




## Current Research Directions in OSPF

Mukul Goyal



## Current Research Directions in OSPF


- Fast Convergence to topology changes in OSPF protocol
- Traffic Engineering with OSPF
- Graceful Restart



## Fast Convergence to Topology Changes

### The Convergence Process

- Detecting the topology change and generating new LSAs
- Flooding the LSAs throughout the network
- SPF calculations on receiving the new LSAs
- IGP routes Installed
- BGP routes are mapped onto IGP routes
- BGP routes installed



## Detecting the topology change

- Up events/down events
  - Speedy detection crucial for down events
- Failure Detection using Hello Protocol
  - Takes 3 to 4 Hello intervals (10 seconds default).
- Hardware Detection
  - Lower level protocols indicate the failure to the OSPF process.
  - Not always available. E.g. Switched Ethernet

## Detecting the topology change

How much can we reduce the HelloInterval to speed up the failure detection?

{PacketDesign1}: Reduce HelloIntervals to milliseconds range.

Can we really? What about the false alarms?

{HelloPriority}: Prioritized Treatment to Hello Packets.

Hello Protocol too complicated; Hellos bulky => Lightweight Heartbeat Protocol

A general purpose heartbeat protocol that can be used by any protocol.

Hello/heartbeat processing by the line card; don't bother the OSPF process.

Pros: reduced load on OSPF process

Cons: No fate sharing. What if OSPF process crashes?

## LSA Generation ([Katz])

When something changes, you have to tell the world  
Traditionally, generation delayed to collect multiple changes, then hold down to limit network traffic (on order of seconds)

More intelligent strategy is to rapidly announce interesting changes, allow several successive changes to be announced quickly before holddown

OSPF requires receivers to drop LSAs updated within five seconds (limiting senders is sufficient)

Suggestion--drop receiver behavior completely, use adaptive strategy on transmit

## Flooding The LSAs

Should not take much time intuitively but for

{PacketDesign1}: Bad implementations

Pacing Delays: Cisco routers forward 1 LS Update Packet per 30ms.

LSA Losses: Retransmission after 5 seconds.

Solutions:

Intelligent scheduling gives "interesting" link-state data flooding priority [katz]

Adaptive retransmission schemes can help when things get tough. [katz]

Flooding handled by line cards [villamizar]

## SPF Calculations On Receiving New LSAs

Limiting the frequency of SPF calculations:

spfDelay: delay between the first new LSA and the resulting SPF calculation.  
Default 5 seconds in Cisco routers.

spfHoldTime: delay between successive SPF calculations. Default 10 seconds in Cisco routers.

Useful in face of frequent topology changes from POV of stability. Bad for convergence.

Do we need to worry about too many SPF calculations?

Depends on whom you ask.

[Katz]: SPF should be able to run back-to-back all day long without threatening stability.

Goal: Fast Convergence AND not more than one SPF per topology change.

Solutions:

Setting spfDelay so that most of the LSAs are received before SPF calculation.

Exponential backoff for spfHoldTime

Correlating LSAs to detect topology changes? LSA triggered SPF calculations versus topology change triggered SPF calculations.

## SPF Calculation Times

Of the order  $(L \log_2 N)$  for L links and N nodes.

Impact of Network Topology (villamizar):

Full Mesh:  $L: N^2 \Rightarrow SPF: N \log_2 N$

Partial Mesh:  $L: 10N \Rightarrow SPF: 10N \log_2 N$

Full Mesh, no area,  $N=1000 \Rightarrow SPF: 10^4$

Partial Mesh, no area,  $N=1000 \Rightarrow SPF: 10^5$

Full Mesh, 100 routers per area,  $N=100 \Rightarrow SPF: 7 \times 10^4$

Partial Mesh, 100 routers per area,  $N=100 \Rightarrow SPF: 7 \times 10^5$

Say Multiplier is 1 us  $\Rightarrow 10^5 = 10s$  and  $7 \times 10^5 = 7ms$

Is there a case for incremental SPF calculations?

Yes [packetdesign1]

Incremental SPF only operates on the subtree downstream of the affected link and is usually much more efficient in all topologies except full mesh.

## Why should Convergence be Slow?

If detection, LSA generation, flooding and SPF can be made fast, why convergence should be slow?

[Katz] A change in IGP next hop may cause a next hop change in many thousands of BGP routes.

Traditionally done in software in order to produce a "flat" forwarding table

Indirect lookup in hardware has minimal forwarding time cost (essentially free if forwarding engine has any free cycles) with huge win in convergence time

## Current Research Directions in OSPF

Fast Convergence to topology changes in OSPF protocol

Traffic Engineering with OSPF

Graceful Restart

## Traffic Engineering with OSPF

Traffic Engineering (TE): performance optimization of operational networks, with the main focus on minimizing over-utilization of capacity when other capacity is available in the network.

OSPF Versus MPLS

OSPF: Hop by Hop routing, shortest paths based.

MPLS: Explicit routes; can consider current link utilizations while deciding routes for new traffic.

Why not MPLS?

New technology  $\Rightarrow$  Immature, there will be teething problems.



## Traffic Engineering with OSPF

Can we achieve traffic engineering goals with OSPF?

To some extent.

**Optimal General Routing:** The flow of each demand is optimally distributed over all paths between source and destination.

If the objective function is piece-wise linear, increasing and convex, we can solve the general routing problem optimally in polynomial time [Thorup1].

Can be formulated as a linear program and hence can be solved optimally in polynomial time.

Can a sufficiently clever weight settings make OSPF routing perform nearly as well as optimal general routing?

No. For an arbitrary  $n$ , we can construct an instance of the routing problem on  $n^2$  nodes where any OSPF routing has its average flow on arcs with utilization  $\epsilon$  times higher than the max utilization in an optimal general solution [Thorup1].

Optimizing the OSPF weights for a given set of demands is NP-hard. [Thorup1]

The condition that the traffic should be split equally among shortest paths can not be formulated as a linear program, which makes the problem NP-hard.

But local search can significantly improve the quality of weights over the weights set using oblivious methods (like weights proportional to distance or inv prop to capacity).

[Thorup2] says weights optimized for a set of key peak traffic matrices usually give good performance for all traffic matrices dominated by convex combination of these peaks.



## Using OSPF to distribute TE Information

Proposal for a new Traffic Engineering LSA: A type 10 opaque LSA. [TELSA]

Contains the following information regarding different links:

Traffic engineering metric (may be different from standard OSPF link cost)

Maximum bandwidth

Maximum reservable bandwidth (may be greater than maximum bandwidth)

Unreserved bandwidth (at 8 different priority levels)

Administrative group: A 4 octet bit mask with each bit representing a particular administrative group. The bit corresponding to a group is set if the interface belongs to it.

Generation of new TE LSAs depends on the implementation but should not be more frequent than 1 per 5 seconds.

The receipt of a new TE LSA does not trigger an SPF Calculation.



## Using OSPF in GMPLS Networks (as stuff distribution protocol)

GMPLS technology aims to have a common control framework for all network layers (IP, optical etc.).  
GMPLS = IP + Optical (Roughly).

GMPLS requirements proposes new information to be carried in TE LSAs [OSPF-GMPLS]:

Link protection type (Unprotected, shared, dedicated 1:1, dedicated 1+1 etc.)

Shared Risk Link Group(s) the link belongs to.

Interface Switching Capability Descriptor (Packet Switch Capable, Layer 2 switch capable, Time Division Multiplex capable, Lambda Switch capable, fiber switch capable)



## Current Research Directions in OSPF

Fast Convergence to topology changes in OSPF protocol

Traffic Engineering with OSPF

Graceful Restart

## Graceful Restart

Routing process needs to reboot often. With current OSPF specification, each reboot will bring down the router with corresponding generation of new LSAs, flooding, SPF, routing table updates etc. The process repeats when the routing process comes up.

Separation of control and forwarding functions. Certain processors are dedicated to control and management tasks such as OSPF routing, while other processors perform the data forwarding tasks.

Graceful Restart: Allowing an OSPF router can stay on the forwarding path even as its OSPF software is restarted.

If the topology is stable, there is no reason why current forwarding table should not be used while the routing process reboots.

## Graceful Restart

Graceful Restart: A router tells its neighbors – My routing process will reboot soon, please continue to use me for forwarding for next X seconds.

The neighbors do not consider the router to be down for next X seconds even if they don't receive any Hello from it.

While the router is rebooting and a topology change happens, the neighbors consider the rebooting router to be down to avoid any routing loops/black holes. [GracefulRestart]

If you are willing to do some extra calculations, you can find out if the use of rebooting router will cause loops/black holes if a topology change happens. In case, loops/blackholes won't occur, the neighbors can continue to use the rebooting router. [Shaikh]

## References

- [PacketDesign1]: C. Alaettinoglu, V. Jacobson and H. Yu, "Towards Millisecond IGP Convergence," NANOG 20  
<http://www.nanog.org/mtg-0010/gp.html>
- [HelloPriority]: G. Choudhary et al., "Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance," draft-ietf-ospf-scalability-05.txt  
<http://www.ietf.org/internet-drafts/draft-ietf-ospf-scalability-05.txt>
- [Katz]: D. Katz, "Why are we scared of SPF? IGP Scaling and Stability," NANOG 25  
<http://www.nanog.org/mtg-0206/katz.html>
- [Vilamizar]: C. Vilamizar, "Convergence and Restoration Techniques for ISP Interior Routing," NANOG 25  
<http://www.nanog.org/mtg-0206/curtis.html>
- [Thorup1]: B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," Infocom 2000  
<http://www.ieee-infocom.org/2000/papers/165.ps>
- [Thorup2]: B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a changing world," IEEE JSAC Special Issue on Advances in Fundamentals of Network Management, June 2002, infocom 2000  
[http://www.research.att.com/~mthorup/PAPER%20change\\_ospf.ps](http://www.research.att.com/~mthorup/PAPER%20change_ospf.ps)
- [GracefulRestart]: J. Moy et al., "Graceful OSPF Restart," draft-ietf-ospf-hitless-restart-08.txt  
<http://www.ietf.org/internet-drafts/draft-ietf-ospf-hitless-restart-08.txt>
- [Shaikh]: A. Shaikh, R. Dube and A. Verma, "Avoiding Instability During Graceful Shutdown of OSPF," INFOCOM 2002  
<http://www.ieee-infocom.org/2002/papers/713.pdf>

## References

- [TELSA]: D. Katz et al., "Traffic Engineering Extensions to OSPF Version 2," draft-katz-yeung-ospf-traffic-10.txt  
<http://www.ietf.org/internet-drafts/draft-katz-yeung-ospf-traffic-10.txt>
- [OSPF-GMPLS]: K. Kompella, Y. Rekhter, "OSPF Extensions in Support of Generalized MPLS," draft-ietf-ccamp-ospf-gmpls-extensions-09.txt  
<http://www.ietf.org/internet-drafts/draft-ietf-ccamp-ospf-gmpls-extensions-09.txt>