Q&A Session

Rüdiger Birkner
Tobias Bühler

https://comm-net.ethz.ch/

ETH Zürich
August 6 2018
Old exam from 2016

3 hours instead of 2.5

Topics which we did not discuss this year

Security, SDN, ...
How do you guide IP packets from a source to the destination?
Essentially, there are three ways to compute valid routing state:

<table>
<thead>
<tr>
<th>Intuition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 Use tree-like topologies</td>
<td>Spanning-tree</td>
</tr>
<tr>
<td>#2 Rely on a global network view</td>
<td>Link-State</td>
</tr>
<tr>
<td>#3 Rely on distributed computation</td>
<td>SDN</td>
</tr>
<tr>
<td></td>
<td>SDN</td>
</tr>
<tr>
<td></td>
<td>Distance-Vector</td>
</tr>
<tr>
<td></td>
<td>BGP</td>
</tr>
</tbody>
</table>
Essentially, there are three ways to compute valid routing state:

<table>
<thead>
<tr>
<th>Intuition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 Use tree-like topologies</td>
<td>Spanning-tree</td>
</tr>
<tr>
<td>#2 Rely on a global network view</td>
<td>Link-State</td>
</tr>
<tr>
<td>#3 Rely on distributed computation</td>
<td>Distance-Vector</td>
</tr>
</tbody>
</table>

SDN
BGP
Distance-vector protocols are based on Bellman-Ford algorithm
Every router maintains a table of all received distance vectors

<table>
<thead>
<tr>
<th>dest.</th>
<th>via</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Y</td>
<td>3</td>
</tr>
<tr>
<td>Z</td>
<td>13</td>
</tr>
<tr>
<td>U</td>
<td>30</td>
</tr>
</tbody>
</table>
Let $d_x(y)$ be the cost of the least-cost path known by $x$ to reach $y$.
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 until convergence.

Every router maintains a table of all received distance vectors.

<table>
<thead>
<tr>
<th>dest.</th>
<th>via</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>Z</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>U</td>
<td>30</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dest. cost</th>
<th>X vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>3</td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>U</td>
<td>2</td>
</tr>
</tbody>
</table>
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 until convergence.

Every router maintains a table of all received distance vectors.

Each node updates its distances based on neighbors’ vectors:

$$d_x(y) = \min \{ c(x,v) + d_v(y) \}$$
over all neighbors $v$. 
Whenever a router uses another one, it will announce it an infinite cost.

The technique is known as **poisoned reverse**.
Internet routing comes into two flavors: *intra-* and *inter-domain* routing

Find paths between networks

Find paths within a network
Find paths *between* networks.

- **inter-domain** routing
- **intra-domain** routing
Internet

↓

Border Gateway Protocol (BGP)
BGP is the routing protocol “glueing” the entire Internet together.
BGP announcements carry complete path information instead of distances
Each AS appends itself to the path when it propagates announcements.
There are 2 main business relationships today:

- customer/provider
- peer/peer

*many* less important ones (siblings, backups,...)
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?
Business relationships conditions

route selection

For a destination $p$, prefer routes coming from

- customers over
- peers over $route$ type
- providers
which path to use?

which path to advertise?
control inbound traffic
Routes coming from peers and providers are only propagated to customers

<table>
<thead>
<tr>
<th></th>
<th>customer</th>
<th>peer</th>
<th>provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</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>
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>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
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>
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)
Each BGP router processes UPDATEs according to a precise pipeline
All acceptable routes

BGP Decision Process

Best route to each destination

Forwarding entries

IP packets → IP forwarding table → IP packets
All acceptable routes

BGP Decision Process

Best route to each destination

forwarding entries

IP packets

IP forwarding table

routing-table

BGP sessions

Adj-RIB-In

Input filters

Attribute Manipulation

Neighbor 1

Neighbor 2

Neighbor n

Output filters

Attribute Manipulation

Neighbor 1

Neighbor 2

Neighbor n

IP packets
All acceptable routes

BGP Decision Process

Best route to each destination

IP forwarding table
All acceptable routes

BGP Decision Process

Best route to each destination

forwarding entries

IP forwarding table

Input filters
Attribute Manipulation

Neighbor1

Neighbor2

Neighborn

Output filters
Attribute Manipulation

BGP Decision Process

Loc-Rib

Best route to each destination

forwarding entries

IP forwarding table

Input filters
Attribute Manipulation

Neighbor1

Neighbor2

Neighborn

Output filters
Attribute Manipulation

route-map OUT
Life of a BGP router is made of three consecutive steps

while true:

- receives routes from my neighbors
- select one best route for each prefix
- export the best route to my neighbors
An AS is more than just one router
BGP sessions come in two flavors
external BGP (eBGP) sessions connect border routers in different ASes
internal BGP (iBGP) connect the routers in the same AS
iBGP sessions are used to disseminate externally-learned routes internally
Putting it all together

Route A
31.89.0.0/16
Path: 90 11 12

Route B
31.89.0.0/16
Path: 23 51 12
Putting it all together

Route A

31.89.0.0/16
Path: 90 11 12

Route B

31.89.0.0/16
Path: 23 51 12
Putting it all together

Route A
- 31.89.0.0/16
- Path: 90 11 12

Route B
- 31.89.0.0/16
- Path: 23 51 12
Putting it all together

Route A
31.89.0.0/16
Path: 90 11 12

Route B
31.89.0.0/16
Path: 23 51 12

31.89.0.0/16
Path: 10 90 11 12

31.89.0.0/16
Path: 10 23 51 12
Assignment 1 - Internet Communication

- Packet
- Frame
- Sending node
- Adapter
- Link layer protocol
- Receiving node
Assignment 1 - Internet Communication

Link Communication medium and Network adapter

Wifi
Ethernet
Fiber
Accessing a website: DNS & HTTP

step 1  Open browser and enter the URL  www.google.com

step 2  Browser invokes DNS to resolve the URL into an IP  216.58.215.238

step 3  Browser creates a HTTP request to retrieve the website  GET / HTTP/1.1
          Host: www.google.com
Accessing a website: DNS & HTTP

What if we do the DNS resolution ourselves?

step 1
Perform a DNS lookup for the given URL

dig www.google.com

step 2
Open browser and enter the IP address

216.58.215.238

step 3
Browser creates a HTTP request to retrieve the website

GET / HTTP/1.1
Host: 216.58.215.238
In practice, multiple URLs can be mapped to the same IP

<table>
<thead>
<tr>
<th>name</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.vanbever.eu">www.vanbever.eu</a></td>
<td>188.165.240.60</td>
</tr>
<tr>
<td><a href="http://www.routeur.be">www.routeur.be</a></td>
<td>188.165.240.60</td>
</tr>
</tbody>
</table>

How does a web server receiving an HTTP request know, which website you want to access?
The host field tells the server which website it should serve

HTTP request: GET / HTTP/1.1
Host: www.google.com
Anycast, Unicast, Multicast

„one-to-one-of-many“

Important, discussed in lecture

Used for scalability, load-balancing (e.g. DNS root server)

Routing finds shortest-paths

Seamless replication

But, potential problems for stateful applications
Anycast, Unicast, Multicast
Anycast, **Unicast**, Multicast

„one-to-one“

Destination address uniquely identifies a single receiver

No replication
Anycast, **Unicast**, Multicast

30.0.0.0/24
Anycast, Unicast, Multicast

„one-to-many-of-many“ („many-to-many-of-many“)

Not important for exam

E.g. useful to stream the same video to multiple receivers
Anycast, Unicast, **Multicast**
Assignment 5 - Detective Work

Poor choice of IP subnets from our side

Indeed, 192.168.0.0/16 is a private subnet space
normally not routed in the Internet
## Assignment 5 - Detective Work

<table>
<thead>
<tr>
<th>SRC MAC Address</th>
<th>DST MAC Address</th>
<th>SRC IP Address</th>
<th>DST IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>11:05:ab:59:bb:02</td>
<td>192.168.11.1</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.16.1</td>
</tr>
<tr>
<td>da:15:00:00:01:11</td>
<td>11:05:ab:59:bb:02</td>
<td>129.132.103.40</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>40:34:00:7a:00:01</td>
<td>192.168.8.2</td>
<td>192.168.15.254</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>ac:00:0a:aa:10:05</td>
<td>192.168.8.2</td>
<td>192.168.9.99</td>
</tr>
<tr>
<td>ac:00:0a:aa:10:05</td>
<td>01:05:3c:34:00:02</td>
<td>192.168.9.99</td>
<td>192.168.13.255</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.8.1</td>
</tr>
</tbody>
</table>
Assignment 5 - Detective Work

<table>
<thead>
<tr>
<th>SRC MAC Address</th>
<th>DST MAC Address</th>
<th>SRC IP Address</th>
<th>DST IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>11:05:ab:59:bb:02</td>
<td>192.168.11.1</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.16.1</td>
</tr>
<tr>
<td>da:15:00:00:01:11</td>
<td>11:05:ab:59:bb:02</td>
<td>129.132.103.40</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>40:34:00:7a:00:01</td>
<td>192.168.8.2</td>
<td>192.168.15.254</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>ac:00:0a:aa:10:05</td>
<td>192.168.8.2</td>
<td>192.168.9.99</td>
</tr>
<tr>
<td>ac:00:0a:aa:10:05</td>
<td>01:05:3c:34:00:02</td>
<td>192.168.9.99</td>
<td>192.168.13.255</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.8.1</td>
</tr>
</tbody>
</table>

☐ Router interface MAC address
## Assignment 5 - Detective Work

<table>
<thead>
<tr>
<th>SRC MAC Address</th>
<th>DST MAC Address</th>
<th>SRC IP Address</th>
<th>DST IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>11:05:ab:59:bb:02</td>
<td>192.168.11.1</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.16.1</td>
</tr>
<tr>
<td>da:15:00:00:01:11</td>
<td>11:05:ab:59:bb:02</td>
<td>129.132.103.40</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>40:34:00:7a:00:01</td>
<td>192.168.8.2</td>
<td>192.168.15.254</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>ac:00:0a:aa:10:05</td>
<td>192.168.8.2</td>
<td>192.168.9.99</td>
</tr>
<tr>
<td>ac:00:0a:aa:10:05</td>
<td>01:05:3c:34:00:02</td>
<td>192.168.9.99</td>
<td>192.168.13.255</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.8.1</td>
</tr>
</tbody>
</table>

- Router interface MAC address
- Dst 192.168.8.2 does not go over router => internal
## Assignment 5 - Detective Work

<table>
<thead>
<tr>
<th>SRC MAC Address</th>
<th>DST MAC Address</th>
<th>SRC IP Address</th>
<th>DST IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>11:05:ab:59:bb:02</td>
<td>192.168.11.1</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.16.1</td>
</tr>
<tr>
<td>da:15:00:00:01:11</td>
<td>11:05:ab:59:bb:02</td>
<td>129.132.103.40</td>
<td>192.168.8.2</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>40:34:00:7a:00:01</td>
<td>192.168.8.2</td>
<td>192.168.15.254</td>
</tr>
<tr>
<td>11:05:ab:59:bb:02</td>
<td>ac:00:0a:aa:10:05</td>
<td>192.168.8.2</td>
<td>192.168.9.99</td>
</tr>
<tr>
<td>ac:00:0a:aa:10:05</td>
<td>01:05:3c:34:00:02</td>
<td>192.168.9.99</td>
<td>192.168.13.255</td>
</tr>
<tr>
<td>6a:00:02:49:a1:a0</td>
<td>da:15:00:00:01:11</td>
<td>192.168.11.1</td>
<td>192.168.8.1</td>
</tr>
</tbody>
</table>

- **Router interface MAC address**
- **Dst 192.168.8.2 does not go over router => internal**
- **Dst 192.168.8.1 reaches router and has to be in the same subnet as 192.168.8.2**

=> **192.168.8.1 is the IP of the router**
DHCP (assignment 10 and book chapter 4)

Assignment uses 2 packets, book 4

Assignment simplifies the process slightly
assumes only one DHCP server

The sender recognizes its response based on the transaction ID
therefore not a problem that also the response is broadcasted
DHCP (assignment 10 and book chapter 4)
VLAN

Access link: part of only one VLAN
normally connects hosts with switches (to get „access“)

Trunk link: can carry traffic for multiple VLANs
normally connects switches to other switches or routers

The per-VLAN spanning tree still spans the entire network
even if some of the switches do not have hosts in all VLANs
=> better optimized paths for hosts in one VLAN
=> ready for new hosts in the future
VLAN (spanning tree from the slides)
TCP Congestion Window

**Additive Increase** pseudo code from the slides:

\[
\text{CWND} = \text{CWND} + \frac{1}{\text{CWND}}
\]
TCP Congestion Window

**Additive Increase** pseudo code from the slides:

\[
\text{CWND} = \text{CWND} + \frac{1}{\text{CWND}}
\]

More precise computation:

\[
\text{CWND}(t+1) = \begin{cases} 
\text{CWND}(t) + a & \text{if no congestion detected} \\
\text{CWND}(t) \times b & \text{if congestion detected}
\end{cases}
\]

With \( a = \text{MSS} \) and \( b = \frac{1}{2} \)

\( t = \text{current RTT} \)
TCP Congestion Window (assignment 8)

Question c): how much time elapsed between E and F?
TCP Congestion Window (assignment 8)

Question c): how much time elapsed between E and F?

=> depends on when F is exactly happening
TCP Congestion Window - diagrams

The presented diagrams do not capture all the details, e.g.

We will make sure that future question precisely define what the marked points represent.
Individual Questions