

QCN Extensions for Monitoring Feedback Request Straw{man,poll}

Mitch Gusat, IBM ZRL

Dallas Plenary
Nov. 12, 2008

What is Fb_Rq? On demand status info

Why: Monitoring, performance profiling... said [here](#) and [here](#)

How: Build on the investment in .1Qau-compliant switches

- => Deliver the existing load data in clear to the edge nodes
- => RP-driven Fb pull in addition to CP's push
- => Extend scope of .1Qau CM: If congestion still arises, call QCN

Fb_Rq Basics

1. RP: Tx Fb_Rq (CNM)

2. CP: Rx Fb_Rq

1. set $P_s=1$

2. dump extended queue status info

1. *QCN CNM +*

2. *Q^{sizemax}, Q^{eq}, Q^{delta}*

How about PngCnt and TxCnt (see [here](#))?

Features @ cost to CP => fine resolution monitoring...

3. CP:Tx Fb_Rp back to RP

4. RP: Rx Fb_Rp

1. send Fb_Rp to upper layer

Concerns about FbRq

1. Cost

1. if $QCN=True$, $\$(Fb_Rq) \rightarrow \varepsilon$
2. else, $\$(Fb_Rq) = O(QCN)$

2. Overhead

1. $\ll 1\%$ with s/w-driven monitoring
2. upper-bounded by CP and RP h/w
3. Re-use CNM format and .1Qau-compliant CP h/w

3. Sim results

→ see next page

Simulation results: Contemplative Stability



Observation instead of control...



No algorithm.

Benefits

1. Timely: on demand L2 feedback to apps
2. Accurate: Detailed Q info is available in CP. Ship it to the RP.
3. Cheap: Info already known. Ship it to the RP.
4. Self-regulating: RP and CP can decide their ovhd. limits.
5. Better / Different from IETF's IPFIX
6. Multiplies the ROI on .1Qau to apps that wouldn't care or trust CM w/o an associated monitoring option

Strawpoll: Reason and Question

- Customers find the Fb_Rq useful and desirable
 - side-effect, increase the acceptance of QCN in 'hostile' markets
- A form of Fb_Rq will likely be de-facto supported by most vendors
 - ..cat's already out of the bag!
- Question is about its standardization...

Should Fb_Rq be an .1Qau option, or better be left to vendors' discretion?

That's all, thank you!