Propane: Programming Distributed Control Planes

Ryan Beckett (Princeton, MSR)
Ratul Mahajan (MSR)
Todd Millstein (UCLA)
Jitu Padhye (MSR)
David Walker (Princeton)
Configuring Networks is Error-Prone

~60% of network downtime is caused by human error

-Yankee group 2002

50-80% of outages are the result of human error

-Juniper 2008
Configuring Networks is Error-Prone

By now, a lot of you have probably read about the Time Warner Cable outage yesterday, which caused a massive service disruption for millions of users across the US. The outage started coming in a little before 3:30 UTC, when many websites were inaccessible, DNS names failing to resolve for users attempting to access the Internet. Companies that paid Time Warner for its cable service, such as ESPN and Dish Network, were also affected.

China routing snafu impacts Cockup, not conspiracy
9 Apr 2010 at 12:24, John Leyden

Bad routing information sourced from China caused a series of global routing failures that impacted the Internet. These failures were related to the China Telecom network, which is owned by China's state-owned telecom giant China Telecom. The routing failures affected a large number of websites, including popular Chinese websites such as Baidu, Sohu, and Sina.

Although it seems that the China Telecom network was not the only one to suffer from routing issues, it is clear that the number of failures and the duration of the outage affected a significant number of users. The cause of the failures is still under investigation, but it is likely that they were related to a misconfiguration or a routing mistake.

One year ago today TTTNet in Turkey (AS9123) went offline on the Internet. And unfortunately for the rest of the Internet, network providers believed them (or at least as far as anyone knows, it was a mistake, not a configuration error). The consequences were far from benign: for some users, it was a complete system outage, with the rest of the Internet unavailable for some time. The event was later traced to a configuration mistake in the AS path, which resulted in the network being unable to connect to the Internet.

What is significant about the YouTube incident?

In light of Pakistan Telecom/YouTube incident, Internet registry official explains how you can avoid having your website victimized by such an attack.

When Pakistan Telecom blocked YouTube's traffic one Sunday evening in February, the ISP created an international incident that wreaked havoc on the popular video site for more than two hours.

RIPE NCC, the European registry for Internet addresses, has conducted an analysis of what happened during Pakistan Telecom's hijacking of YouTube's traffic and the steps that YouTube took to stop the attack.

We posed some questions to RIPE NCC's Chief Scientist Daniel Karrenberg about the YouTube incident. Here's what he had to say:

How frequently do hijacking incidents like the Pakistan Telecom/YouTube incident happen?

Misconfigurations of iBGP (internal BGP, the protocol used between the routers in the same Autonomous System) happen regularly and are usually the result of an error. One such misconfiguration caused the Pakistan Telecom/YouTube incident. It appears that the Pakistan Telecom/YouTube incident was not an "attack" as some have labeled it, but a configuration error. See Columnist John Till Johnson's take on the topic.

What is significant about the YouTube incident?
Objectives: Network-wide

- Prefer traffic go through AT&T over Sprint
- Don’t use our network as transit between A and B
- Traffic must stay within national boundaries
- Adhere to policies even when *failures* occur
Objectives: Network-wide

- Prefer traffic to go through AT&T over Sprint
- Don’t use our network as transit between A and B
- Traffic must stay within national boundaries
- Adhere to policies even when failures occur

Mechanisms: Device-by-Device
What about SDN?

Software Defined Networks

• Simpler, centralized programming models
• Network-wide abstractions like paths
• Centralized controller programs the network
What about SDN?

Software Defined Networks

- Simpler, centralized programming models
- Network-wide abstractions like paths
- Centralized controller programs the network

But they are not a panacea:

- Usually **intra-domain** routing only
- Poor **Latency**, and **scalability** for large networks
- Require careful design and engineering to be robust to **failures**
- Do not exploit **existing** network infrastructure
Fundamental Tradeoff?

Distributed

Centralized

Mechanism

Programming

Distributed

Centralized

Scalability
Low-latency
Inter-domain
Management
Configuration

Scalability
High-latency
Inter-domain
Management
Configuration
Fundamental Tradeoff?

![Diagram showing a matrix with two axes: Programming and Mechanism. The axes intersect at a 'Sweet Spot!' The top-right quadrant is labeled 'Centralized' and contains 'Scalability', 'Low-latency', 'Inter-domain', and 'Management Configuration'. The bottom-left quadrant is labeled 'Distributed' and contains 'Scalability', 'High-latency', 'Inter-domain', and 'Management Configuration'.]
Propane: Programming a Distributed Control Plane

I) Language for expressing high-level operator objectives with:

- Network-wide programming abstraction
- Uniform abstractions for intra- and inter-domain routing
- Paths constraints and relative preferences with fall-backs in case of failures
2) Compiler to generate a low-level distributed implementation:

- Efficient algorithms to synthesize a set of *policy-compliant* BGP configs
- Static analysis guarantees policy compliance under all failures
Example 1: A Data Center Network

Goals

• Local prefixes reachable only internally
• Global prefixes reachable externally
• Aggregate global prefixes as PG
• Prefer leaving through Peer1 over Peer2
• Prevent transit traffic between peers
Example 1: A Data Center Network

Goals

- Local prefixes reachable only internally
- Global prefixes reachable externally
- Aggregate global prefixes as PG
- Prefer leaving through Peer1 over Peer2
- Prevent transit traffic between peers

Attempt (I)

- Don’t export from G, H to external
- Aggregate externally as PG
Example 1: A Data Center Network

**Goals**
- Local prefixes reachable only internally
- Global prefixes reachable externally
- Aggregate global prefixes as PG
- Prefer leaving through Peer1 over Peer2
- Prevent transit traffic between peers

**Attempt (I)**
- Don’t export from G, H to external
- Aggregate externally as PG

![Diagram showing network with nodes labeled A, B, C, D, E, F, G, H, X, Y, PG1, PG2, PL1, PL2, Peer1, and Peer2. The diagram illustrates the network topology and traffic flow, with arrows indicating connectivity and direction.]
Example 1: A Data Center Network

Goals

- Local prefixes reachable only internally
- Global prefixes reachable externally
- Aggregate global prefixes as PG
- Prefer leaving through Peer1 over Peer2
- Prevent transit traffic between peers

Attempt (I)

- Don’t export from G, H to external
- Aggregate externally as PG
- X,Y block routes through each other
Example 1: A Data Center Network

Goals

- Local prefixes reachable only internally
- Global prefixes reachable externally
- Aggregate global prefixes as PG
- Prefer leaving through Peer1 over Peer2
- Prevent transit traffic between peers

Attempt (I)

- Don’t export from G, H to external
- Aggregate externally as PG
- \(X, Y\) block routes through each other
Example 1: A Data Center Network

**Goals**
- Local prefixes reachable only internally
- Global prefixes reachable externally
- Aggregate global prefixes as PG
- Prefer leaving through Peer1 over Peer2
- Prevent transit traffic between peers

**Attempt (I)**
- Don’t export from G, H to external
- Aggregate externally as PG
- X,Y block routes through each other
Example 1: A Data Center Network

Goals

• Local prefixes reachable only internally
• Global prefixes reachable externally
• Aggregate global prefixes as PG
• Prefer leaving through Peer1 over Peer2
• Prevent transit traffic between peers

Attempt (1)

• Don’t export from G, H to external
• Aggregate externally as PG
• X,Y block routes through each other

Aggregation-Induced Black Hole!
Example 1: A Data Center Network

Goals

• Local prefixes reachable only internally
• Global prefixes reachable externally
• Aggregate global prefixes as PG
• Prefer leaving through Peer1 over Peer2
• Prevent transit traffic between peers
Example 1: A Data Center Network

```
define Ownership =
{ PG1 => end(A)
  PG2 => end(B)
  PL1 => end(E)
  PL2 => end(F)
  true => exit(Peer1 >> Peer2) }
```
Example 1: A Data Center Network

\textbf{define Ownership} =
{PG1 => end(A),
 PG2 => end(B),
 PL1 => end(E),
 PL2 => end(F),
 true => exit(Peer1 >> Peer2)}

\textbf{define Locality} =
{PL1 | PL2 => always(in)}
Example 1: A Data Center Network

```
define Ownership =
{PG1 => end(A)
 PG2 => end(B)
 PL1 => end(E)
 PL2 => end(F)
 true => exit(Peer1 >> Peer2)}

define Locality =
{PL1 | PL2 => always(in)}

define NoTransit =
{true => !transit({Peer1, Peer2})}
```
Example 1: A Data Center Network

**define Ownership =**
{PG1 => end(A),
 PG2 => end(B),
 PL1 => end(E),
 PL2 => end(F),
 true => exit(Peer1 >> Peer2)}

**define Locality =**
{PL1 | PL2 => always(in)}

**define NoTransit =**
{true => !transit({Peer1, Peer2})}

**define Main =**
Ownership & Locality &
NoTransit & agg(PG, in => out)
Compilation

Front End Constraint Language

Regular Expression-based IR

Vendor-independent BGP

Failure Analyses

Vendor-specific configurations

Abstract BGP

Product Graph IR

Regular IR

Propane

Topology
Compilation Time

- Microsoft Data center & Backbone configurations (~40loc)
- Parallelize compilation by prefix

Data center (< 9 min)  
Backbone (< 3 min)
Configuration Size

- Perform configuration minimization during generation
- Avoid using community tags when choices are unambiguous
- Fall-through elimination of route maps
Propane: Summary

High-level language

- **Centralized** network programmability
- **Distributed** Implementation via BGP
- Same abstractions for both **Inter-domain** and **Intra-domain** routing

Compiler

- Automatically generates filters, preferences, community values, etc
- Static analysis guarantees policy compliance under **all failures**
- Lower bound on failures for aggregation-induced **black holes**
- **Scales** to reasonably sized network topologies