Internet Hackathon

This Thursday @6pm in ETZ hall + ETZ E6

2016 edition
No exercise session this Thursday
Q&A session today (April 1)

3pm to 5pm in ETZ G71.2

and online on #routing_project
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

   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.
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:

$$d_x(y) = \min \{ c(x,v) + d_v(y) \} \quad \text{over all neighbors } v$$
Internet routing
from here to there, and back

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

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
700k 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 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**
Intuition

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

Deutsche Telekom

Provider

$$$

Customer

swisscom
The amount paid is based on peak usage, usually according to the 95th 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.

<table>
<thead>
<tr>
<th>Year</th>
<th>Internet Transit Price</th>
<th>% decline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>$1,200.00 per Mbps</td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>$800.00 per Mbps</td>
<td>33%</td>
</tr>
<tr>
<td>2000</td>
<td>$675.00 per Mbps</td>
<td>16%</td>
</tr>
<tr>
<td>2001</td>
<td>$400.00 per Mbps</td>
<td>41%</td>
</tr>
<tr>
<td>2002</td>
<td>$200.00 per Mbps</td>
<td>50%</td>
</tr>
<tr>
<td>2003</td>
<td>$120.00 per Mbps</td>
<td>40%</td>
</tr>
<tr>
<td>2004</td>
<td>$90.00 per Mbps</td>
<td>25%</td>
</tr>
<tr>
<td>2005</td>
<td>$75.00 per Mbps</td>
<td>17%</td>
</tr>
<tr>
<td>2006</td>
<td>$50.00 per Mbps</td>
<td>33%</td>
</tr>
<tr>
<td>2007</td>
<td>$25.00 per Mbps</td>
<td>50%</td>
</tr>
<tr>
<td>2008</td>
<td>$12.00 per Mbps</td>
<td>52%</td>
</tr>
<tr>
<td>2009</td>
<td>$9.00 per Mbps</td>
<td>25%</td>
</tr>
<tr>
<td>2010</td>
<td>$5.00 per Mbps</td>
<td>44%</td>
</tr>
<tr>
<td>2011</td>
<td>$3.25 per Mbps</td>
<td>35%</td>
</tr>
<tr>
<td>2012</td>
<td>$2.34 per Mbps</td>
<td>28%</td>
</tr>
<tr>
<td>2013</td>
<td>$1.57 per Mbps</td>
<td>33%</td>
</tr>
<tr>
<td>2014</td>
<td>$0.94 per Mbps</td>
<td>40%</td>
</tr>
<tr>
<td>2015</td>
<td>$0.63 per Mbps</td>
<td>33%</td>
</tr>
</tbody>
</table>

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 *out of common interest*

DT and ATT exchange *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

IP traffic
Business relationships conditions

*route selection*

For a destination $p$, prefer routes coming from

- customers over
- peers over
- providers

(route type)
which path to use?

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.
Routes coming from peers and providers are only propagated to customers

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

which path to use?
control outbound traffic

Export

which path to advertise?
control inbound traffic
provider

AS A

customer

AS D
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?  \(\text{Yes/No}\)
Will (G,D,A,B,E,H) actually see packets?  Yes/No
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.
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>inform neighbor that the connection is alive</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>
</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 (only seen on iBGP)
  - or global (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

Provider #1 ($$)

swisscom

Provider #2 ($)

Deutsche Telekom

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.

\[ \text{p: 82.130.64.0/18} \]
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, \textit{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)

big router
set LP to 200

smaller router, set LP to 50

set MED to 20

set MED to 10

p: 82.130.64.0/18
In this case, Swisscom will not care about the MED value and still push the traffic via its left router.

- Big router: set LP to 200
- Smaller router: set LP to 50
- Set MED to 20
- Set MED to 10
- p: 82.130.64.0/18
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.
BGP UPDATEs carry an IP prefix together with a set of attributes.

- **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

IP forwarding table
Given the set of all acceptable routes for each prefix, the BGP Decision process elects a single route.
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
Let’s look at how operators implement customer/provider and peer policies in practice
To implement their selection policy, operators define input filters which manipulates the LOCAL-PREF

For a destination $p$, prefer routes coming from

- customers over peers over providers

route type
input filter:
match *, set LP := 200

input filter:
match *, set LP := 100

input filter:
match *, set LP := 50

AS10

AS 40 provider

AS 30 peer

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>customer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<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)
input filter:
match *, set TAG := PEER
output filter:
match TAG := CUST, allow else deny

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

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

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

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

Reachability

Security

Convergence

Performance

Anomalies

Relevance
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
BGP (lack of) security

#1 BGP does not validate the origin of advertisements

#2 BGP does not validate the content of advertisements
BGP (lack of) security

#1 BGP does not validate the origin of advertisements

#2 BGP does not validate the content of advertisements
IP Address Ownership and Hijacking

• IP address block assignment
  – Regional Internet Registries (ARIN, RIPE, APNIC)
  – Internet Service Providers

• Proper origination of a prefix into BGP
  – By the AS who owns the prefix
  – ... or, by its upstream provider(s) in its behalf

• However, what’s to stop someone else?
  – Prefix hijacking: another AS originates the prefix
  – BGP does not verify that the AS is authorized
  – Registries of prefix ownership are inaccurate
Prefix Hijacking

- **Blackhole**: data traffic is discarded
- **Snooping**: data traffic is inspected, then redirected
- **Impersonation**: traffic sent to bogus destinations
Hijacking is Hard to Debug

• The victim AS doesn’t see the problem
  – Picks its own route, might not learn the bogus route

• May not cause loss of connectivity
  – Snooping, with minor performance degradation

• Or, loss of connectivity is isolated
  – E.g., only for sources in parts of the Internet

• Diagnosing prefix hijacking
  – Analyzing updates from many vantage points
  – Launching traceroute from many vantage points
Sub-Prefix Hijacking

- Originating a more-specific prefix
  - Every AS picks the bogus route for that prefix
  - Traffic follows the longest matching prefix

12.34.0.0/16

12.34.158.0/24
How to Hijack a Prefix

• The hijacking AS has
  – Router with BGP session(s)
  – Configured to originate the prefix

• Getting access to the router
  – Network operator makes configuration mistake
  – Disgruntled operator launches an attack
  – Outsider breaks in to the router and reconfigures

• Getting other ASes to believe bogus route
  – Neighbor ASes do not discard the bogus route
  – E.g., not doing protective filtering
YouTube Outage on Feb 24, 2008

• YouTube (AS 36561)
  – Web site www.youtube.com (208.65.152.0/22)
• Pakistan Telecom (AS 17557)
  – Government order to block access to YouTube
  – Announces 208.65.153.0/24 to PCCW (AS 3491)
  – All packets to YouTube get dropped on the floor
• Mistakes were made
  – AS 17557: announce to everyone, not just customers
  – AS 3491: not filtering routes announced by AS 17557
• Lasted 100 minutes for some, 2 hours for others
Timeline (UTC Time)

• 18:47:45
  – First evidence of hijacked /24 route in Asia

• 18:48:00
  – Several big trans-Pacific providers carrying the route

• 18:49:30
  – Bogus route fully propagated

• 20:07:25
  – YouTube starts advertising /24 to attract traffic back

• 20:08:30
  – Many (but not all) providers are using valid route
Timeline (UTC Time)

- **20:18:43**
  - YouTube announces two more-specific /25 routes

- **20:19:37**
  - Some more providers start using the /25 routes

- **20:50:59**
  - AS 17557 starts prepending (“3491 17557 17557”)

- **20:59:39**
  - AS 3491 disconnects AS 17557

- **21:00:00**
  - Videos of cats flushing toilets are available again!
Another Example: Spammers

- Spammers sending spam
  - Form a (bidirectional) TCP connection to mail server
  - Send a bunch of spam e-mail, then disconnect
- But, best not to use your real IP address
  - Relatively easy to trace back to you
- Could hijack someone’s address space
  - But you might not receive all the (TCP) return traffic
- How to evade detection
  - Hijack unused (i.e., unallocated) address block
  - Temporarily use the IP addresses to send your spam
BGP (lack of) security

#1 BGP does not validate the origin of advertisements

#2 BGP does not validate the content of advertisements
Bogus AS Paths

• Remove ASes from the AS path
  – E.g., turn “701 3715 88” into “701 88”

• Motivations
  – Attract sources that normally try to avoid AS 3715
  – Help AS 88 look like it is closer to the Internet’s core

• Who can tell that this AS path is a lie?
  – Maybe AS 88 does connect to AS 701 directly
Bogus AS Paths

• Add ASes to the path
  – E.g., turn “701 88” into “701 3715 88”

• Motivations
  – Trigger loop detection in AS 3715
    • Denial-of-service attack on AS 3715
    • Or, blocking unwanted traffic coming from AS 3715!
  – Make your AS look like it has richer connectivity

• Who can tell the AS path is a lie?
  – AS 3715 could, if it could see the route
  – AS 88 could, but would it really care?
Bogus AS Paths

• Adds AS hop(s) at the end of the path
  – E.g., turns “701 88” into “701 88 3”

• Motivations
  – Evade detection for a bogus route
  – E.g., by adding the legitimate AS to the end

• Hard to tell that the AS path is bogus...
  – Even if other ASes filter based on prefix ownership
Invalid Paths

• **AS exports a route it shouldn’t**
  – AS path is a valid sequence, but violated policy

• **Example: customer misconfiguration**
  – Exports routes from one provider to another

• **Interacts with provider policy**
  – Provider prefers customer routes
  – Directing all traffic through customer

• **Main defense**
  – Filtering routes based on prefixes and AS path
Missing/Inconsistent Routes

• Peers require consistent export
  – Prefix advertised at all peering points
  – Prefix advertised with same AS path length

• Reasons for violating the policy
  – Trick neighbor into “cold potato”
  – Configuration mistake

• Main defense
  – Analyzing BGP updates, or traffic,
  – ... for signs of inconsistency
BGP Security Today

• Applying best common practices (BCPs)
  – Securing the session (authentication, encryption)
  – Filtering routes by prefix and AS path
  – Packet filters to block unexpected control traffic

• This is not good enough
  – Depends on vigilant application of BCPs
  – Doesn’t address fundamental problems
    • Can’t tell who owns the IP address block
    • Can’t tell if the AS path is bogus or invalid
    • Can’t be sure the data packets follow the chosen route
Routing attacks can be used to de-anonymize Tor users

RAPTOR: Routing Attacks on Privacy in Tor

Yixin Sun  Anne Edmundson  Laurent Vanbever  Oscar Li
Princeton University  Princeton University  ETH Zurich  Princeton University
Jennifer Rexford  Mung Chiang  Prateek Mittal
Princeton University  Princeton University  Princeton University

Abstract

The Tor network is a widely used system for anonymous communication. However, Tor is known to be vulnerable to attackers who can observe traffic at both ends of the communication path. In this paper, we show that prior attacks are just the tip of the iceberg. We present a suite of new attacks, called Raptor, that can be launched by Autonomous Systems (ASes) to compromise user anonymity. First, AS-level adversaries can exploit the asymmetric nature of Internet routing to increase the chance of observing at least one direction of user traffic at both ends of the communication. Second, AS-level adversaries can exploit natural churn in Internet routing to lie on the BGP paths for more users over journalists, businesses and ordinary citizens concerned about the privacy of their online communications [9].

Along with anonymity, Tor aims to provide low latency and, as such, does not obfuscate packet timings or sizes. Consequently, an adversary who is able to observe traffic on both segments of the Tor communication channel (i.e., between the server and the Tor network, and between the Tor network and the client) can correlate packet sizes and packet timings to de-anonymize Tor clients [45, 46].

There are essentially two ways for an adversary to gain visibility into Tor traffic, either by compromising (or owning enough) Tor relays or by manipulating the underlying network communications so as to put herself on the forwarding path for Tor traffic. Regarding net-
Routing attacks can be used to partition the Bitcoin network

Hijacking Bitcoin: Routing Attacks on Cryptocurrencies

Maria Apostolaki
ETH Zürich
apmaria@ethz.ch

Aviv Zohar
The Hebrew University
avivz@cs.huji.ac.il

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

Abstract—As the most successful cryptocurrency to date, Bitcoin constitutes a target of choice for attackers. While many attack vectors have already been uncovered, one important vector has been left out though: attacking the currency via the Internet routing infrastructure itself. Indeed, by manipulating routing advertisements (BGP hijacks) or by naturally intercepting traffic, Autonomous Systems (ASes) can intercept and manipulate a large fraction of Bitcoin traffic.

This paper presents the first taxonomy of routing attacks and their impact on Bitcoin, considering both small-scale attacks, targeting individual nodes, and large-scale attacks, targeting the network as a whole. While challenging, we show that two key properties make routing attacks practical: (i) the efficiency of routing manipulation; and (ii) the significant centralization of Bitcoin in terms of mining and routing. Specifically, we find that any network attacker can hijack few (<100) BGP prefixes to isolate ~50% of the mining power—even when considering that mining pools are heavily multi-homed. We also show that on-path network attackers can considerably slow down block propagation by interfering with few key Bitcoin messages.

We demonstrate the feasibility of each attack against the deployed Bitcoin software. We also quantify their effectiveness on a Bitcoin.

One important attack vector has been overlooked though: attacking Bitcoin via the Internet infrastructure using routing attacks. As Bitcoin connections are routed over the Internet—in clear text and without integrity checks—any third-party on the forwarding path can eavesdrop, drop, modify, inject, or delay Bitcoin messages such as blocks or transactions. Detecting such attackers is challenging as it requires inferring the exact forwarding paths taken by the Bitcoin traffic using measurements (e.g., traceroute) or routing data (BGP announcements), both of which can be forged [41]. Even ignoring detectability, mitigating network attacks is also hard as it is essentially a human-driven process consisting of filtering, routing around or disconnecting the attacker. As an illustration, it took Youtube close to 3 hours to locate and resolve rogue BGP announcements targeting its infrastructure in 2008 [6]. More recent examples of routing attacks such as [51] (resp. [52]) took 9 (resp. 2) hours to resolve in November (resp. June) 2015.

One of the reasons why routing attacks have been overlooked in Bitcoin is that they are often considered too challenging to be practical. Indeed, perturbing a vast peer-to-peer
<table>
<thead>
<tr>
<th>Problems</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td><strong>Convergence</strong></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>
With arbitrary policies, BGP may have multiple stable states

preference list
1 prefers to reach 0 via 2 rather than directly
If AS2 is the first to advertise 2 0, the system stabilizes in a state where AS 1 is happy.
If **AS1** is the *first* one to advertise 1 0, the system stabilizes in a state where **AS 2** is happy.
The actual assignment depends on the ordering between the messages

Note that AS1/AS2 could change the outcome by manual intervention

... this is not always possible *

* https://www.nanog.org/meetings/nanog31/presentations/griffin.pdf
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 chose 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

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

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

Memory

\[
\begin{array}{c}
R \\
S
\end{array}
\quad \text{NOR} \quad Q \\
\text{NOR} \quad \bar{Q}
\]
Logic gates

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

Memory

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

Theorem 2: Determining whether an infinite BGP network converges is Turing-complete.

Checking BGP correctness is as hard as checking the termination of a general program.
Abstract—Because of its practical relevance, the Border Gateway Protocol (BGP) has been the target of a huge research effort since more than a decade. In particular, many contributions aimed at characterizing the computational complexity of BGP-related problems. In this paper, we answer computational complexity questions by unveiling a fundamental mapping between BGP configurations and logic circuits. Namely, we describe simple networks containing routers with elementary BGP configurations that simulate logic gates, clocks, and flip-flops, and we show how to interconnect them to simulate arbitrary logic circuits. We then investigate the implications of such a mapping on the feasibility of solving BGP fundamental problems, and prove that, under realistic assumptions, BGP has the same computing power as a Turing Machine. We also investigate the impact of restrictions on the expressiveness of BGP policies and route propagation (e.g., route propagation rules in iBGP and Local Transit Policies in eBGP) and the impact of different message timing models. Finally, we show that the mapping is not limited to BGP and can be applied to generic routing protocols that use several metrics.

We build this mapping assuming a simplified model for BGP routing policies which does not include advanced BGP features like MED or conditional advertisement.

In this paper, we investigate the theoretical consequences of the existence of such a mapping between BGP configurations and logic circuits. We make the following four contributions.

First, we leverage the mapping to characterize the computational complexity of several routing problems in a “bounded” asynchronous model. Contrary to previous works on BGP complexity, in this model each network link is associated with a network delay bounded between finite minimum and maximum values. This effectively imposes a partial order on the exchange of BGP updates. Previous lower bounds for BGP related problems have been proved in models that allow BGP messages to be arbitrarily (even if not indefinitely) delayed [2], [3], [10], [11], [12], [13], [14]. Moreover, the rest of the literature pays little attention to the message timing and to the maximum delay expected.

Second, we leverage the mapping to show that, even when BGP messages are transmitted in a bounded delay network, finding optimal BGP routes can be computationally hard. We focus on the routing problem for which we provide a hardness proof.

Third, we use the mapping to construct a strategy for finding “approximate” BGP routes. Namely, we significantly reduce the complexity of finding a route that is not the best but perfectly acceptable in terms of cost. In particular, we provide a polynomial time algorithm that guarantees an optimal solution within a given multiplicative error.

Fourth, we leverage the mapping to construct a two-layered BGP routing strategy. Namely, we provide a polynomial time algorithm that guarantees a route that is not the best but perfectly acceptable in terms of cost. In particular, we provide a polynomial time algorithm that guarantees a route that is not the best but perfectly acceptable in terms of cost.
In practice though, BGP does not oscillate “that” often

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
Problems

Reachability
Security
Convergence

Performance
Anomalies
Relevance
BGP path selection is mostly economical, not based on accurate performance criteria.
Problems

Reachability

Security

Convergence

Performance

Anomalies

Relevance
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
Google routing blunder sent Japan's Internet dark on Friday

Another big BGP blunder

By Richard Chirgwin 27 Aug 2017 at 22:35

Last Friday, someone in Google fat-thumbed a border gateway protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.

The trouble began when The Chocolate Factory "leaked" a big route table to Verizon, the result of which was traffic from Japanese giants like NTT and KDDI was sent to Google on the expectation it would be treated as transit.

Since Google doesn't provide transit services, as BGP Men explains, that traffic either filled a link beyond its capacity, or hit an access control list, and disappeared.

The outage in Japan only lasted a couple of hours, but was so severe that Japan Times reports the country's Internal Affairs and Communications ministries want carriers to report on what went wrong.

BGP Men dissects what went wrong here, reporting that more than
In August 2017

Someone in Google fat-thumbed a Border Gateway Protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.
In August 2017

Someone in Google fat-thumbed a Border Gateway Protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.

[...] Traffic from Japanese giants like NTT and KDDI was sent to Google on the expectation it would be treated as transit.
In August 2017

Someone in Google fat-thumbed a Border Gateway Protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.

Traffic from Japanese giants like NTT and KDDI was sent to Google on the expectation it would be treated as transit.

The outage in Japan only lasted a couple of hours but was so severe that [...] the country's Internal Affairs and Communications ministries want carriers to report on what went wrong.
Another example, this time from November 2017

Widespread impact caused by Level 3 BGP route leak

Research // Nov 7, 2017 // Doug Madory

For a little more than 90 minutes yesterday, internet service for millions of users in the U.S. and around the world slowed to a crawl. Was this widespread service degradation caused by the...
For a little more than 90 minutes [...],

Internet service for millions of users in the U.S. and around the world slowed to a crawl.

The cause was yet another BGP routing leak, a router misconfiguration directing Internet traffic from its intended path to somewhere else.
“Human factors are responsible for 50% to 80% of network outages”

Ironically, this means that the Internet works better during the week-ends...

source: Job Snijders (NTT)
<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>
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, ...