9. Fairness

Editors' Notes: To be removed prior to final publication.

This document is not part of the current 802.17 draft. It has no official standing, all disclaimers apply, etc. etc.

9. Fairness

This clause describes the functions performed by the fairness control unit (FCU). The principal activity of the FCU is the computation of fair rates of ringlet access for client-sourced fairness-eligible traffic. The use of fair rates prevents one station from occupying a disproportionate share of available ringlet capacity by virtue of having a position upstream with respect to other stations on the ringlet. The method of calculating fair rates is known as the fairness algorithm.

Fairness control functions reside within a fairness control unit (FCU), located in the MAC control sublayer of the MAC, as illustrated by the shaded region of Figure 9.1.

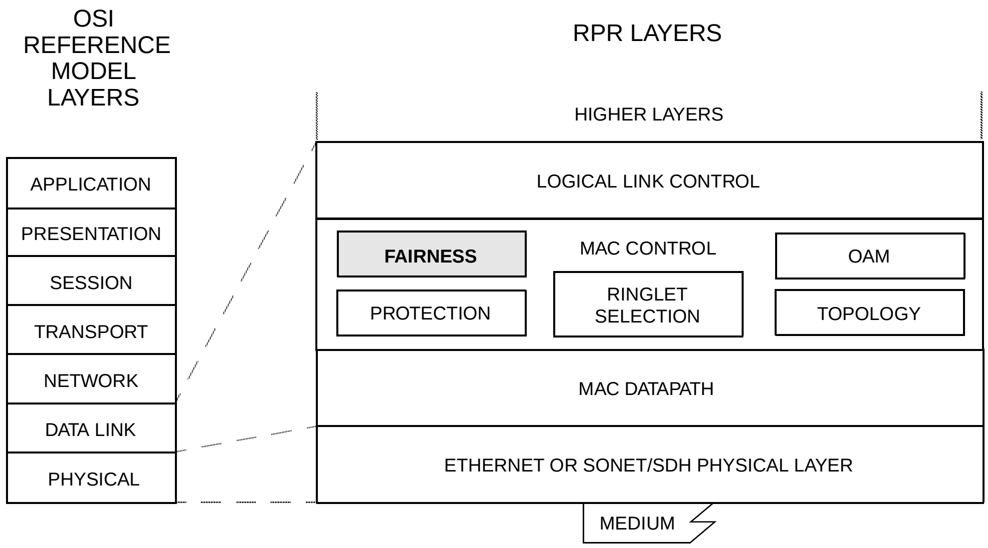

Figure 9.1—The Fairness Control Unit

The FCU supports the ScFcmInd and McFcmInd opcodes of the MA_control.ind primitive (table 5.3). This portion of the MAC service interface provides optional reporting of rate information to the MAC client, allowing the client to deploy fairness features, such as per-destination queuing, that are beyond the scope of the MAC layer.

The following are features and key characteristics of the FCU:

a) Fairness control is applied independently on each ringlet.
b) Fairness control messages are sent on the ringlet opposing that of the associated data.
c) Fairness is applicable only to fairness-eligible traffic (ie. classB excess-rate and classC).
d) A fair rate is associated with the source station (ie. source-based versus, for example, based on source-destination pair).
e) A distinct fair rate is applied to that portion of added traffic transiting a point of congestion on the ringlet.
f) Fair rates are scaled in proportion to administratively assigned station weights (weighted-fairness).
g) Ringlet capacity not explicitly allocated, is treated as available capacity.
h) Capacity explicitly allocated to subclassA0 or classB, if not in use, is treated as available capacity (bandwidth reclamation).
i) The fair rate computation method depends on whether single or dual transit queues are deployed.
j) Methods of aggressive and conservative fair rate adjustment are provided.

k) Optional reporting of rate information to the MAC client allows the client to deploy fairness methods such as per-destination queuing that are beyond the scope of the MAC...

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

The following requirements appear in the current draft, but should be restated so as to be verifiable.

l) **Fast response time**—Because data traffic tends to be bursty, in order to ensure maximum ring bandwidth utilization and to ensure that the protocol is responsive to instantaneous changes in traffic load, it must have a fast response time.
m) **High bandwidth utilization on the ring—**The protocol should be able to achieve very high levels of bandwidth utilization even under heavy loads approaching 100% of the ring capacity.
n) **Scalability**—The protocol should be scalable and should be able to function predictably for all ringlet speeds and ring diameters allowed by this standard.
o) **Stability**—The fairness protocol should enter a steady state within a finite time when presented with a steady input traffic pattern.

This clause includes:

a) a description of FCU operation (subclause 9.1)
b) terminology specific to this fairness control
c) the fairness control state machine specification (subclause 9.5).
d) fairness control message formats (subclause 9.7)
e) guidance in selecting threshold values (informative)
f) C-code example implementation (informative)

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

C-code will be moved to informative annex G after synchronization with state machines.

9.1 Operation

Fairness control activities are categorized by the intervals at which they occur. The categories, in order of increasing interval length, are:

a) per-agingInterval (interval at which allowed rates are locally computed)
b) per-advertisementInterval (interval at which proposed rates are advertised upstream)
c) per-reportingInterval (interval at which local fair rates are broadcast to all stations)

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

It isn't clear to me what rate is reported to the MAC client in the ScFcmInd. I'm assuming the indication is issued when the advertisement is sent.

9.1.1 Overview

At the end of each agingInterval, a station computes allowed rates for client-sourced fairness-eligible traffic. One rate (allowedRate) is applied to all fairness-eligible traffic sourced by the station, while the other (allowedRateCongested) is applied to that portion of the fairness-eligible traffic sourced by the station that transits the most congested station (congestionPoint) on the ringlet from the perspective of the source station. The two rates, as well as the distance (in hops) from source station to congestionPoint, are made available to the datapath, allowing it to regulate access to the ringlet. The per-agingInterval activities are described in subclause 9.1.2.

At the end of each advertisementInterval, a station computes an advertisedRate based on (1) local rate and congestion information and (2) information contained in the advertisement most recently received by the local station. When an advertisement is received by a station, the information it contains is saved for use in computing allowed rates at the end of each agingInterval. The advertisedRate is optionally reported to the MAC client via the ScFcmInd indication. The per-advertisementInterval activities are described in subclause 9.1.3.

At the end of each reportingInterval, rate information is broadcast by the local station to all stations on the ringlet, and is optionally reported to the MAC client via the McFcmInd indication. This allows the client to support fairness functions, such as per-destination queuing, that are beyond the scope of the MAC. The per-reportingInterval activities are described in subclause 9.1.4.

9.1.1.1 Rates

Data rates referenced by the FCU are specified in units of *bytes per agingWindow* where the agingWindow is defined as ‘AGECOEF agingIntervals’ (ie. AGECOEF * agingInterval). The agingWindow is a time period referenced by the aging method that is used to compute addRate, addRateCongested, fwRate, fwRateCongested, and nrXmitRate. As other rates used by the FCU are derived from these rates, the unit is used generally by the FCU.

For purposes of description, the addRate, addRateCongested, fwRate, fwRateCongested, and nrXmitRate, maintained by the datapath as byte counts but aged by the FCU to rates, are collectively called ‘aged rates’. The allowedRate and allowedRateCongested, and the hopsToCongestion associated with the allowedRateCongested, are collectively called ‘allowed rates’.

9.1.1.2 Congestion

A station maintains a local congestion state (localCongested) and a downstream congestion state (downstreamCongested). The latter indicates whether or not the local station has detected downstream congestion based on information received in a control message called a rate advertisement. In the event of downstream congestion, the received rate advertisement identifies the station that is the most congested among the set of contiguous congested stations lying downstream with respect to the local station. The most congested station is known as the congestionPoint and the segment of the ringlet between the local station and the congestionPoint is known as the fairness domain.

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

Draft 2.0 section 9.6.1 contains the following paragraph in the section titled 'Threshold settings'.

Three thresholds, *fullThreshold*, *highThreshold*, and *lowThreshold*, are used to specify congestion thresholds. For dual-queue MACs, they are based on occupancy of the STQ. For single-queue MACs, they are based on the rate at which traffic is flowing through the MAC. The formulas for calculating these thresholds are provided in 9.7.1. When the *lowThreshold* is crossed, the station is considered to be approaching a congested state, and advertises a reduced fair rate. When the *highThreshold* is crossed, the station is considered to be in a congested state, and no more fairness-eligible traffic is accepted from the MAC client for transmission on to the ringlet. For dual-queue MACs, if the occupancy of the STQ crosses *fullThreshold*, it is almost full, and the traffic from the STQ will be prioritized above all other traffic until the occupancy of the STQ drops below *fullThreshold* (by any amount)

The values of STQFullThreshold and rateFullThreshold are initialized but it does not appear that these thresholds are used by the FCU. Are they used by the datapath? Does the FCU initialize values for the datapath? Row 5 of table 9.4 Is it correct that in congested state, no more fairness-eligible traffic is accepted. In conservative mode, isn't the rate just reduced?

From Stein G.: I believe that STQxxxThreshold is mainly the same as xxxThreshold, but by using the STQ prefix it is clearer that these thresholds are a constant number of bytes and are applicable to STQ only. I (Stein) have submitted a comment about a clearification of the nature of the xxxThresholds (are they integer values (in bytes) or are they boolen values that are set to true when the threshold is crossed? The current text is inconsistent.

9.1.1.3 Ramping

Ramping is a method of allowing a gradual increase or decrease of a rate. A rate can ramp-up by a fraction of the difference between its current and its maximum value or it can ramp-down by a fraction of the difference between its current and its minimum value (which is equal to zero in FCU rate computations).

```
example of ramping-up: localFairRate += (unreservedRate - localFairRate) / RAMPCOEF
```

```
example of ramping down: localFairRate -= localFairRate / RAMPCOEF;
```

9.1.1.4 Conservative and aggressive operation

The FCU supports both conservative and aggressive operation.

When operating conservatively, the FCU adjusts rates by ramping so that rate change is gradual. The effect of the previous rate change can be examined before making further change. This avoids congestion as rates increase and underutilization as rates decrease.

When operating aggressively, the FCU sets rates to computed values rather than allowing them to rise or fall gradually. Aggressive operation allows the station to more quickly increase rates to make use of available capacity or more quickly decrease rates to eliminate congestion.

9.1.1.5 Rate normalization

Per-advertisementInterval and per-reportingInterval activities both include the transfer of rate information to other stations on the ringlet. In such cases, the rate is transformed to a 'normal form' (ie. is normalized) by dividing the local rate value by a normalization coefficient. When received by a station, the normalized rate can be (1) restored through multiplication by the normalization coefficient or (2) maintained in the normal form for propagation to other stations. A local (ie. non-normalized) rate can be compared to a normalized rate by creating a normalized copy of the local rate.

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *Is there any requirement that the AGECOEF be identical at all stations on the ringlet?* My understanding (Stein G) is that different stations may have differnet values of LPCOEF, AGECOEF and RAMP-COEF.
>
> They are only smoothing parameters and/or included in the normalization so that they are not visible to the outside (but of course the effect of those parameters are visible).
>
> I am not sure if one station can get some kind of preference by using eg. a small RAMPCOEF. A small RAMPCOEF will enable the station to send more traffic at an earlier time.

The normalization coefficient is computed locally as the product of the local station weight (WEIGHT), the rate coefficient (RATECOEF), and the age coefficient (AGECOEF). The local station weight allows scaling of fair rates among stations on the ringlet so, for example, clients with larger traffic requirements can be granted a larger share of available capacity. The rate coefficient allows scaling of the rate values to a standard link rate. While the link rate is identical on all links of the ringlet, the scaling allows the magnitude of rates to be reduced, avoiding overflow of the field containing the transferred rate. The age coefficient allows the rate to be scaled to a standard agingWindow (ie. AGECOEF agingIntervals), again preventing overflow of the rate field.

9.1.1.6 Operation on a wrapped ring

A rate advertisement message travels hop-by-hop on a ringlet and is processed by the FCU at each station. When a ring wraps, the FCM also wraps at the wrapPoint. In the case of a station deploying edge-wrapping, the FCM is processed on both of the fairness control instances associated with the station. In the case of center-wrapping, the FCM is processed by the fairness control instance associated with the originating ringlet but is then forwarded to the next station on the opposing ring. The FCM retains the RI of the 'wrap from' ringlet. This allows stations on the 'wrap to' ringlet to identify that the FCM has wrapped.

Fairness control functions are performed independently on each ringlet. In cases where the FCM indicates downstream congestion (ie. advertisedRate less than FULL_RATE) when it transits the wrapPoint, the wrapPoint can be viewed as a congestionPoint on the 'wrap to' ringlet, as it has no downstream neighbor on the 'wrap to' ringlet. The distance between a local station on the 'wrap to' ringlet and the congestionPoint is equal to the hopsToWrapPoint MIB attribute that is populated by the protection control unit.

9.1.2 Per-agingInterval activities

The following sequence of activities is performed by the FCU at the completion of each agingInterval.

a) aging (subclause 9.1.2.1) of counts maintained by the datapath. Aging has the effect of transforming the counts maintained by the datapath to rates referenced by the FCU.
b) low-pass filtering (subclause 9.1.2.2) of previously aged rates. Low-pass filtering has the effect of smoothing the rate value with respect to the sequence of previous rate values.

c) determination of the local congestion state (subclause 9.1.2.3).
d) computation of a localFairRate (subclause 9.1.2.4).
e) update of local congestion state to reflect changes associated with computed localFairRate (subclause 9.1.2.6)
f) computation of allowed rates (subclause 9.1.2.7) made available to the datapath.f)

The sections that follow describe in more detail the activities performed per-agingInterval.

9.1.2.1 Aging

The global variables addRate, addRateCongested, fwRate, fwRateCongested, and nrXmitRate (see subclause 9.5.3 for definitions) are used to accumulate the rates at which traffic is added to and/or transited by the local station. The value is incremented by the datapath as each byte conforming to the definition of the variable is added or transited by the station. The value is 'aged' by the FCU at the conclusion of the agingInterval (subclause 9.1.2.1).

Aging is accomplished by multiplying the variable by the factor (AGECOEF-1)/AGECOEF. AGECOEF is a coefficient that specifies the relative weights assigned to (a) the change in value of a variable during the most recent *agingInterval* and (b) the value of the variable at the end of the previous agingInterval. The computation is:

$$\texttt{rateValue}_{new} = (\texttt{rateValue}_{prev} * (\texttt{AGECOEF} - 1)) / \texttt{AGECOEF}$$

where AGECOEF is a coefficient that specifies the relative weights assigned to (a) and (b) above.

```
e.g.: addRate = (addRate * (AGECOEF - 1)) / AGECOEF;
```

The aging operation allows an upper-bound to be placed on the value of the variable, insuring that overflow does not occur. Further, if the value of the variable increases by a fixed amount (e.g. FIXED_INC) in each agingInterval, the value of the variable asymptotically approaches the value AGECOEF*FIXED_INC. Hence, the application of aging to the variable, allows the value to be interpreted as a rate. The rate is specified in bytes per AGECOEF agingIntervals, or bytes per agingWindow where an agingWindow is equal to AGECOEF agingIntervals.

NOTE—The upper-bound on the rate value is AGECOEF*MAX_INC where MAX_INC is the maximum amount by which the count can increase during an agingInterval. The value of AGECOEF is selected so as to avoid overflow of the variable.

9.1.2.2 Low-pass filtering

Low-pass filtering is a method of smoothing that removes rapid oscillations from a sequence of values. The FCU applies low-pass filtering to the rates derived using the aging method described subclause 9.1.2.1. A new low-pass filtered value is calculated at the end of each agingInterval as a weighted sum of (a) the observed value during the most recent agingInterval and (b) the previous low-pass filtered value. The computation is:

$$\texttt{lpFilteredValue}_{new} = (\texttt{observedValue}_{new} + (\texttt{lpFilteredValue}_{prev} * (\texttt{LPCOEF} - 1))) / \texttt{LPCOEF}$$

where LPCOEF is a coefficient that specifies the relative weights assigned to (a) and (b) above.

e.g.: `lpAddRate = ((LPCOEF-1) * lpAddRate + addRate) / LPCOEF;`

NOTE—Aging and low-pass filtering can be contrasted in that the primary purpose of aging is to convert a byte count to a rate and to insure that the value does not overflow the variable in which it is contained, while the primary purpose of low-pass filtering is to smooth successive values of the rates.

9.1.2.3 Computing the local congestion state (before localFairRate computation)

The congestion state of the station is evaluated at the end of each agingInterval after low-pass filtering has been performed. The station is in the congested state if any one of following independent conditions are met:

a) The total non-reserved traffic transmitted (ie. added or transited) by the station (lpNrXmitRate) exceeds the capacity available for unreserved use (unreservedRate). The latter value is the difference between the LINK_RATE and the reservedRate.
b) *Considered only when the MAC employs a single transit queue (~dualQueueMAC):* The total non-reserved traffic transmitted (lpNrXmitRate) exceeds a rate threshold value (rateLowThreshold) associated with the outbound link
c) *Considered only when the MAC employs a single transit queue (~dualQueueMAC):* The access delay threshold for either classB or ClassC traffic expires, indicating a high level of transit traffic.
d) *Considered only when the MAC employs dual transit queues (dualQueueMAC):* The depth of the secondary transit queue (STQDepth) exceeds an STQ depth threshold (STQLowThreshold).

In all other cases, the station is placed in the uncongested state.

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *The C-code indicates that the state is not set, in some cases, until the localFairRate computation, but it seems clearer to say that the state is set in all cases, but it can be changed in some cases after the localFairRate computation.*

9.1.2.4 Computing the localFairRate

At the end of each agingInterval, after determining the local congestion state, the station computes a fair rate value (localFairRate). The method of computation depends on the local congestion state and mode of operation of the station, and whether or not the station has made a transition from the uncongested state to the congested state.

a) *Uncongested state:* set to the total unreserved capacity (unreservedRate). The unreservedRate is the difference between the LINK_RATE and the total capacity reserved on the ringlet (reservedRate). The unreservedRate is the maximum value to which the localFairRate can be set.
b) *Congested state and aggressive mode of operation:* set to the value of the aged and low-pass filtered rate of of addition (lpAddRate) to the ringlet.
c) *Transition to congested state during current agingInterval and conservative mode of operation:* set to the unreservedRate divided by the sum of the weights of active stations on the ringlet (activeWeights). The computation of activeWeights is described in subclause 9.1.2.5.
d) *Congested state and conservative mode of operation (station was in congested state during previous agingInterval):* In this case, the value of the localFairRate is assigned as follows:
   1) *round-trip time (RTT) elapsed since localFairRate last updated and nrXmitRate below rateLowThreshold (ie. currently no rate congestion):* ramp-up localFairRate (subclause 9.1.1.6) by 1 / RAMPCOEF of the difference between the current localFairRate and the available rate.
      ie.: $\texttt{localFairRate}_{\texttt{new}}$ `+= (unreservedRate - ` $\texttt{localFairRate}_{\texttt{crnt}}$ `) / RAMPCOEF`
   2) *round-trip time (RTT) elapsed since localFairRate last updated and nrXmitRate above rateHighThreshold (ie. current rate congestion):* ramp-down localFairRate (subclause 9.1.1.7) by 1 / RAMPCOEF of the current localFairRate.
      ie.: $\texttt{localFairRate}_{\texttt{new}}$ `-= (` $\texttt{localFairRate}_{\texttt{crnt}}$ `) / RAMPCOEF`
   3) *all other cases:* localFairRate unchanged from previous agingInterval.

9.1.2.5 Computing activeWeights

The activeWeights computation provides the value of the localFairRate in the conservative mode of operation when the transition from the uncongested state to the congested state occurs during the current agingInterval. In this case, the localFairRate cannot be computed by ramping-up or ramping-down, as these methods rely on the existence of a ‘previous’ value of localFairRate. Instead, the FCU performs a weighted partition of available capacity over those stations on the ringlet designated as ‘active’. The exact method of computing activeStations is implementation-specific. It is required only that the number lie between 1 and the number of stations on the ring (numStations) in order that the fairness algorithm converge. Alterative methods of computing activeWeights will, in general, differ in the time required for convergence.

One method of computing activeWeights is to record, at the local station, the identities of stations from which fairness-eligible frames were received during the current agingInterval. At the end of the agingInterval, the weights of those stations from which frames were received, are summed to yield activeWeights,

9.1.2.6 Computing the local congestion state (after localFairRate computation)

In the conservative mode of operation, the congestion state can change from congested to uncongested as a result of the localFairRate computation (9.1.2.4). This transition occurs if both of the following conditions are met:

a) the localFairRate reaches the level of the unreservedRate (ie. localFairRate is fully ramped-up)
b) the total non-reserved transmit rate (nrXmitRate) falls below the low rate-congestion threshold (rateLowThreshold).

9.1.2.7 Computing allowed rates

At the end of each agingInterval, following the computation of the localFairRate and local congestion state, the FCU calculates the values of the allowedRate, allowedRateCongested, and hopsToCongestion. These variables are globally defined and shared with the datapath.

Computation of the allowedRate, allowedRateCongested, and hopsToCongestion is described in more detail in the sections that follow.

9.1.2.7.1 Computing the allowedRate

The allowedRate is the value communicated to the datapath, specifying the aggregate rate at which client-sourced fairness-eligible traffic can be added to the ringlet by the station. The method of computing the value of the allowedRate depends on the operational mode of the station as follows:

a) *aggressive operation:* assigned the configured maximum rate permitted on the ringlet.
b) *conservative operation:* depends on the local congestion state of the station as follows:
    locally congested: assigned the value of the localFairRate.
    locally uncongested: value of allowedRate is ramped-up (subclause 9.1.1.3) by 1 / RAMPCOEF of the difference between the current allowedRate and the maximum allowed value of this rate.
    ie.: $\texttt{allowedRate}_{\texttt{new}}$ `+= (MAX_ALLOWED_RATE - ` $\texttt{allowedRate}_{\texttt{crnt}}$`) / RAMPCOEF`

9.1.2.7.2 Computing the allowedRateCongested

The allowedRateCongested is the value communicated to the datapath, specifying the rate at which client-sourced fairness-eligible traffic destined for stations beyond the congestionPoint, can be added to the ringlet. The method of computing the allowedRateCongested depends on the downstream congestion state. Downstream congestion is inferred when the rcvdFairRate extracted from the rate advertisement is less than the special value FULL_RATE.

a) *downstream congested:* value based on the weight-adjusted localFairRate of the most congested station (congestionPoint) in the fairness domain. The fairness domain is that portion of the ringlet over which the local station contributes to downstream congestion. The congestionPoint is also the station lying furthest downstream in the fairness domain. The weight-adjustment of the localFairRate accounts for differences in configurable station weight between the congestionPoint and the local station. Rate information associated with the congestionPoint is obtained from the most recently received rate advertisement (subclause 9.1.3.3).
b) *downstream uncongested:* value of allowedRateCongested is ramped-up (subclause 9.1.1.6) by 1 / RAMPCOEF of the difference between the current allowedRateCongested and the maximum allowed value of this rate.

ie.: $\texttt{allowedRateCongested}_{\texttt{new}}$ += (MAX_ALLOWED_RATE - $\texttt{allowedRateCongested}_{\texttt{crnt}}$) / RAMPCOEF

9.1.2.7.3 Computing the hopsToCongestion

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

I need to better understand the following comment that appears in the C-code:

```
// hopsToCongestion should be left at the last congestion point so that
// the allowedRate is not all of a sudden presented to the former
// congestion point. One could add the following statement:
// if (allowedRateCongested == allowedRate)
// hopsToCongestion = 255;
// But it is unnecessary since it has no effect upon the rate at which
// the station may transmit.
```

It does seem clear that it isn't necessary to change the value of hopsToCongestion as congestion can be inferred from the equality of allowedRate and allowedRateCongested. It's not clear to me, however, exactly how the old value of hopsToCongestion is used by the datapath.

The hopsToCongestion is the distance (in hops) between the local station and the congestionPoint. In the absence of downstream congestion, the hopsToCongestion is set to the value MAX_TTL. In the presence of downstream congestion, the method of computation is normally:

hopsToCongestion = MAX_TTL - rcvdTTL

where rcvdTTL is the value of TTL in the most recently received rate advertisement.

If, however, the ring is in the wrapped state, and the rate advertisement originated at a congestionPoint on the opposing ringlet, hopsToCongestion is set to the number of hops between the local station and the wrap Point (hopsToWrap). This value is obtained from the rprIfHopsToWrap MIB attribute whose value is supplied by protection control.

NOTE —Fairness control activities are performed independently on each ringlet. When a hop-by-hop rate advertisement wraps from ringlet X to ringlet Y, the wrapPoint on ringlet Y is also a congestionPoint. The wrapPoint cannot have an upstream neighbor on ringlet Y as the ringlet is broken at that point.

9.1.3 Per-advertisementInterval activities

The following activities are performed by the FCU at the completion of each advertisementInterval:

a) computation of the advertisedRate (subclause 9.1.3.1).
b) transmission of a rate advertisement to the upstream neighbor (subclause 9.1.3.2)

The FCU also performs activities associated with the reception of rate advertisements (subclause 9.1.3.3) sent by the downstream neighbor. The per-advertisementInterval activities are described in more detail in the sections that follow.

9.1.3.1 Computing the advertisedRate

At the conclusion of each advertisementInterval, the local station sends an advertisedRate to its upstream neighbor. The advertisedRate is assigned one of three values:

a) the rcvdFairRate (normalized localFairRate of the downstream congestionPoint)
b) the normLocalFairRate (normalized localFairRate of the station)
c) the FULL_RATE (indicating the absence of both local and downstream congestion)

The rcvdFairRate is assigned when it is determined that the local station is contributing to downstream congestion by forwarding fairness-eligible traffic at a normalized rate greater than the normalized localFairRate of the congestionPoint. Advertising the rcvdFairRate remedies this by reducing the rate of the upstream neighbor to the advertisedRate of the congestionPoint. When the downstream congestionPoint lies on the same ringlet as the local station, the forwarding rate of the local station includes all fairness-eligible traffic forwarded by the local station to the congestionPoint. This value is specified by normLpFwRate. When the congestionPoint lies on the opposing ringlet (ie. the advertisement originated on the opposing ringlet and the ring is in the wrapped state), the wrapPoint is a congestionPoint on the local ringlet. In this case, the traffic forwarded by the local station to the congestionPoint on the opposing ringlet, passes through the congestionPoint (wrapPoint) on the local ringlet. This rate of forwarding is specified by normLpFwRateCongested, as it is traffic that passes through the congestionPoint of the local ringlet.

The normLocalFairRate is assigned when the local station is congested and the degree of congestion is greater than or equal to the degree of congestion of the downstream station. This reflects the rule that a station advertises the rate of the most congested station of which it is aware. In this case it is the local station that is most congested.

The value FULL_RATE is assigned when neither local nor downstream congestion has been detected. Advertising this rate to the upstream station indicates the absence of downstream congestion.

9.1.3.2 Sending the advertisedRate

At the conclusion of each advertisementInterval, a rate advertisement is sent from the local station to its immediate upstream neighbor. The advertisement is carried in a control message known as a single-choke fairness control message (SC-FCM). This message is identified by a fairnessMsgType of SC (0x00). The advertisedRate is carried in the fairnessControlValue field. The message also contains source address (SA), time-to-live (TTL), and ringlet identifier (RI) fields.

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *Is the RI that appears in the rate advertisement that of the direction of the associated control flow or the associated data flow?*

When the advertisedRate takes the value of the normLocalFairRate or the value FULL_RATE, the local station is the originator of the message. The SA is set to the local MAC address, the TTL is set to the value MAX_TTL, and the RI is set to the RI of the local ringlet.

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *Where in the state machine (or C-code) is the TTL decremented?*

When the advertisedRate takes the value of the rcvdFairRate, an upstream station is the originator of the message. The values of rcvdSA, rcvdTTL, and rcvdRI are copied from the corresponding fields of the received message.

9.1.3.3 Receiving the advertisedRate

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *John Lemon commented that controlValue should be changed to fairnessControlValue when the context is not clear. I'd further suggest that the name be changed to reflect the contents of the field, as is the case with all the other fields in the message. Why not something like msg.fairRate?*

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> 'In the event that the SA is the same as the local MAC address, the rcvdFairRate is always set to the value FULL_RATE'. Is this statement true if the message has wrapped?

On receiving a fairness control message with a fairnessMsgType SC (0x00), the station saves the fairness-ControlValue, SA, TTL, and RI in the rcvdFairRate, rcvdSA, rcvdTTL, and rcvdRI respectively. In the event that the SA is the same as the local MAC address, the rcvdFairRate is always set to the value FULL_RATE. The received values are not referenced until the end of the current agingInterval.

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *Draft 2.0 indicates that an scFcmInd is issued by FCU to MAC Client when an SC-FCM is received. Table 5.3 indicates that the following information is communicated:* allowedRate, allowedRateCongested, hopsTo-Congestion, ringletID. This is difficult to understand since the SC-FCM contains the advertisedRate (in fairnessCon-trolValue field), but not the fields mentioned. The allowed rate fields are computed at the end of each local aging Interval, so they don't even have the same periodicity as the SC-FCM. What is the intention here? To provide the allowed rates to the client at each agingInterval? To provide the allowed rates to the client at each advertisementInter-val, the provide the advertisedRate to the client on receipt of an SC-FCM, or something else?

Optionally (by configuration) issue scFcmInd to MAC client.

9.1.4 Per-reportingInterval activities

> ***Editors' Notes:*** *To be removed prior to final publication.*
>
> *Why is the interval for sending the MC-FCM not given its own name? Suggest reportingInterval. Why is it a fixed (ie 10) multiple of the advertisementInterval. Explain that the reportingInterval is a fixed multiple of the advertisementInterval.*

At the end of each reportingInterval, each station issues a rate report. On receiving a rate report, the station may send information contained in the report to the MAC client. These activities are described in the sections that follow.

9.1.4.1 Sending the rate report

At the end of each reportingInterval, the local station broadcasts a rate report to all stations on the ringlet. The rate report is contained in a multichoke fairness control message (MC-FCM), identified by the fairnessMsgType MC (0x01). If the station is locally congested, the fairnessControlValue is set to the normalized value of the localFairRate (normLocalFairRate). If the station is not locally congested, the fairnessControlValue is set to the FULL_RATE. The SA field is set to the local source MAC address, the TTL field is set to MAX_TTL, and the RI field is set to the RI of the fairness instance.

9.1.4.2 Receiving the rate report

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

The list in table 5.3 also includes the fairnessMsgType, but it seems this is implied by the name of the indication.

On receiving an MC-FCM, the station optionally (by configuration) provides the fairnessControlValue, SA, TTL, and ringletID via an McFcmInd (see table 5.3) issued to the MAC client.

The client can use this information, for example, to identify multiple points of congestion on the ringlet and to deploy distinct queues per congestionPoint in order to prevent head-of-line (HOL) blocking.

NOTE (informative) —The client can also take advantage of Virtual Destination Queueing (VDQ) by utilizing the multi-choke concept of FA. VDQ combined with FA can increase ring utilization. The multi-choke concept deals with the case where a station wants to send traffic to a destination that is closer than a congested link. As an example, consider the case where station 1 wants to send traffic to station 2, and the link between stations 2 and 3 is congested. FA will allow station 1 to send as much traffic as it wants to station 2, and will only limit traffic to stations beyond the congested link to the fair rate. In a multi-choke implementation of the FA, each client will track advertised fair rates for congested stations. A station is allowed to send unlimited traffic to any station between itself and the first congested station (choke-point). It can send traffic to stations between the first and second choke-point based on the first choke-point's advertised fair rate. In general, a station can send traffic to a particular destination if it has satisfied the fair rate conditions for all choke-points between itself and the destination. The maximum possible number of choke-points is equal to the number of stations on the ring.