Last week on

Communication Networks
Internet routing

http://www.opte.org
Internet routing
from here to there, and back

1  Intra-domain routing
   Link-state protocols
   Distance-vector protocols

2  Inter-domain routing
   Path-vector protocols
Internet routing
from here to there, and back

1. Intra-domain routing
   - Link-state protocols
   - Distance-vector protocols

2. Inter-domain routing
   - Path-vector protocols
In Link-State routing, routers build a precise map of the network by flooding local views to everyone

Each router keeps track of its incident links and cost as well as whether it is up or down

Each router broadcast its own links state to give every router a complete view of the graph

Routers run Dijkstra on the corresponding graph to compute their shortest-paths and forwarding tables
During network changes, the link-state database of each node might differ. It is necessary for all nodes to have the same link-state database to ensure the global forwarding state validity, which directs packets to their destinations.
Inconsistencies lead to transient disruptions in the form of blackholes or forwarding loops.

Avoiding transient loops during the convergence of link-state routing protocols

Pierre François and Olivier Boudaventure
Université catholique de Louvain

Abstract—When using link-state protocols such as OSPF or IS-IS, forwarding loops can occur transiently when the routers adapt their forwarding tables as a response to a topological change. In this paper, we present a mechanism that lets the network converge to its optimal forwarding state without risking any transient loops and the related packet loss. The mechanism is based on an ordering of the updates of the forwarding tables of the routers. Our solution can be used in the case of a planned change in the state of a set of links and in the case of unpredictable changes when combined with a local protection scheme. The supported topology changes are link transitions from up to down, down to up, and updates of link metrics. Finally, we show by simulations that sub-second loop-free convergence is possible on a large Tier-1 ISP network.

I. INTRODUCTION

The link-state intra-domain routing protocols that are used in IP networks [2], [3] were designed when IP networks were research networks carrying best-effort packets. The same protocols are now used in large commercial ISPs with stringent SLAs. Furthermore, for most services, fast convergence in case of failures is a key problem that must be solved [4], [5]. Today, customers are requiring 99.999% reliability or better and providers try to minimize the time to recovery of links. Figure 1 shows the IGP topology of this network. Assume that the link between IP and KC fails but was protected by an MPLS tunnel between IP and KC via AT and HS. When AT receives a packet with destination DN, it forwards it to IP, which forwards it back to AT, but inside the protection tunnel, so that KC will decapsulate the packet, and forward it to its destination, DN.

This suboptimal routing should not last long, and thus after a
Distance-vector protocols are based on Bellman-Ford algorithm
Let $d_x(y)$ be the cost of the least-cost path known by $x$ to reach $y$

Each node bundles these distances into one message (called a vector) that it repeatedly sends to all its neighbors.

Each node updates its distances based on neighbors' vectors:

$$d_x(y) = \min \{ c(x,v) + d_v(y) \} \quad \text{over all neighbors } v$$
Unlike Link-State protocols, Distance-Vector protocols converge slowly
Internet routing
from here to there, and back

Intra-domain routing
Link-state protocols
Distance-vector protocols

2

Inter-domain routing
Path-vector protocols
The Internet is a network of networks, referred to as Autonomous Systems (AS)
BGP is the routing protocol “glueing” the Internet together
Using BGP, ASes exchange information about the IP prefixes they can reach, directly or indirectly.
BGP needs to solve three key challenges: scalability, privacy and policy enforcement

There is a huge # of networks and prefixes
600k prefixes, >50,000 networks, millions (!) of routers

Networks don’t want to divulge internal topologies or their business relationships

Networks needs to control where to send and receive traffic without an Internet-wide notion of a link cost metric
Link-State routing \textit{does not} solve these challenges

Floods topology information
high processing overhead

Requires each node to compute the entire path
high processing overhead

Minimizes some notion of total distance
works only if the policy is shared and uniform
Distance-Vector routing is on the right track

**Pros**

- Hide details of the network topology
- Nodes determine only “next-hop” for each destination
Distance-Vector routing is on the right track, but not really there yet…

**pros**
- Hide details of the network topology
- Nodes determine only “next-hop” for each destination

**cons**
- It still minimizes some common distance
- Impossible to achieve in an inter domain setting

It converges slowly
- Counting-to-infinity problem
BGP relies on **path-vector routing** to support flexible routing policies and avoid count-to-infinity

**key idea** advertise the **entire path** instead of distances
BGP announcements carry complete path information instead of distances.
Each AS appends itself to the path when it propagates announcements.
This week on Communication Networks
Border Gateway Protocol

policies and more

1. BGP Policies
   Follow the Money

2. Protocol
   How does it work?

3. Problems
   security, performance, ...
Border Gateway Protocol
policies and more

1  BGP Policies
   Follow the Money

Protocol
How does it work?

Problems
security, performance, ...
The Internet topology is shaped according to **business relationships**

![Diagram showing internet topology with AS numbers: AS10, AS20, AS30, AS40, AS50, skynet, Swisscom, Deutsche Telekom, ETH, and AT&T.](image)
2 ASes connect only if they have a business relationship
BGP is a “follow the money” protocol
There are 2 main business relationships today:

- customer/provider
- peer-peer

*many less important ones (siblings, backups,...)*
There are 2 main business relationships today:

- customer/provider
- peer/peer
Customers pay providers to get Internet connectivity
The amount paid is based on peak usage, usually according to the 95\textsuperscript{th} percentile rule.

Every 5 minutes, DT records the \# of bytes sent/received.

At the end of the month, DT:
- sorts all values in decreasing order
- removes the top 5\% values
- bills wrt highest remaining value
Most ISPs discounts traffic unit price when pre-committing to certain volume

<table>
<thead>
<tr>
<th>commit</th>
<th>unit price ($)</th>
<th>Minimum monthly bill ($/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Mbps</td>
<td>12</td>
<td>120</td>
</tr>
<tr>
<td>100 Mbps</td>
<td>5</td>
<td>500</td>
</tr>
<tr>
<td>1 Gbps</td>
<td>3.50</td>
<td>3,500</td>
</tr>
<tr>
<td>10 Gbps</td>
<td>1.20</td>
<td>12,000</td>
</tr>
<tr>
<td>100 Gbps</td>
<td>0.70</td>
<td>70,000</td>
</tr>
</tbody>
</table>

Examples taken from The 2014 Internet Peering Playbook
Internet Transit Prices have been continuously declining during the last 20 years.

The reason? Internet commoditization & competition.
There are 2 main business relationships today:

- customer/provider
- peer/peer
Peers don’t pay each other for connectivity, they do it \textit{out of common interest}.

DT and ATT exchange \textit{tons} of traffic. They save money by directly connecting to each other.
To understand Internet routing,
follow the money
Providers transit traffic for their customers.
Peers do not transit traffic between each other.
Customers do not transit traffic between their providers.
These policies are defined by constraining which BGP routes are *selected* and *exported*.

- **Selection**: which path to use?
- **Export**: which path to advertise?
which path to use?
control outbound traffic

which path to advertise?
always prefer Deutsche Telekom routes over AT&T
always prefer Deutsche Telekom routes over AT&T
Business relationships conditions

*route selection*

For a destination $p$, prefer routes coming from

- customers over
- peers over
- providers

$route\ type$
Selection

which path to use?

Export

which path to advertise?
control inbound traffic
do not export ETH routes to AT&T
do not export ETH routes to AT&T
Business relationships conditions

route exportation

send to

customer peer provider

customer

from peer

provider
Routes coming from customers are propagated to everyone else.

<table>
<thead>
<tr>
<th>send to</th>
<th>customer</th>
<th>peer</th>
<th>provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>from peer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>provider</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Routes coming from peers and providers are only propagated to customers

<table>
<thead>
<tr>
<th></th>
<th>send to</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer</td>
<td>✓</td>
</tr>
<tr>
<td>peer</td>
<td>✓</td>
</tr>
<tr>
<td>provider</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>from</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>customer</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>peer</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>provider</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Selection

which path to use?
control outbound traffic

Export

which path to advertise?
control inbound traffic
Is (B, A, D) a valid path? Yes/No
Is (H, E, D) a valid path?  
Yes/No
Is (G,D,A,B,E,H) a valid path?  Yes/No
Will \((G, D, A, B, E, H)\) actually see packets?  Yes/No
What's a valid path between G and I?
What’s a valid path between G and I?
Border Gateway Protocol

policies and more

BGP Policies
Follow the Money

Protocol
How does it work?

Problems
security, performance, …
BGP sessions come in two flavors
external BGP (eBGP) sessions connect border routers in different ASes
eBGP sessions are used to learn routes to external destinations

129.132.0.0/16
Path: 20
internal BGP (iBGP) sessions connect the routers in the same AS
iBGP sessions are used to disseminate externally-learned routes internally.
I can reach “129.132/16” via **SEAT**, internal NH is **CHIC**

learned via IGP (*e.g.*, OSPF)
Routes disseminated internally are then announced externally again, using eBGP sessions.

129.132.0.0/16
Path: 10 20
On the wire, BGP is a rather simple protocol composed of four basic messages:

<table>
<thead>
<tr>
<th>Type</th>
<th>Used to</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN</td>
<td>establish TCP-based BGP sessions</td>
</tr>
<tr>
<td>NOTIFICATION</td>
<td>report unusual conditions</td>
</tr>
<tr>
<td>UPDATE</td>
<td>inform neighbor of a new best route</td>
</tr>
<tr>
<td></td>
<td>a change in the best route</td>
</tr>
<tr>
<td></td>
<td>the removal of the best route</td>
</tr>
<tr>
<td>KEEPALIVE</td>
<td>inform neighbor that the connection is alive</td>
</tr>
</tbody>
</table>
UPDATE

inform neighbor of a new best route
a change in the best route
the removal of the best route
BGP UPDATEs carry an IP prefix together with a set of attributes
BGP UPDATEs carry an IP prefix together with a set of attributes

- **IP prefix**
- **Attributes**
  - Describe route properties used in route selection/exportation decisions
  - are either local \((\text{only seen on iBGP})\) or global \((\text{seen on iBGP and eBGP})\)
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT-HOP</td>
<td>egress point identification</td>
</tr>
<tr>
<td>AS-PATH</td>
<td>loop avoidance</td>
</tr>
<tr>
<td></td>
<td>outbound traffic control</td>
</tr>
<tr>
<td></td>
<td>inbound traffic control</td>
</tr>
<tr>
<td>LOCAL-PREF</td>
<td>outbound traffic control</td>
</tr>
<tr>
<td>MED</td>
<td>inbound traffic control</td>
</tr>
</tbody>
</table>
The **NEXT-HOP** is a global attribute which indicates where to send the traffic next
The NEXT-HOP is set when the route enters an AS, it does not change within the AS.
The **AS–PATH** is a global attribute that lists all the ASes a route has traversed *(in reverse order)*.
The **LOCAL-PREF** is a *local* attribute set at the border, it represents how “preferred” a route is
set LOCAL-PREF to 50

set LOCAL-PREF to 100
By setting a higher LOCAL-PREF, all routers end up using DT to reach any external prefixes, even if they are closer (IGP-wise) to the Swisscom egress.
The **MED** is a *global* attribute which encodes the relative “proximity” of a prefix wrt to the announcer.
Swisscom receives two routes to reach $p$
Swisscom receives two routes to reach $p$ and chooses (arbitrarily) its left router as egress.
Yet, ETH would prefer to receive traffic for $p$
on its right border router which is closer to the actual destination.
ETH can communicate that preferences to Swisscom by setting a higher MED on $p$ when announced from the left.
Swisscom receives two routes to reach $p$ and, given it does not cost it anything more, chooses its right router as egress.
Swisscom receives two routes to reach $p$ and, given it does not cost it anything more, chooses its right router as egress.

But what if it does?
Consider that Swisscom always prefer to send traffic via its left egress point (bigger router, less costly)
In this case, Swisscom will not care about the MED value and still push the traffic via its left router.
Lesson

The network which is sending the traffic **always** has the final word when it comes to deciding where to forward

Corollary

The network which is receiving the traffic can just **influence** remote decision, not control them
With the MED, an AS can influence its inbound traffic between multiple connection towards the same AS.

ETH cannot use the MED to move incoming traffic to Swisscom.

\[ p: 82.130.64.0/18 \]
BGP UPDATEs carry an IP prefix together with a set of attributes

- **IP prefix**
- **Attributes**
  - Describe route properties
  - used in route selection/exportation decisions
  - are either local (only seen on iBGP)
  - or global (seen on iBGP and eBGP)
Each BGP router processes UPDATEs according to a precise pipeline
All acceptable routes

BGP Decision Process

Best route to each destination

forwarding entries

IP forwarding table
Given the set of all acceptable routes for each prefix, the BGP Decision process elects a single route. BGP is often referred to as a single path protocol.
Prefer routes…

with higher LOCAL-PREF

with shorter AS-PATH length

with lower MED

learned via eBGP instead of iBGP

with lower IGP metric to the next-hop

with smaller egress IP address (tie-break)
learned via eBGP instead of iBGP

with lower IGP metric to the next-hop
These two steps aim at directing traffic as quickly as possible out of the AS (early exit routing)
ASes are selfish

They dump traffic as soon as possible to someone else

This leads to asymmetric routing

Traffic does not flow on the same path in both directions

[Diagram showing multiple peering points with arrows indicating traffic flow]
Let’s look at how operators implement customer/provider and peer policies in practice
To implement their selection policy, operators define input filters which manipulate the LOCAL-PREF

For a destination $p$, prefer routes coming from

- customers over
- peers over $route$ type
- providers
input filter:
match *, set LP := 50

input filter:
match *, set LP := 100

input filter:
match *, set LP := 200

AS 40
provider

AS 30
peer

AS10

AS 20
customer
To implement their exportation rules, operators use a mix of import and export filters

<table>
<thead>
<tr>
<th>from</th>
<th>customer</th>
<th>peer</th>
<th>provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>peer</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>provider</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(send to)

<table>
<thead>
<tr>
<th>send to</th>
<th>customer</th>
<th>peer</th>
<th>provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
input filter:
  match *, set TAG := PEER
output filter:
  match TAG := CUST, allow
  else deny

input filter:
  match *, set TAG := PROV
output filter:
  match TAG := CUST, allow
  else deny

input filter:
  match *, set TAG := CUST
output filter:
  match TAG := *, allow

input filter:
  match *, set TAG := CUST
output filter:
  match TAG := *, allow

Border Gateway Protocol
policies and more

BGP Policies
Follow the Money

Protocol
How does it work?

Problems
security, performance, …
BGP suffers from many rampant problems

Problems

Reachability
Security
Convergence
Performance
Anomalies
Relevance
<table>
<thead>
<tr>
<th>Problems</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Convergence</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td>Anomalies</td>
</tr>
<tr>
<td></td>
<td>Relevance</td>
</tr>
</tbody>
</table>
Unlike normal routing, policy routing does not guarantee reachability even if the graph is connected.

Because of policies, Swisscom cannot reach DT even if the graph is connected.
Problems

Reachability

Security

Convergence

Performance

Anomalies

Relevance
Many security considerations are simply absent from BGP specifications.

ASes can advertise any prefixes even if they don’t own them!

ASes can arbitrarily modify route content e.g., change the content of the AS-PATH

ASes can forward traffic along different paths than the advertised one
We’ll do a deep dive into BGP security next week
Problems

Reachability

Security

Convergence

Performance

Anomalies

Relevance
With arbitrary policies, BGP may fail to converge
preference list

1 prefers to reach 0 via 3 rather than directly.
Initially, all ASes only know the direct route to 0
AS 1 advertises its path to AS 2
Upon reception,
AS 2 switches to 2 1 0 (preferred)
AS 3 advertises its path to AS 1
Upon reception, AS 1 switches to 1 3 0 (preferred)
AS 1 advertises its new path 1 3 0 to AS 2
Upon reception, AS 2 reverts back to its initial path 2 0
AS 2 advertises its path 2 0 to AS 3
Upon reception, AS 3 switches to 3 2 0 (preferred)
AS 3 advertises its new path 3 2 0 to AS 1
Upon reception,
AS 1 reverts back to 1 0 (initial path)
AS 1 advertises its new path 1 0 to AS 2
Upon reception, AS 2 switches to 2 1 0 (preferred)
AS 2 advertises its new path 2 1 0 to AS 3
Upon reception, AS 3 switches to its initial path 3 0
We are back where we started, from there on, the oscillation will continue forever
Policy oscillations are a direct consequence of policy autonomy

ASes are free to choose and advertise any paths they want. Network stability argues against this.

Guaranteeing the absence of oscillations is hard, even when you know all the policies!
 Guaranteeing the absence of oscillations is hard even when you know all the policies!

How come?
Theorem

Computationally, a BGP network is as “powerful” as

see “Using Routers to Build Logic Circuits: How Powerful is BGP?”
How do you prove such a thing?
How do you prove such a thing?

Easy, you build a computer using BGP…
Logic gates

\[ i_1 \quad \text{OR} \quad o \quad + \quad i \quad \text{NOT} \quad o \]
Logic gates

\[ i_1 \rightarrow \text{OR} \rightarrow o + i \rightarrow \text{NOT} \rightarrow o + \]

Memory

\[ R \rightarrow \text{NOR} \rightarrow Q \]
\[ S \rightarrow \text{NOR} \rightarrow \bar{Q} \]
Logic gates

\[ i_1 \quad \text{OR} \quad i_2 \quad o \quad + \quad i \quad \text{NOT} \quad o \quad + \quad \text{Memory} \quad + \quad \text{Clock} \]
BGP has it all!
BGP has it all!

famous incorrect BGP configurations (Griffin et al.)
Instead of using Minecraft for building a computer... use BGP!

Hack III, Minecraft's largest computer to date
Together, BGP routers form the largest computer in the world!

Router-level view of the Internet, OPTE project
Checking BGP correctness is as hard as checking a general program

Theorem 1
Determining whether a finite BGP network converges is PSPACE-hard

Theorem 2
Determining whether an infinite BGP network converges is Turing-complete
Theorem
If all AS policies follow the cust/peer/provider rules,
BGP is guaranteed to converge.

Intuition
Oscillations require “preferences cycles”
which make no economical sense.

In practice though,
BGP does not oscillate that often.

known as “Gao-Rexford” rules
Problems

Reachability

Security

Convergence

Performance

Anomalies

Relevance
BGP path selection is mostly economical, not based on accurate performance criteria.
<table>
<thead>
<tr>
<th>Problems</th>
<th>Reachability</th>
<th>Security</th>
<th>Convergence</th>
<th>Performance</th>
<th>Anomalies</th>
<th>Relevance</th>
</tr>
</thead>
</table>

BGP configuration is hard to get right, you’ll understand that very soon

BGP is both “bloated” and underspecified
lots of knobs and (sometimes, conflicting) interpretations

BGP is often manually configured
humans make mistakes, often

BGP abstraction is fundamentally flawed
disjoint, router-based configuration to effect AS-wide policy
“Human factors are responsible for 50% to 80% of network outages”

Problems

Reachability
Security
Convergence
Performance
Anomalies
Relevance
The world of BGP policies is rapidly changing

ISPs are now eyeballs talking to content networks
*e.g.*, Swisscom and Netflix/Spotify/YouTube

Transit becomes less important and less profitable
traffic move more and more to interconnection points

No systematic practices, yet
details of peering arrangements are private anyway
Border Gateway Protocol
policies and more

BGP Policies
Follow the Money

Protocol
How does it work?

Problems
security, performance, …
Internet Hackathon
April 12 @6pm in ETZ hall
2016 edition