

# Shapers

This presentation addresses the following:

- Issues with shaperC (i.e. FE shaper)
- Issues with ShaperM (i.e. Control shaper)
- shaperB and Marking
- Issues with the general shaper definition

Comments against the Draft

All issues can be resolved, however, the issue is complexity

# ShaperC (i.e. FE)

shaperC

decSize	incrSize	hiLimit	LowLimit	passC	!passC
clientC	allowRate*time	2*sizeMTU	sizeMTU	creditC>=loLimit	creditC<loLimit

shaperD

decSize	incrSize	hiLimit	LowLimit	passD	!passD
!A0	unreservedRate*time	2*sizeMTU	sizeMTU	creditD>=loLimit	creditD<loLimit

stopC = !(passC && passD && addRateOK)

addRateOK = (addRate < allowRate) && (nrXmitRate < unreservedRate) &&  
 ((STQdepth == 0)|| (fwdRate > addRate)|| (STQdepth < STQHighTreshold))

addRateOKCongested = addRateOK && (addRateCong < allowRateCong)

# ShaperC issues

- ShaperC definition is complex and redundant
- ShaperC is defined as having an increment of  $\text{allowRate} * \text{TIME}$
- TIME can be:
  - Equal to Aging interval (logical choice), in this case the condition is  $(\text{addRate} < \text{allowRate}) \rightarrow \text{addRateOK}$  does this already.
  - Greater/Smaller than Aging interval (not very wise requires multipliers or dividers), and still will not provide anything more.
- Fix the ShaperC, increment should maxburst for FE add traffic (something like  $\text{MAX\_ALLOW} * \text{TIME}$ ),  $\text{addRateOk}$  given the aging behaviour and the  $\text{addRate} < \text{allowRate}$  test does the same thing
- Note:  $\text{addRateOK}$  condition does not require the condition  $(\text{nrXmitRate} < \text{unresrevedRate})$ , it does not hurt. One could use  $\text{passD}$  as an indication.

# ShaperM

We want to stop control packets going beyond a certain threshold (i.e. not have control use up the data credit)?

There is two issues:

- There is “essential control” i.e. protection/Topology that should not be prevented from being sent
- Non essential control (i.e. OAM) need to be shaped

Now we have shaperM (creditM) and the shaper can take credit from A0, A1, B, FE.

Issue: If I send loads of OAM packets, taking credit from shaperM.

When a Protection packet is to be sent there is no more credit...

Solution:

Protection/topology should not be controlled by shaperM.

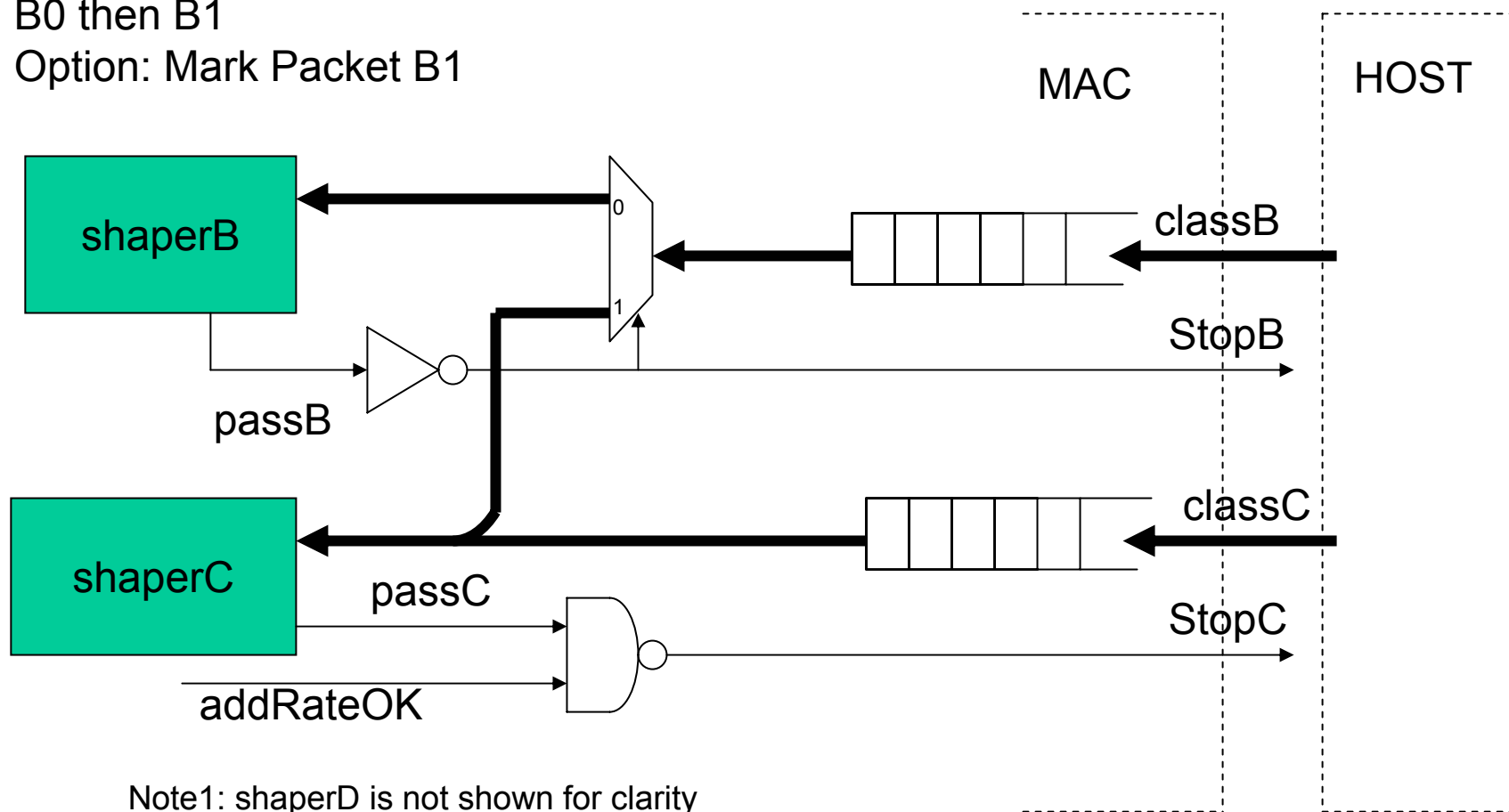
# ShaperM

	Service Class	Shaper
Idle	A0	shaperI (fast/slow timers), provisioned
Fairness	A0	Fast/slow timers (i.e. single/multi-choke Interval), provisioned
Protection/Topology	A0	Fast, slow timer, provisioned
Other Control (OAM)	<ul style="list-style-type: none"><li>• A0/A1</li><li>• B</li><li>• C</li></ul>	shaperM and <ul style="list-style-type: none"><li>• shaperA0</li><li>• shaperA1</li><li>• shaperB</li><li>• shaperFE</li></ul>

# ShaperB Issues

- Issue with Marking traffic as FE
- If source is allowed to mark packet as FE, it can cause B0 traffic to be stopped
- B traffic need not only to watch passB but also passC (which break the design rule that back-pressure are independent).
- This optional behavior, should be left outside the standard.

Default Behavior allocate  
B0 then B1  
Option: Mark Packet B1



Note1: shaperD is not shown for clarity

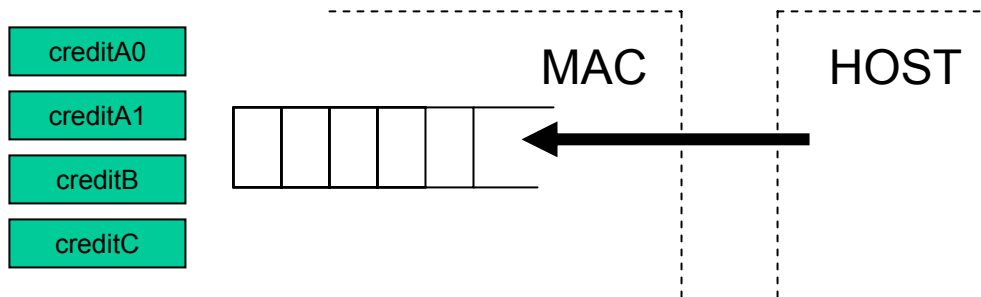
Note2: addRateOK (suppose it is a 0 or 1)

# General Definition of a Shaper

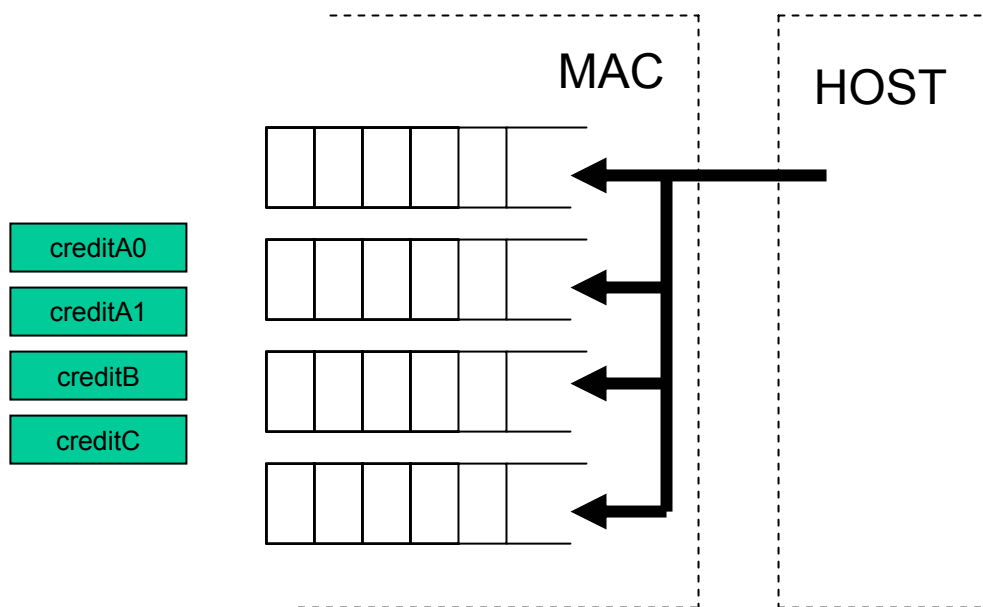
- Shaper reset credit when there is no packets waiting
- This is to prevent a corner case with classA1 jitter guarantees
- Two main types of implementation for mac queuing:
  - Store and forward (every service class has its own queue storage queue  $> 1$  MTU)
  - Cut-thru (small storage, and a single queue for all classes)



## Cut Thru



## Store and Forward



- Issues:
  - complexity and cost
- Store and Forward has no side effect (but expensive solution for a number of applications)
- Cut-thru requires other type of solutions (e.g. Timers)
  - Testing and verifying these type of implementation is not easy.
  - Instead of complicating the solution space, why not look at provisioning parameters properly (i.e. set realistic values for A1 traffic)
- Solution:
  - Remove this condition
  - Define in the case of 2 TB acceptable values for A1 bandwidth

