Controling LSP Flooding Speed in IS-IS

a rebuttal to
draft-decraene-lsr-isis-flooding-speed

Les Ginsberg, Cisco
Peter Psenak , Cisco
Acee Lindem, Cisco

Key Proposals in

draft-decraene-lsr-isis-flooding-speed

New TLV in Hellos advertising:

- minimumInterfaceLSPReceptionInterval: the minimum interval, in milliseconds, between two LSPs the receiver expects the upstream node to send on a single interface

- maximumInterfaceLSPReceptionBurst: the maximum number of LSPs that the receiver expects the upstream node to transmit on a single interface with a separation interval shorter than minimumInterfaceLSPReceptionInterval (or even 'back to back').

Potentially adjust these numbers **dynamically** (but not too often) by updating the information in hellos.

What we agree On

Base specification flooding limits (one LSP/33 ms) is overly conservative for modern networks

Increasing flooding speeds entails some risk – therefore some means of flow control is needed

What we Disagree On

Flooding rate is functionally a per interface value (it isn’t)

Flow control should be done by the receiver (it shouldn’t be)

Flooding Rate and Convergence

Convergence is a network-wide behavior

Convergence depends on the timely distribution of LSPs network-wide (not just to neighbors)

Flooding at different rates on different interfaces – or even using different values/node will lead to longer convergence times => loops/blackholes for longer periods

Just as with SPF timers, inconsistent flooding rates will lengthen convergence

Interface Independent Flooding

From ISO 10589 7.3.14.3

> *"The Update Process scans the Link State Database for Link State PDUs with SRMflags set. When one is found, provided the timestamp lastSent indicates that it was propagated no more recently than minimumLSPTransmissionInterval, the IS shall*
>
> ***a) transmit it on all circuits with SRMflags** set, "*

Note the timer is per LSP (not per interface)

This is to maximize the probability that all nodes in the network receive the same LSP at the same time – which is what we want for convergence.

Receiver Driven Flow Control

LSP input queue implementations are typically interface independent FIFOs

Overloaded Receiver does not know which senders are disproportionately causing the overflow

LSPs may be dropped at lower layers – IS-IS receiver may be unaware that the overload condition exists

Updating hellos dynamically to alter flooding transmission rate is an OOB signaling mechanism consuming resources at a time when routers are the most busy

Consistent flooding rates will require updated hellos be sent to all neighbors – exacerbating the cost on both sender and receiver

Receiver Driven Flow Control(2)

Depends on protocol extensions – therefore requires upgrades before it is effective.

As we want consistent flooding rates on all interfaces/all nodes brownfield deployments are ineffective

Tx Based Flow Control(P2P)

LSPs which are to be flooded are marked per interface (SRM)

LSPs remain marked until acknowledged

Retransmit timer results in periodic retransmissions of unacknowledged LSPs

Based on the size of the “retransmission queue” sender knows neighbor has been unable to process the LSPs sent (and which ones)

This accounts for all reasons (drops before receive queueing, receiver overloaded CPU, etc.)

No protocol extensions required.

Example Flow Control Algo

MaxLSPTx = maximum # LSPs transmitted/second/interface
Umax = Maximum Unacknowledged LSP/Interface
Usafe = Safe level of Unacknowledged LSP/Interface
U(i) = # of unacknowledged LSPs previously transmitted/interface
LSPTx(i) = max # LSPs transmitted/second for a given interface

1)LSPTx(i) = MaxLSPTx

2)U(i) >= Umax (this should be logged)

- **only retransmissions of unacknowledged LSPs are performed**
- **LSPTx(i) = MaxLSPTx/2**

3)For each second U(i) >= Usafe

- **LSPTx(i) = LSPTx(i)/2**

4)When U <= Usafe

- **LSPTx(i) = MaxLSPTx**
- **new LSPs may be transmitted**

What has been done already…

Fast convergence introduced “fast-flooding” – sending a burst of LSPs prior to doing SPF

Implementations which support greater scale (large # of nodes) have reduced the flooding interval to speed up LSPDB synchronization following adjacency formation

Implementations have increased flooding rate

What should we do…

Encourage vendors to increase flooding rate.

Emphasize this needs to be done consistently on all nodes/links or convergence will suffer.

Use Tx based flow control to dampen flooding rate when necessary.

Do we need protocol extensions? NO!!

Do we need a BCP draft? MAYBE…