draft-ginsberg-lsr-isis-flooding-scale

Les Ginsberg, Cisco
Peter Psenak , Cisco
Marek Karasek, Cisco
Acee Lindem, Cisco
Tony Przygienda, Juniper

Significant Changes

No update to the draft

Additional Test Results:

Tested w receiver pacing rather than drops

Algorithm Guidelines

Flooding burst durations are not long-lived

- 2000 LSPs/300 per sec is ~7 seconds

Receiver performance may be affected by transient conditions

Faster recovery requires minimizing retransmissions =>

- Response time in small number of seconds (< 5)

Aggressive slowdown / Less aggressive speedup

Must work with enhanced nodes and legacy nodes

- Receiver may ack quickly or slowly
- Flooding optimizations? (Parallel link suppression, dynamic flooding)
- Receiver may/may not implement optimized packet priority

POC Algorithm Overview

Tracks rate of transmissions vs rate of acknowledgments over a multi-second history

Configurable Parameters

- Maximum LSP Transmission Rate (per node) (LSPTxMax)
- **Receiver ACK Delay in ms (per neighbor)

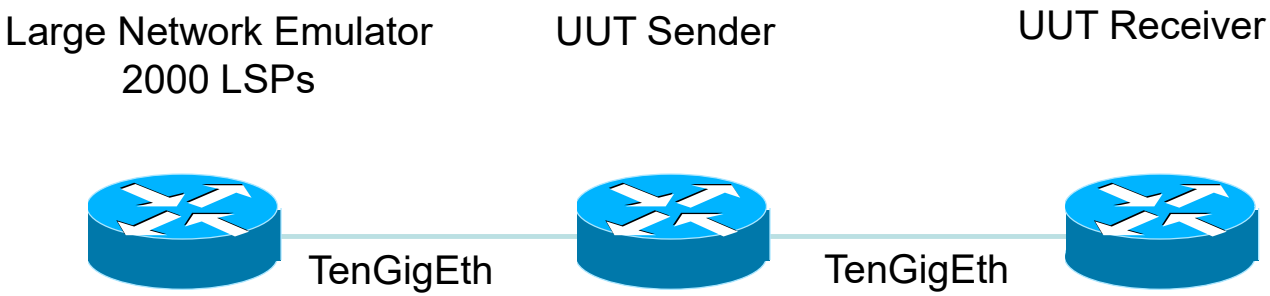
Incorporates expected ACK delay

Agnostic to reason for delay (Tx loss, receiver input queue loss, punt path performance, CPU contention, …)

CurrentLSPTxMax is the current active flooding rate/second

** Candidate for advertisement

Setup and Test procedure



Test procedure:

- Reset UUT Receiver and measure time to download 2000 LSPs from UUT Sender over P2P interface
- 1 ms PSNP delay used in most tests
- Control rate of receiver processing w test code
  - Drop excess (congestion) or
  - Defer processing – no drops

LSP Transmission Bursts

Simplest scheme is to send one LSP every 1/LSPTxMax ms

At higher transmission rates it is unrealistic to expect scheduling at this resolution

Burst size is adjusted based on LSPTxRate with the expectation that:

- Transmitter will only be scheduled a limited number of times/second
- Some scheduling delays may occur – therefore we may need to “catch up”

We refer to this as “Optimized”

Baseline flooding

Test Procedure

- 2000 LSPs
- Receiver Unlimited
- No Tx adjustment

Flooding	Start/End LSPTxRate	LSPTxMax	Time [ms]	Retrans
Base	33/33	NA	66528	0
Base	333/333	NA	7432	0
Optimized	300/300	300	6324	0
Optimized	1000/1000	1000	1768	0
Optimized	2000/2000	2000	1092	0
Optimized	5000/5000	5000	832	0

SlowingDown

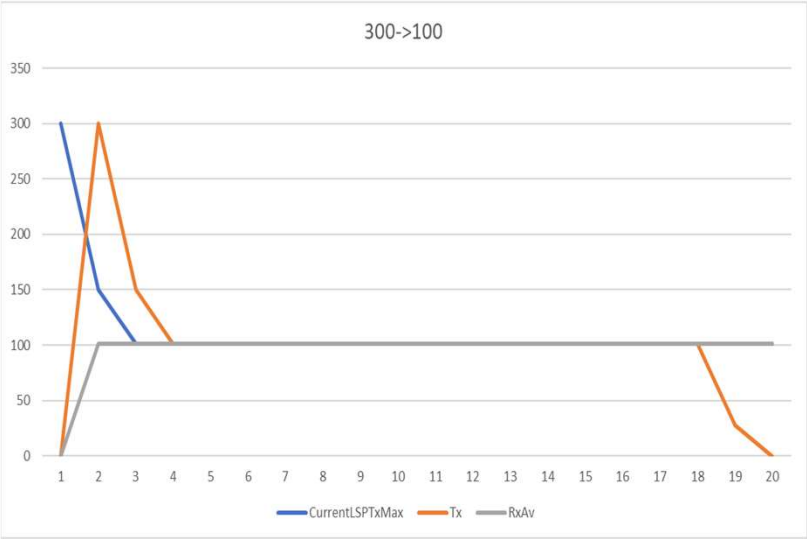
Test procedure:

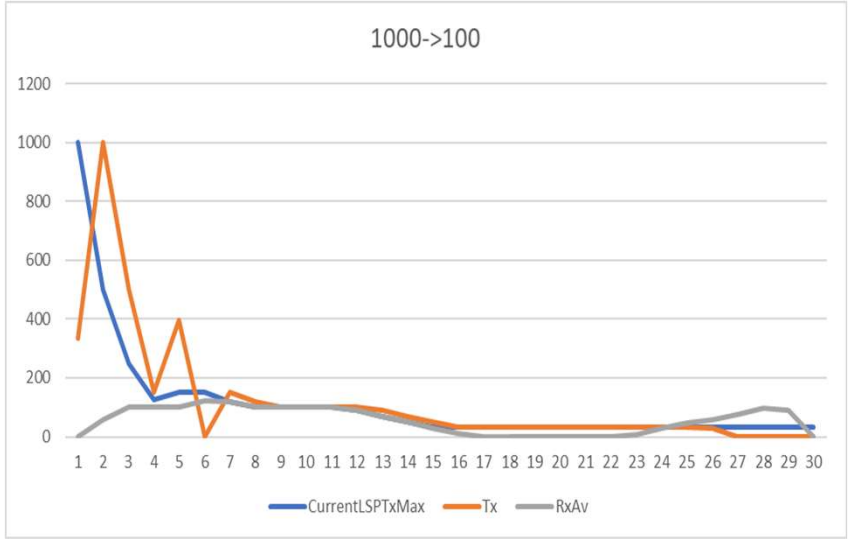
- 2000 LSPs
- Receiver's capability changed to 100 LSP/sec:
  - additional LSPs dropped
  - additional LSPs not dropped
- Sender detects lagging acks and adjusts rate

Flooding	Start/End LSPTxRate	LSPTxMax	Time [ms]	Retrans
Base	33/33	NA	66268	0
Base	333/333	NA	20988	2439 (122%)
Optimized w Drops	300/100	300	20076	257 (13%)
Optimized w Drops	1000/100	1000	19456	1475 (74%)
Optimized – no Drops	300/100	300	19004	0
Optimized - no Drops	1000/100	1000	24908	1804 (90%) **

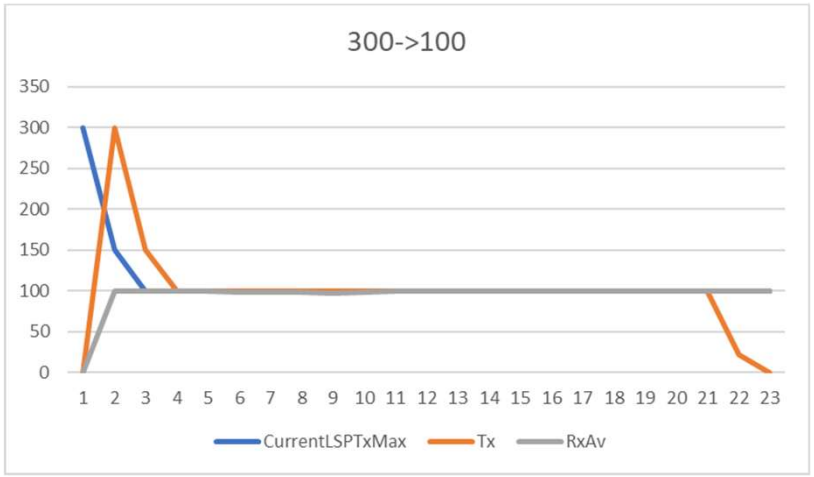
** Retransmissions are due to receiver processing delays

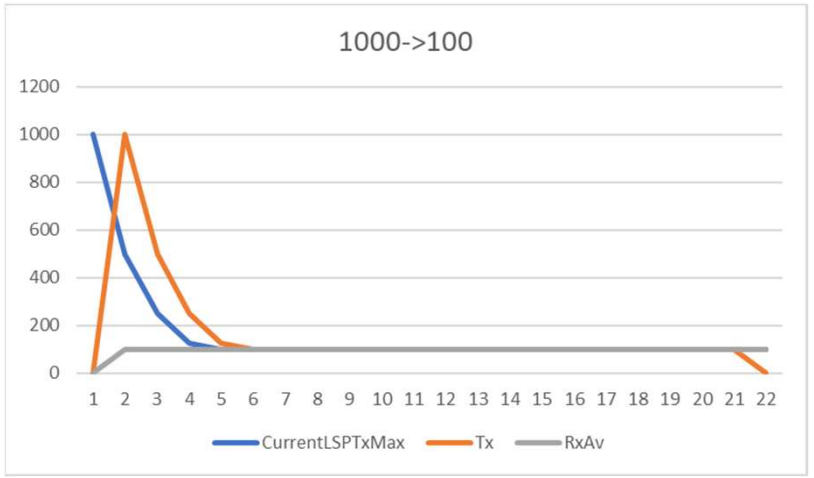
SlowingDown w Drops





SlowingDown no Drops





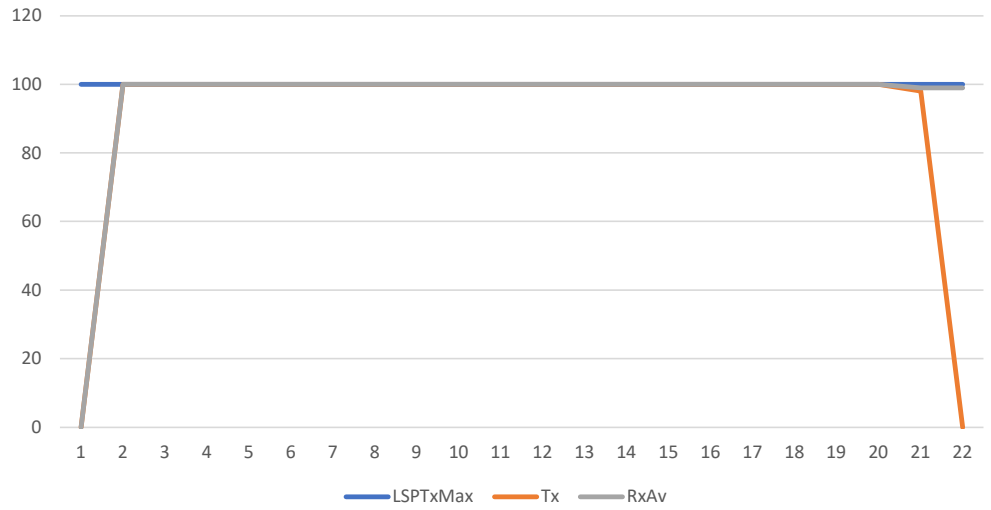
Steady state, receiver affected

Test procedure:

- 2000 LSPs
- Receiver's capability is 100 LSP/sec
- Sender no adjustment required

Flooding	Start/End LSPTxRate	LSPTxMax	Time [ms]	Retrans
Base	33/33	NA	66268	0
Base	333/333	NA	20988	2439 (122%)
Optimized	100/100	300	19444	0
Optimized	100/100	1000	19488	0

Steady state, receiver affected



SpeedingUp

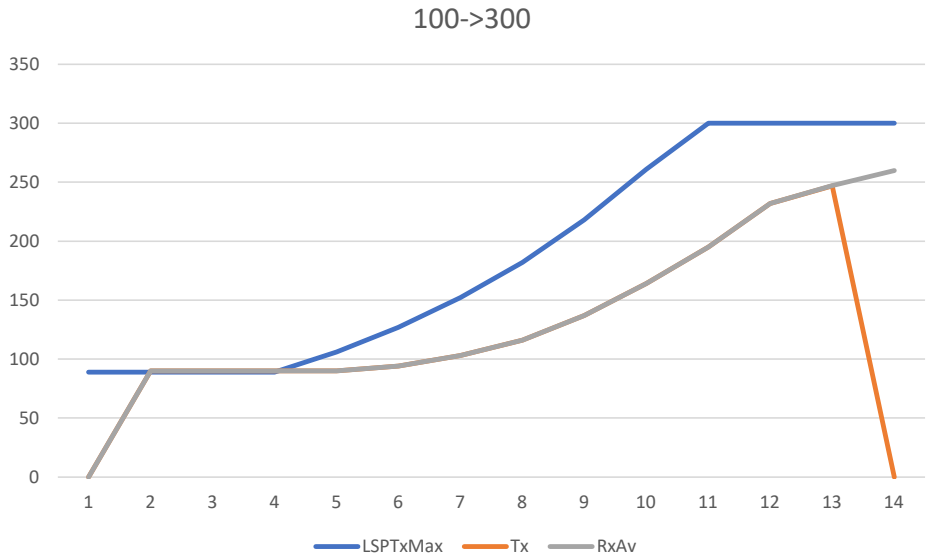
Test procedure:

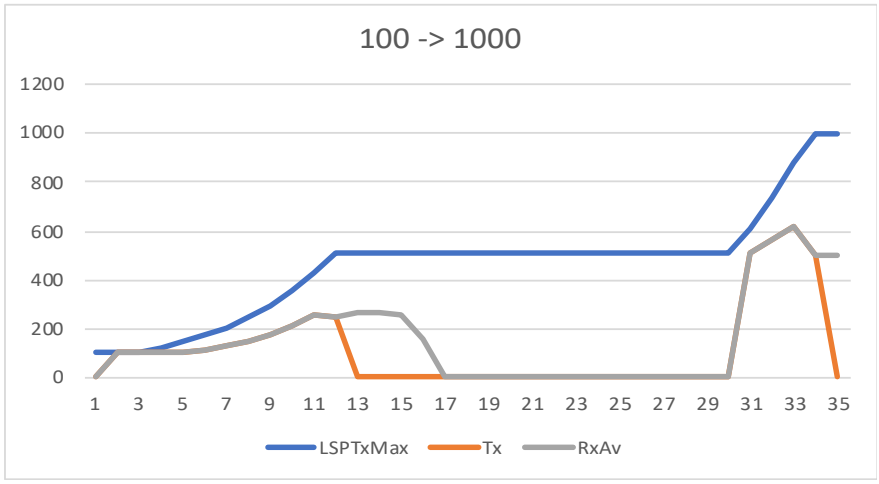
- 2000 LSPs
- Receiver's capability changed from 100 LSP/sec to no limit
- Sender has to detect change and adjust rate

Flooding	Start/End LSPTxRate	LSPTxMax	Time [ms]	Retrans
Base	33/33	NA	66528	0
Base	333/333	NA	7432	0
Optimized	100/300	300	11388	0
Optimized	100/1000	1000	10200/3072**	0

** Multiple burst needed

SpeedingUp


100->300
350
300
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14
LSPTxMax
Tx
RxAv

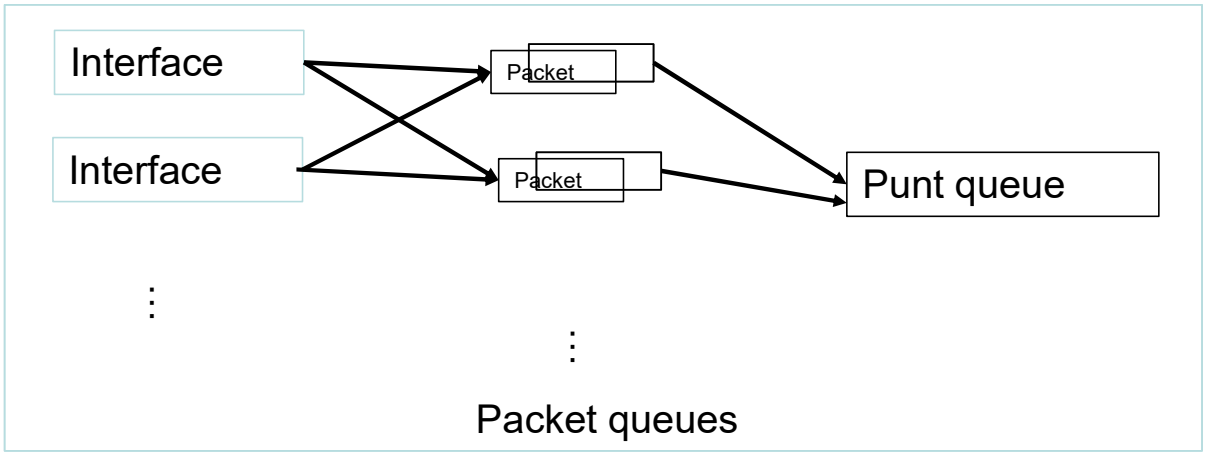

100 -> 1000
1200
1000
800
600
400
200
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35
LSPTxMax
Tx
RxAv

Future Plans

Future Plans:

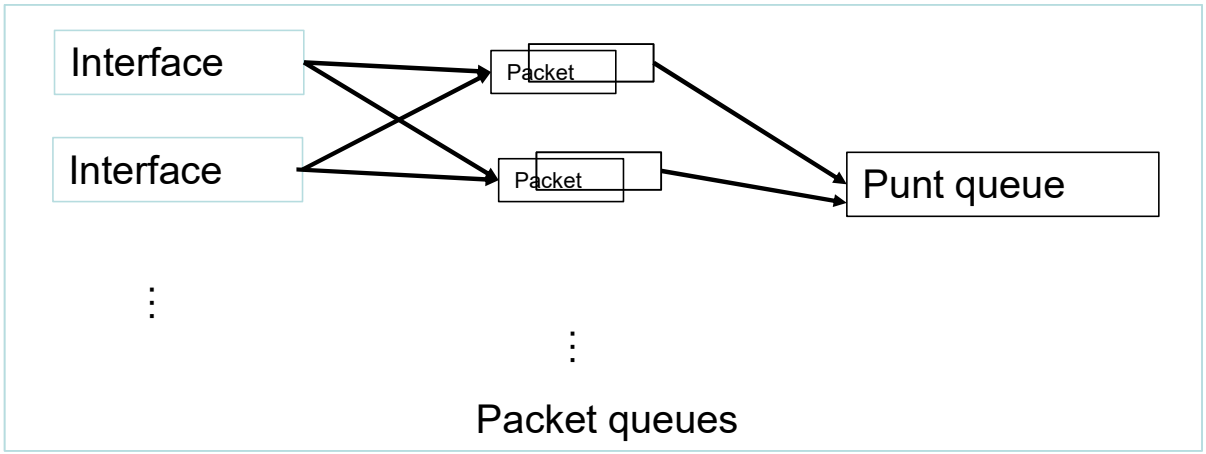
- Tuning Parameters to reduce Retransmissions on steep slowdown
- Testing: Large network, multiple neighbors, simultaneous sending/receiving

Router Dataplane Architecture



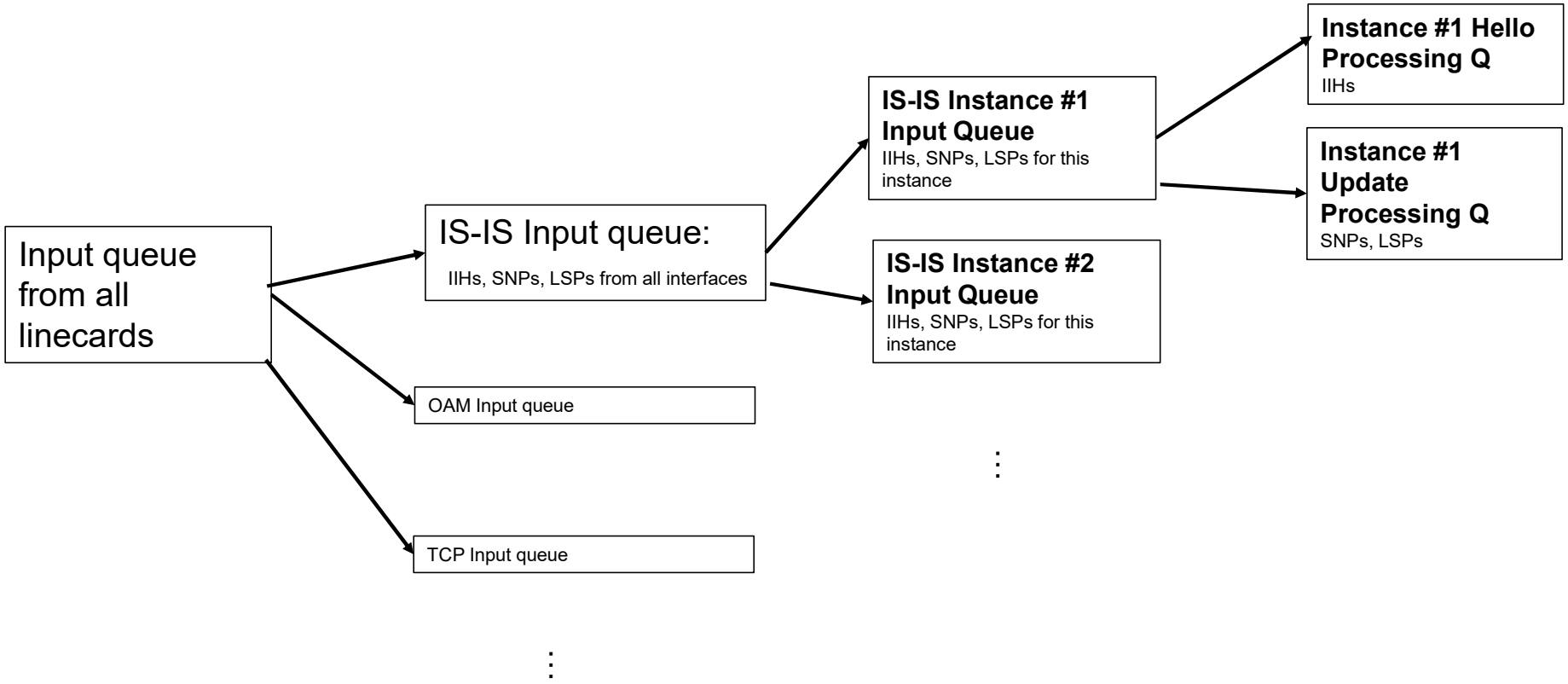
Each linecard has a set of interfaces.
Each interface collects packets to be sent to the control plane in a variety of packet queues based on packet types (NLPID, IP port #, etc.)
Each packet queue may have one type of packet or many types of packets.
The size of each packet queue is limited as well as the rate at which packets of a certain type can be accepted.
When multiple packet types go to a single queue, priorities/limits may be imposed/type.

Router Dataplane Architecture



Punt queue assembles packets from the various input queues to be sent to the control plane.
Size of the punt queue may be limited and the priority associated with each packet queue may vary.

Router Control Plane Punt Packet Processing for IS-IS



Advertisement of State by the Receiver

- Policed Input queue
  - Per Line card (not per interface)
  - May be per protocol or combine multiple protocols (“all routing”)
- Punt queue
  - May be per input queue or combine many input queues
- Control plane input queue (multiple line cards)
- IS-IS Input queue (IIHs, LSPs, SNPs) Multiple interfaces/Line Cards
  - Distribution to specific Instance
  - Separation of PDU types (Prioritization)

How does IS-IS in control plane stay aware of current state of all relevant elements in real time (sub-second)?

And map this per interface?

Other Factors Affecting Receiver Performance

- Number of neighbors
- # of nodes in the network
- Flooding optimizations supported (mesh groups, parallel neighbor suppression, dynamic flooding) by each neighbor
- Other features (BGP, BFD, OAM, link PM)
- CPU Scheduling
- Link bandwidth
- Hardware speed/memory
- …

Signaling In Real Time

Flooding burst is when fast flooding is needed…

During a burst both transmitter and receiver are busy
Nodes act as both transmitter and receiver simultaneously
Hellos and SNPs are unreliable – may be dropped
Signaling delays will increase likelihood of retransmissions

Bursts are short duration…a few seconds to a few 10s of seconds…
Not much time to adapt

Accurate realtime
signaling of current state of receiver is challenging

(We prefer not to rely on it)