Seamless Network-Wide IGP Migrations

Laurent Vanbever, Stefano Vissicchio, Cristel Pelsser, Pierre Francois, and Olivier Bonaventure
laurent.vanbever@uclouvain.be

SIGCOMM
August 18, 2011
It is not the strongest of the species that survives, nor the most intelligent.

— Leon Megginson

(miss-attributed to Darwin)
It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change.

— Leon Megginson
(miss-attributed to Darwin)
Is there any reason to run IS-IS over OSPF in the service provider core? Currently, we are running IS-IS but we are redesigning our core and now would be a good time to switch.

I would like to switch to OSPF, mostly because of familiarity with OSPF over IS-IS.

What does everyone think?

NANOG thread, OSPF vs IS-IS, 11/08/11
Migrating the IGP is about network-wide reconfiguration

How do we get from here ...
Migrating the IGP is about network-wide reconfiguration

... to there?
Reconfiguring the IGP can provide immediate benefits to the network.

IGP reconfigurations can improve the:

- manageability
- performance
- stability
- security

of the entire network.
Migrating the IGP is operationally complex.

Reconfigure a running network while respecting Service Level Agreement.

Make highly distributed changes on all the routers, in a coordinated manner.

Face potential routing anomalies as non-migrated routers interact with migrated ones.
Current approaches do not entirely solve the problem

Reconfigure weights/links
Disruption free topology reconfiguration [Francois et al. INFOCOMM’2007]
Loop-free updates of forwarding tables [Fu et al. IEEE TNSM 2008, Shi et al. ICC’2009]
Graceful Network Operations [Raza et al. INFOCOMM’2009]

Modify the routers
Shadow Configuration [Alimi et al. SIGCOMM’2008]

Take advantage of virtualization
VROOM [Wang et al. SIGCOMM’2008]
BGP Grafting [Keller et al. NSDI’2010]
Problem Replace the anomaly-free IGP configuration of a running network, router-by-router, without causing any routing anomalies
Sub-problem 1

Current Practice

Run the two IGP configurations in parallel

Sub-problem 1

Replace the anomaly-free IGP configuration of a running network, router-by-router, without causing any routing anomalies
Migrating the IGP usually requires running two routing planes

Abstract model of a router

Control-plane

Initial IGP

At first, the initial IGP dictates the forwarding paths being used

Data-plane

Initial Forwarding paths
Migrating the IGP usually requires running two routing planes

Abstract model of a router

<table>
<thead>
<tr>
<th>Control-plane</th>
<th>Data-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial IGP</td>
<td>Initial</td>
</tr>
<tr>
<td>Final IGP</td>
<td>Forwarding paths</td>
</tr>
</tbody>
</table>

Then, the final IGP is introduced without changing the forwarding
Migrating the IGP usually requires running two routing planes

Abstract model of a router

Control-plane
- Initial IGP
- Final IGP

Data-plane
- Final Forwarding paths

After having converged, the final IGP is used by flipping the preference.
Migrating the IGP usually requires running two routing planes.

Abstract model of a router

- **Control-plane**
  - Initial IGP
  - Final IGP

- **Data-plane**
  - Final Forwarding paths

After having converged, the final IGP is used by flipping the preference.
Migrating the IGP usually requires running two routing planes.

Abstract model of a router:

- **Control-plane**: Final IGP
- **Data-plane**: Final Forwarding paths

The initial IGP is removed as it is not used anymore.
Sub-problem 1

Replace the anomaly-free IGP configuration of a running network, router-by-router, without causing any routing anomalies
Sub-problem 2    Replace the anomaly-free IGP configuration of a running network, router-by-router, without causing any routing anomalies
Migrating the IGP can create *migration loops*

A lot of networks experience loops

Up to 90 *migration loops* can arise during an IGP migration
Sub-problem 2: Replace the anomaly-free IGP configuration of a running network, router-by-router, without causing any routing anomalies.
Sub-problem 2  
Replace the anomaly-free IGP configuration of a running network, router-by-router, without causing any routing anomalies.

Contributions  
Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering.
Contributions: Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering, which one?
Contributions

1. Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering.
Contributions

1. Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering

2. Decide if an ordering exists is NP-complete
Contributions

1. Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering.

2. Decide if an ordering exists is NP-complete.

3. Develop an exponential algorithm as well as a heuristic to compute the ordering.
Contributions

1. Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering

2. Decide if an ordering exists is NP-complete

3. Develop an exponential algorithm as well as a heuristic to compute the ordering

4. Provide fallback solutions when no ordering exists
Contributions

1. Seamless IGP migration is possible as long as the reconfiguration process follows a *strict ordering*

2. Decide if an ordering exists is NP-complete

3. Develop an exponential algorithm as well as a heuristic to compute the ordering

4. Provide fallback solutions when no ordering exists

5. Outline solutions for link failures and congestion
Seamless IGP migration is possible as long as the reconfiguration process follows a strict ordering which one?
Seamless Network-Wide IGP Migrations

1. Identify the ordering
   Avoid anomalies

2. Compute the ordering
   Manage complexity

3. Apply the ordering
   Stable, efficient
Seamless Network-Wide IGP Migrations

1. Identify the ordering
   Avoid anomalies

   Compute the ordering
   Manage complexity

   Apply the ordering
   Stable, efficient
Reconfiguring the IGP might change the forwarding paths being used

In a flat IGP, routers forward traffic according to the shortest-path towards the destination.

In a flat IGP, R4 reaches R1 via R3
Reconfiguring the IGP might change the forwarding paths being used

In a hierarchical IGP, routers prefer paths contained within a single zone over the ones crossing several zones.

In a hierarchical IGP, R4 reaches R1 via R2.
Whenever the forwarding paths change, forwarding loops can be created.

Forwarding paths towards R1

initial paths

flat IS-IS

final paths

hierarchical OSPF
Forwarding paths towards R1

initial paths

flat IS-IS

intermediate paths

final paths

hierarchical OSPF

Forwarding paths towards R1
First, we migrate R3

Forwarding paths towards R1

- **initial paths**
  - flat IS-IS

- **intermediate paths**
  - hierarchical OSPF

- **final paths**
First, we migrate R3

Forwarding paths towards R1

flat IS-IS

initial paths

intermediate paths

final paths

hierarchical OSPF
Then, we migrate R4

Forwarding paths towards R1

<table>
<thead>
<tr>
<th>Flat IS-IS</th>
<th>Intermediate paths</th>
<th>Final paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 -- R2 -- R4</td>
<td>R1 -- R3 -- R4</td>
<td>R1 -- R3 -- R4</td>
</tr>
<tr>
<td>100 10 1</td>
<td>100 10 1</td>
<td>100 10 1</td>
</tr>
<tr>
<td>R2 -- R1</td>
<td>R3 -- R1</td>
<td>R3 -- R4</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

hierarchical OSPF
Then, we migrate R4

Forwarding paths towards R1

initial paths

flat IS-IS

intermediate paths

hierarchical OSPF

final paths
Whenever the forwarding paths change, forwarding loops can be created.

A loop is created if R4 is migrated before R2.

Forwarding paths towards R1:

- **Initial paths**: Flat IS-IS
- **Intermediate paths**: Hierarchical OSPF
- **Final paths**: Hierarchical OSPF
Migrations have to be performed following a precise ordering.

No loop arises if R2 is migrated before R4.

Forwarding paths towards R1:
- **Initial paths**: R1 <-> R2 <-> R4 <-> R1
- **Intermediate paths**: R1 <-> R3 <-> R4 <-> R1
- **Final paths**: R1 <-> R2 <-> R4 <-> R1

100 10 1
10 100 1
10 100 1
10 100 1

Flat IS-IS
Hierarchical OSPF
Migrations have to be performed following a precise ordering

No loop arises if R2 is migrated before R4

Forwarding paths towards R1

flat IS-IS

hierarchical OSPF
Migrations have to be performed following a precise ordering.

No loop arises if R2 is migrated before R4.

Forwarding paths towards R1:

- **Initial paths**: Flat IS-IS
  - R1 → R3: 10
  - R3 → R4: 10
  - R4 → R1: 100
  - R2 → R1: 1

- **Intermediate paths**: Hierarchical OSPF
  - R1 → R3: 10
  - R3 → R4: 10
  - R4 → R1: 100
  - R2 → R1: 1

- **Final paths**: Hierarchical OSPF
  - R1 → R3: 10
  - R3 → R4: 10
  - R4 → R1: 100
  - R2 → R1: 1
Seamless Network-Wide IGP Migrations

Identify the ordering
Avoid anomalies

2 Compute the ordering
Manage complexity

Apply the ordering
Stable, efficient
Finding and even deciding if an ordering exists is NP-complete

The Enumeration Algorithm [correct & complete]

1. Merge the initial and the final forwarding paths
2. For each migration loop in the merged graph, output ordering constraints such that at least one router in the initial state is migrated before at least one in the final
3. Solve the system by using Linear Programming
Finding and even deciding if an ordering exists is NP-complete

The Enumeration Algorithm [correct & complete]

1. Merge the initial and the final forwarding paths
2. For each migration loop in the merged graph,
   Output ordering constraints such that at least one router in the initial state is migrated before at least one in the final
3. Solve the system by using Linear Programming
Finding and even deciding if an ordering exists is NP-complete

The Enumeration Algorithm [correct & complete]

1. Merge the initial and the final forwarding paths
2. For each migration loop in the merged graph,
   Output ordering constraints such that at least one router in the initial state is migrated before at least one in the final
3. Solve the system by using Linear Programming

In every migration loop, at least one router is not migrated (R2) while at least one is migrated (R4, R3)
Finding and even deciding if an ordering exists is NP-complete

The Enumeration Algorithm [correct & complete]

1. Merge the initial and the final forwarding paths
2. For each migration loop in the merged graph,
   Output ordering constraints such that at least one router in the initial state is migrated before at least one in the final
3. Solve the system by using Linear Programming

Migrate R2 before R3 or R4 avoids the loop
In all the tested scenarios, the algorithm has found a solution.
More than 20% of the routers might be involved in the ordering
To deal with failures during the migration, time-efficient techniques are needed.

Failures can change the computed ordering as they modify the underlying IGP topology.

Solutions

- Precompute failover orderings
- Compute a new ordering when a failure is detected
To manage complexity, we implemented a correct, but not complete heuristic.

The heuristic is:

- based on sufficient, but not necessary condition
- migrate each router after all its successors
- one order of magnitude faster than the complete algorithm
The heuristic may not find a solution, even if it exists.

**Algorithm**

- Tested networks

**Heuristic**

- No solution found

Routers involved in ordering

0 30%
The heuristic involves more routers in the ordering than needed.
Seamless Network-Wide IGP Migrations

Identify the ordering
Avoid anomalies

Compute the ordering
Manage complexity

Apply the ordering
Stable, efficient
We implemented a provisioning system which automates the process

Network in which IGP 1 is replaced by IGP 2
First, the *Ordering Component* computes the ordering (if any)

Network in which IGP 1 is replaced by IGP 2
First, the *Ordering Component* computes the ordering (if any)

Network in which IGP 1 is replaced by IGP 2
Second, the *IGP Monitor* builds a dynamic view of the IGP and assesses its stability.
Third, the Configuration Manager introduces the, final configuration (not yet used) on all the routers.
Fourth, the final IGP’s completeness and stability are verified by the *IGP Monitor*.
Fifth, the *Configuration Manager* reconfigures each router – according to the ordering — so that it uses the final IGP.
Fifth, the *Configuration Manager* reconfigures each router – according to the ordering – so that it uses the final IGP.
Fifth, the *Configuration Manager* reconfigures each router – according to the ordering — so that it uses the final IGP.

Network in which IGP 1 is replaced by IGP 2.
Fifth, the *Configuration Manager* reconfigures each router – according to the ordering — so that it uses the final IGP.
Sixth, the IGP migration is over. The *Configuration Manager* removes the initial IGP configuration from each router.

Network in which IGP 1 is replaced by IGP 2.
Let’s reconfigure an existing network from a flat IGP ...

GEANT
European research network
36 routers
53 links
Let’s reconfigure an existing network from a flat IGP to a hierarchical IGP

GEANT
European research network
36 routers
53 links
Lossless reconfiguration is possible, by following the precomputed ordering

Traffic gets lost during more than 80% of the process

No loss occurs with proper ordering

Average results (50 repetitions) computed on 700+ pings per step from every router to 5 problematic destinations
Seamless Network-Wide IGP Migrations

1. Identify the ordering
   Avoid anomalies

2. Compute the ordering
   Manage complexity

3. Apply the ordering
   Stable, efficient
Don’t fear network reconfiguration, adapt the network to its environment.

Add flexibility in network management
seamlessly move to the current best configuration

Apply to other types of network migrations
that translate to a change of forwarding paths

Introduce a whole new family of problems
How do you reconfigure BGP, MPLS, multicast, etc.
Seamless Network-Wide IGP Migrations
towards more agile networking