to reclaim provisioned bandwidth in a such a way that it does not violate the delay and jitter bound properties of other traffic. I'm not sure how to format informative notes.

Traffic can be sent more than one hop when the local station is VDQ aware and knows that there is no provisioning that would be violated on the links subsequent to the first link, and when there is no traffic in the PTQ, regardless of any provisioning on the first link. This is because the maximum delay for any PTQ traffic coming into Station 1 is 1 MTU, because no provisioning would be violated on the links subsequent to the first link, and because the add frame gets stripped at the destination station, creating an idle space for any PTQ traffic that needs to enter at the destination station.

Traffic can be sent to any destination station when there is no traffic in the PTQ, if the local station has a STQ sufficiently large to hold the maximum queueable traffic, and if the STQ is below its low threshold. This is because the maximum delay for any PTQ traffic coming into the local station is 1 MTU, and because any STQ traffic coming into the local station can be buffered long enough to advise the sender through a fairness message to decrease its rate.

A.2 Vector shaping overview

A.2.1 Vector flow-control

A primary objectives of RPR is to support spatial reuse in the RPR ring, e.g. to maximize the link utilization for frame flows with arbitrary (source, destination) pairs. For any particular station, this translates into a desire to maximize the link utilization for frame flows based on the frame destination.

If the MAC does not allow an independent access rate per destination, the MAC typically has to set a low access rate low, to satisfy the bandwidth allocated to one congested destination, which severely limits the access rates to other uncongested destinations.

Another potential performance limitation is that associated with head of line (HoL) blocking. If the MAC client uses a single FIFO to buffer frames awaiting access on to the ringlet, a frame that is destined to a congested destination (but is at the head of the FIFO) may block transmissions to other uncongested destinations.

This standard addresses these problems through vector shapers, where each element of the shaper corresponds to a single link or contiguous set of links (called a segment) on the ringlet. This allows the client to support multiple output queues, where each queue represents one segment.

To fully utilize the spatial properties of the ringlet, the MAC and client need to support independent shapers for each ring segment. Furthermore, the class-A and class-B flow control information supplied by the MAC needs to specify the hop-count distance to the destination.

A.2.1.1 Head-of-line blocking limitations of uniform provisioning

In the case of RPR, head-of-line blocking involves blocking of a nearby transmission due to congestion at a distant station, as illustrated in the top half of Figure A.1. In this example, station *R* has two aggregate flows, one destined to station *V* and another destined to station *S*. If any link between station *S* and station *U* is congested, the later arriving station *R*-to-*S* traffic will also be delayed, as though the station *R*-to-*S* link were congested.

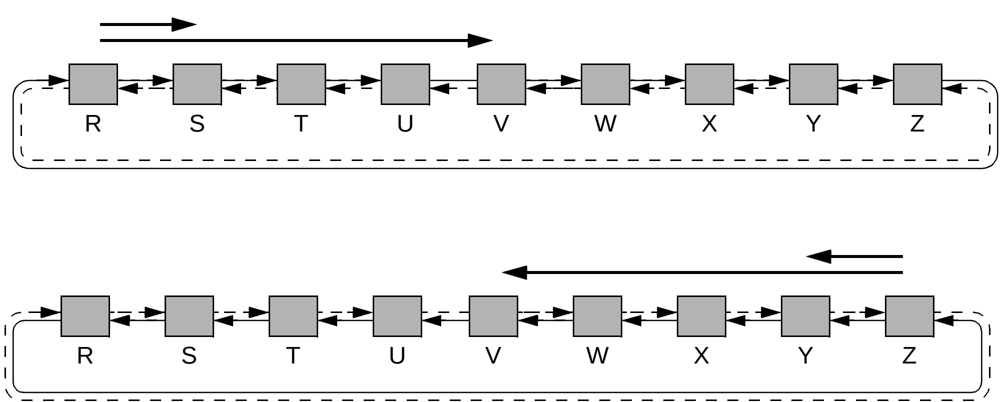

Figure A.1—Head-of-line blocking limitations of uniform provisioning

In this example, station *Z* could have two aggregate flows, one destined to station *V* and the other destined to station *Y*. If any link between station *X* and station *V* is congested, the later arriving station *Z*-to-*Y* traffic would also be delayed, as though the station *Z*-to-*Y* link were congested

When packets destined for Node 5 reach the head of the queue in Figure 2, the packets destined for Node 2 will be slowed or blocked by the packet(s) destined for Node 5. This is head of line blocking. This problem has been long addressed in high-speed crossbar switches.

A.2.1.2 Central-hub limitations of uniform provisioning

A simple example of the potential advantages of spatial provisioning is provided by a ringlet which supports central-hub accesses, as illustrated in Figure A.2. In this example, a large portion of the accesses on ringlet-0 flow into or out-of the central hub *V*. The common accesses on ringlet-0 and ringlet-1 are illustrated in the top and bottom halves of Figure A.2 respectively.

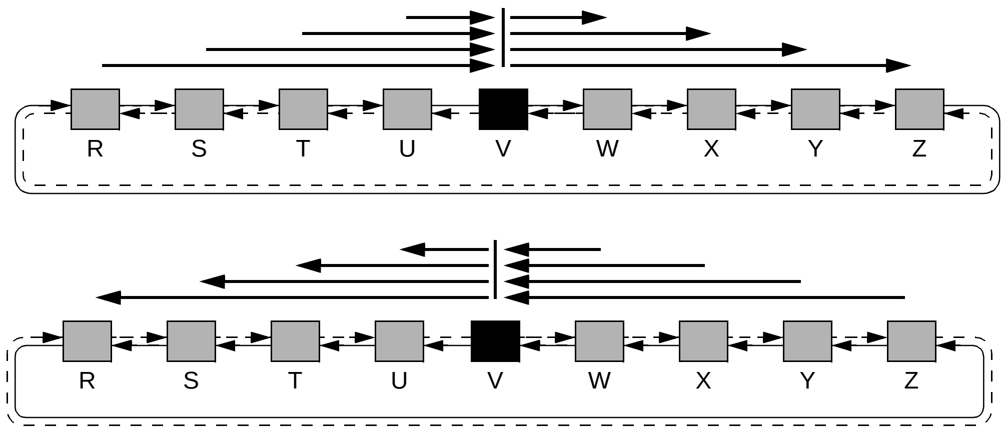

Figure A.2—Central-hub limitations of uniform provisioning

With uniform provisioning, the cumulative bandwidths of all eight flows on ringlet-0, as well as all eight flows on ringlet-1 are constrained to be less than any link capacity. With spatial provisioning, the cumulative bandwidths of any four flows (ringlet-0-leftside, ringlet-0-rightside, ringlet-1-leftside, or ringlet1-rightside) are constrained to be less than any link capacity. Thus, twice the levels of provisioned bandwidth can be supported.

A.2.2 Vector flow-control model

Using class-A an example, the hop-count information allows a single logical path (illustrated in the left of Figure A.3) to behave as multiple virtual paths (illustrated in the right of Figure A.3), each with their own go/no-go indication. The *sendA* indication indicates the number of hops before the first shaper-limited per-destination flow. Only those virtual queues before that congested location are enabled, the virtual queues after than congested location are disabled.

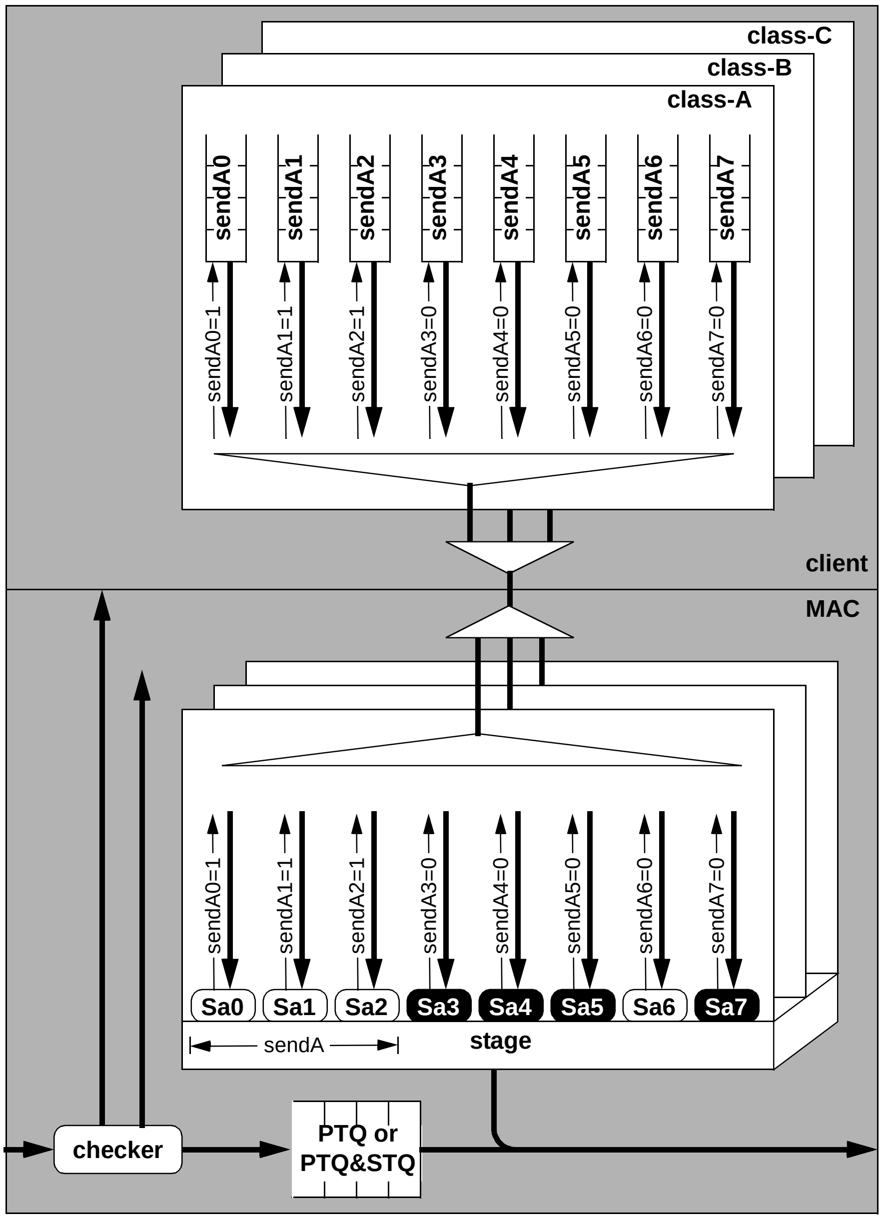

Figure A.3—Vector flow-control model

To generate the send information within the MAC, arrays of class-A, class-B, and class-D shapers are required, one for each of the relevant segments. The MAC-generated sendA indication represents the number of passing shapers (shaded white) before the first of the blocking shapers (shaded black). These shapers are described in the following subclauses.

A.3 Vector shapers

A.3.1 Class-A vector shaper

A.3.1.1 Class-A vector shaper parameters

The class-A vector shaper limits class-A frame transmissions to their provisioned levels on a per-segment basis, based on the parameters listed in table A.1. Each of the flow-control outputs *sendA[n]* shall be set to one if and only if *creditA[n]>=loLimitA*.

Table A.1—Class-A vector shaper parameters

Values		Hop	Amounts		Row	hiLimit		loLimit	
credits	**rate**		**dSize**	**uSize**		**Name**	**Value**	**Name**	**Value**
creditA[n]	rateA[n]	n<=d	Ca,Ma	Ta,Tb,Tc, Cb,Cc,N	1	hiLimitA	limitA	loLimitA	MTU
		n>d	–	Ta,Tb,Tc,Ma, Ca,Cb,Cc,N	2				

#define limitA (TBD)

Row 1: This row refers to intermediate links between the source and destination stations. As indicated by the dSize column, the *creditB[n]* credits decrease by the scaled sizes of client-supplied class-A and MAC-supplied class-A frame transmissions. As indicated by the uSize column, these credits increase by the scaled sizes of other (transit-supplied class-A, transit-supplied class-B, transit-supplied class-C, client-supplied class-B, client-supplied classC, and null) frames.

Row 2: This row refers to following links destination and source stations. As indicated by the uSize column, the *creditB[n]* credits increase by the scaled sizes of all (transit-supplied class-A, transit-supplied class-B, transit-supplied class-C, MAC-supplied class-A, client-supplied class-A, client-supplied class-B, client-supplied classC, and null) frames.

A.3.1.2 Class-A vector shaper resets

TBD

A.3.1.3 Spatial sendA generation

The class-A vector shaper-output indication, *passA*, identifies the hop-count distance over which frames can be transmitted, as illustrated in figure C.2. An exact definition of the *passA* generation is specified in table A.2, where rows are evaluated in top-to-bottom order.

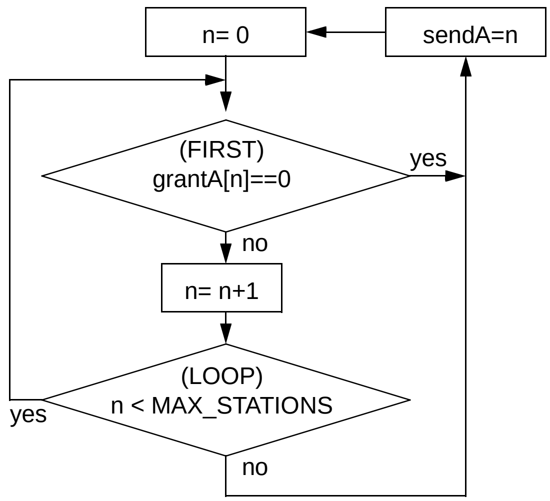

Figure C.2—Vector passA generation

Table A.2—Vector passA generation

Last state	Condition	Row	Action	Next state
FIRST	grantA[n]	1	n= n+1	FINAL
	—	2	passA= n, n= 0	FIRST
FINAL	n < MAX_STATIONS	3	—	FIRST
	—	4	passA= n, n= 0	

Row 1: Checking continues beyond the successful hop-count distances.
Row 2: Checking terminates when insufficient vector credit-A shaper credits are discovered.
Row 3: Checking continues up to the maximum hop-count distance.
Row 4: The range value is set to the checked hop-count distance and the comparisons continue.

A.3.2 Class-B vector shaper

A.3.2.1 Class-B vector shaper parameters

The class-B vector shaper limits class-B frame transmissions to their provisioned levels on a per-segment basis, based on the parameters listed in table A.3 Each of the flow-control outputs *sendB[n]* shall be set to one if and only if *creditB[n]>=loLimitB*.

Table A.3—Class-B vector shaper parameters

Values		Hop	Amounts		Row	hiLimit		loLimit	
credits	**rate**		**dSize**	**uSize**		**Name**	**Value**	**Name**	**Value**
creditB[n]	rateB[n]	n<=d	Cb	Ta,Tb,Tc,Ma, Ca,Cc,N	1	hiLimitB	limitB	loLimitB	MTU
		n>d	–	Ta,Tb,Tc,Ma, Ca,Cb,Cc,N	2				

#define limitB (TBD)

Row 1: This row refers to intermediate links, between the source and destination stations. As indicated by the dSize column, the *creditB[n]* credits decrease by the scaled sizes of client-supplied class-B frame transmissions; they increase by the scaled sizes of other (transit-supplied class-A, transit-supplied class-B, transit-supplied class-C, MAC-supplied class-A, client-supplied class-B, client-supplied classC, and null) frames.

Row 2: This row refers to following links, between the destination and source stations. As indicated by the uSize column, the *creditB[n]* credits increase by the scaled sizes of all (transit-supplied class-A, transit-supplied class-B, transit-supplied class-C, MAC-supplied class-A, client-supplied class-A, client-supplied class-B, client-supplied classC, and null) frames.

A.3.2.2 Class-B vector shaper resets

TBD.

A.3.2.3 Spatial sendB generation

The class-B vector shaper-output indication, *passB*, identifies the hop-count distance over which frames can be transmitted, as illustrated in figure A.4. An exact definition of the *passB* generation is specified in table A.4, where rows are evaluated in top-to-bottom order.

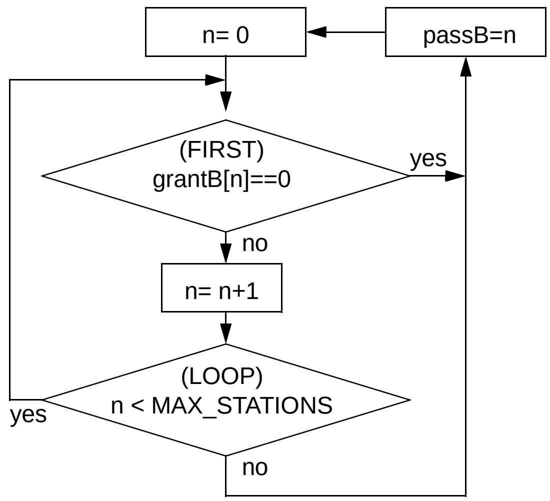

Figure A.4—Vector passB generation

Table A.4—Vector passB generation

Last state	Condition	Row	Action	Next state
FIRST	grantB[n]	1	n= n+1	FINAL
	—	2	passB= n, n= 0	FIRST
FINAL	n < MAX_STATIONS	3	—	FIRST
	—	4	passB= n, n= 0	

Row 1: Checking continues beyond the successful hop-count distances.
Row 2: Checking terminates when insufficient vector credit-B shaper credits are discovered.
Row 3: Checking continues up to the maximum hop-count distance.
Row 4: The range value is set to the checked hop-count distance and the comparisons continue.

A.3.3 Downstream vector shaper

A.3.3.1 Downstream vector shaper parameters

The downstream vector shaper limits class-B and class-C frame transmissions to their provisioned levels on a per-segment basis, based on the parameters listed in table A.5. Each of the flow-control outputs *sendD[n]* shall be set to one if and only if *creditD[n]>=loLimitD*.

Table A.5—Downstream vector shaper parameters

Values		Hop	Amounts		Row	hiLimit		loLimit	
credits	**rate**		**dSize**	**uSize**		**Name**	**Value**	**Name**	**Value**
creditD[n]	rateD[n]	n<=d	Ca,Ta,Ma,N	Tb,Tc, Cb,Cc,N	1	hiLimitD	limitA	loLimitD	MTU
		n>d	–	Ta,Tb,Tc,Ma, Ca,Cb,Cc,N	2				

#define limitA (TBD)
#define limitB (TBD)

Row 1: This row refers to intermediate links between the source and destination stations. As indicated by the dSize column, the *creditB[n]* credits decrease by the scaled sizes of client-supplied class-A, MAC-supplied class-A, and transit-supplied class-A frame transmissions; they increase by the scaled sizes of other (transit-supplied class-B, transit-supplied class-C, client-supplied class-B, client-supplied classC, and null) frames.

Row 2: This row refers to following links, between the destination and source stations. As indicated by the uSize column, the *creditB[n]* credits increase by the scaled sizes of all (transit-supplied class-A, transit-supplied class-B, transit-supplied class-C, MAC-supplied class-A, client-supplied class-A, client-supplied class-B, client-supplied classC, and nonexistent) frames.

A.3.3.2 Downstream vector shaper resets

TBD.

A.3.3.3 Vector sendD generation

The downstream vector shaper-output indication, *passD*, identifies the hop-count distance over which frames can be transmitted, as illustrated in figure A.4. An exact definition of the *passD* generation is specified in table A.6, where rows are evaluated in top-to-bottom order.

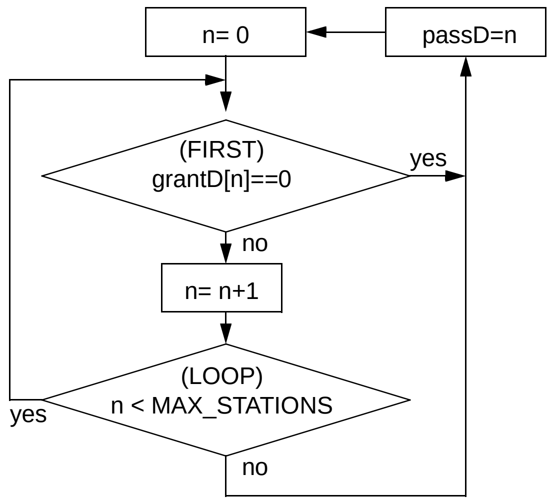

Figure A.4—Vector passD generation

Table A.6—Vector passD generation

Last state	Condition	Row	Action	Next state
FIRST	grantD[n]	1	n= n+1	FINAL
	—	2	passD= n, n= 0	FIRST
FINAL	n < MAX_STATIONS	3	—	FIRST
	—	4	passD= n, n= 0	

Row 1: Checking continues beyond the successful hop-count distances.
Row 2: Checking terminates when insufficient vector downstream shaper credits are discovered.
Row 3: Checking continues up to the maximum hop-count distance.
Row 4: The range value is set to the checked hop-count distance and the comparisons continue.

A.3.4 Vector sendA reception

The reception of client-to-MAC class-A transmissions can result in frame rejects, if the client ignores *sendA* flow-control requests, as illustrated in figure A.2. An exact definition of the class-A reception processing is specified in table A.2, where rows are evaluated in top-to-bottom order.

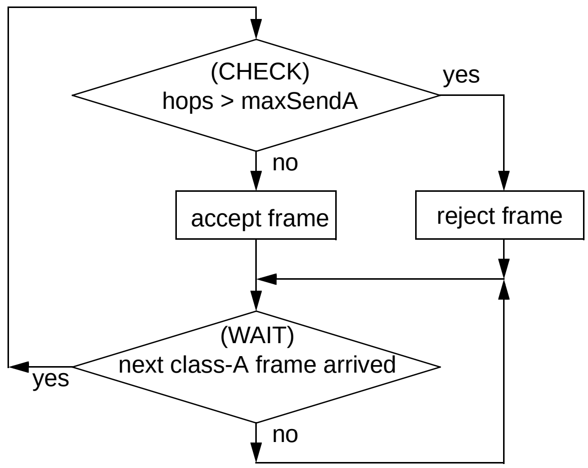

Figure A.2—Vector sendA reception

Table A.7—Vector sendA reception

Last state	Condition	Row	Action	Next state
CHECK	hops > maxSendA	1	reject frame	WAIT
	—	2	accept frame	
WAIT	next class-A client-supplied frame arrives	3	—	CHECK
	—	4	—	WAIT

Row 1: Frames are sometimes rejected if the client ignores flow-control indications.
(The *maxSendA* indication is a maximum of recent *sendA* indications, see A.4.)
Row 2: Frames are always accepted if the client ignores flow-control indications.

Row 3: Row 4: Wait for the next client-supplied class-A frame reception.

A.3.5 Vector class-B flow control

A.3.5.1 Vector sendB generation

The class-B flow-control signal, *sendB*, identifies the hop-count distance over which frames can be transmitted, as illustrated in figure A.4. An exact definition of the *sendB* generation is specified in table A.8, where rows are evaluated in top-to-bottom order.

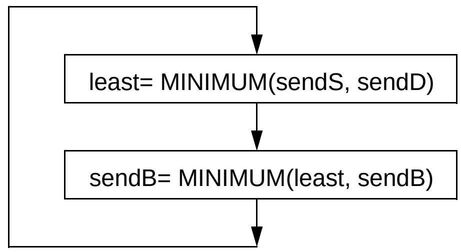

Figure A.4—Vector sendB generation

Table A.8—Vector sendB generation

Last state	Condition	Row	Action	Next state
FIRST	—	1	least= MINIMUM(sendS, sendD)	FINAL
FINAL	—	2	sendB= MINIMUM(least,sendB)	FIRST

Row 1: The allowable distance is no more than the transit-shaper and downstream-shaper restrictions.
Row 2: The allowable distance is no more than the class-B shaper restriction.

A.3.5.2 Vector class-B reception

The reception of client-to-MAC class-B transmissions can result in frame rejects, if the client ignores *sendA* flow-control requests, as illustrated in figure A.5. An exact definition of the class-B reception processing is specified in table A.9, where rows are evaluated in top-to-bottom order.

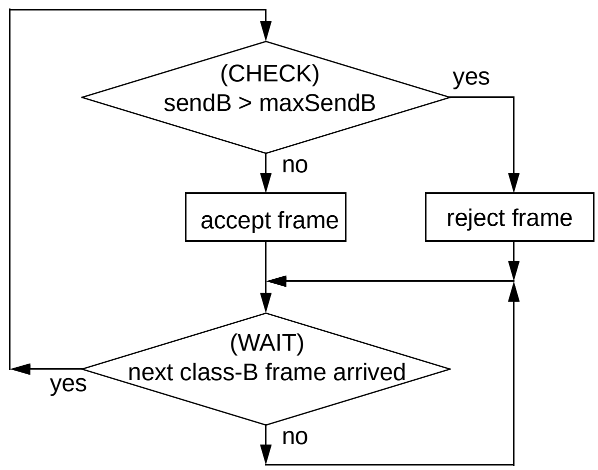

Figure A.5—Vector sendB reception

Table A.9—Vector class-B reception

Last state	Condition	Row	Action	Next state
CHECK	sendB > maxSendB	1	reject frame	WAIT
	—	2	accept frame	
WAIT	next class-B client-supplied frame arrives	3	—	CHECK
	—	4	—	WAIT

Row 1: Frames can sometimes be rejected if the client ignores flow-control indications.
(The *maxSendB* indication is a maximum of recent sendB indications, see A.4.)
Row 2: Frames are always accepted if the client obeys flow-control indications.

Row 3: Wait for the next client-supplied class-B frame reception.
Row 4: Check the next client-supplied class-B frame for excess credit violations.

A.4 Vector range ratings

The MAC-supplied sendA, sendB, and sendC indications specify a hop-count distance for safe frame transmissions. The client cannot be expected to respond to these indications immediately and (in particular) the current MAC-specified hop-count distance could decrease during the client's frame transmission.

Thus, techniques are needed to validate the client's conformance to recently-generated flow-control indications. For that purpose, a MAC implementation may desire to monitor the maximum and minimum recently asserted hop-count distances, as listed in figure A.6. An exact definition of these credit-restoring conditions is specified in table A.10, where rows are evaluated in top-to-bottom order.

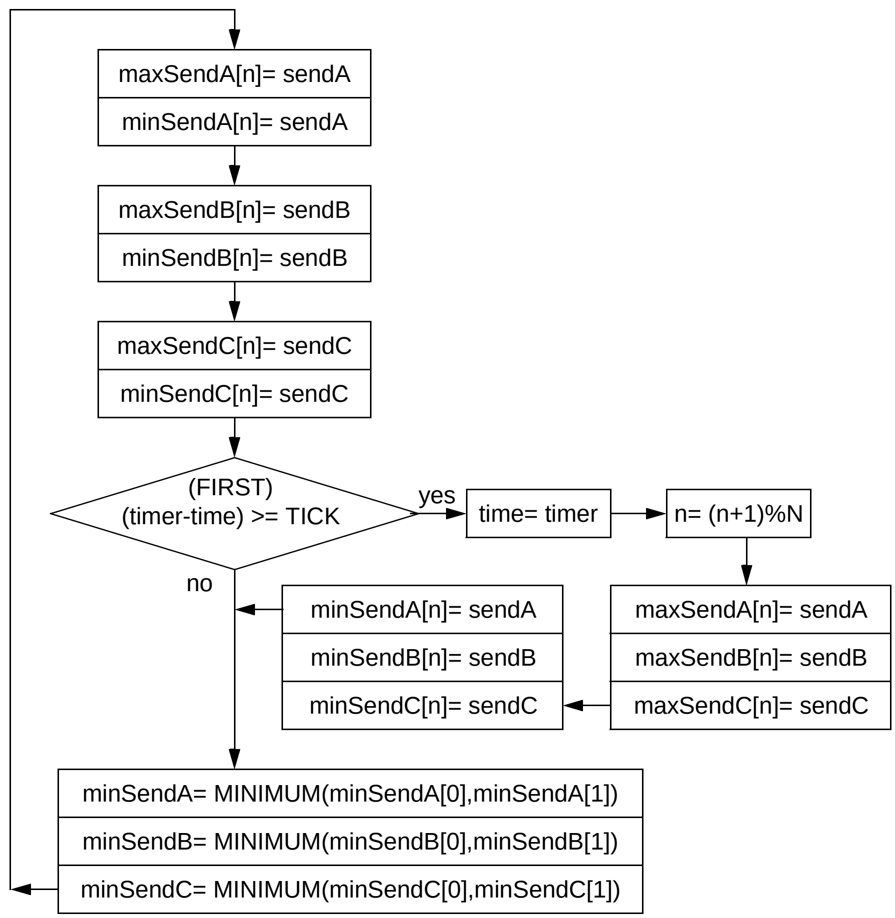

Figure A.6—Vector range ratings

Table A.10—Vector range ratings

Last state		Row	Next state	
state	**condition**		**action**	**state**
FIRST	(time-timer) < TICK	1	maxSendA[n]= MAXIMUM(sendA,maxSendA[n]), minSendA[n]= MINIMUM(sendA,minSendA[n]), maxSendB[n]= MAXIMUM(sendB,maxSendB[n]), minSendB[n]= MINIMUM(sendB,minSendB[n]), maxSendC[n]= MAXIMUM(sendC,maxSendC[n]), minSendC[n]= MINIMUM(sendC,minSendC[n])	FINAL
	—	2	timer= time, n= (n+1)%N, maxSendA[n]= sendA, minSendA[n]= sendA, maxSendB[n]= sendB, minSendB[n]= sendB, maxSendC[n]= sendC, minSendC[n]= sendC	
FINAL	—	3	maxSendA= MAXIMUM(maxSendA[0],maxSendA[1]), minSendA= MINIMUM(minSendA[0],minSendA[1]), maxSendB= MAXIMUM(maxSendB[0],maxSendB[1]), minSendB= MINIMUM(minSendB[0],minSendB[1]), maxSendC= MAXIMUM(maxSendC[0],maxSendC[1]), minSendC= MINIMUM(minSendC[0],minSendC[1])	FIRST

Row 1: Determine minimum and maximum send-interval values.
Row 2: Initialize values for the next sampling interval.
Row 3: Computed maximum and minimum of both window values.