A Hierarchical Model for BGP Routing Policies

Laurent Vanbever, Bruno Quoitin and Olivier Bonaventure
UCL, Belgium

PRESTO’09
Friday, 21 Aug 2009
Human factors are responsible for 50 to 80 percent of network device outages

A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

Conclusion
A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

Conclusion
A BGP Router at a Glance

BGP Adj-RIB-In → BGP Loc-Rib → BGP Adj-RIB-Out

Neighbor1 → Input filters → Attribute Manipulation → BGP Decision Process → Best route to each destination → Output filters → Neighbor1

Neighbor2 → Input filters → Attribute Manipulation → BGP Decision Process → Best route to each destination → Output filters → Neighbor2

Neighborn → Input filters → Attribute Manipulation → BGP Decision Process → Best route to each destination → Output filters → Neighborn
Talk is about BGP Policies

BGP sessions → BGP Adj-RIB-In → BGP Loc-Rib → BGP Adj-RIB-Out → BGP sessions

Neighbor₁ → Input filters → Attribute Manipulation → Neighbor₁

Neighbor₂ → Input filters → Attribute Manipulation → Neighbor₂

Neighborₙ → Input filters → Attribute Manipulation → Neighborₙ

All acceptable routes → BGP Decision Process → Best route to each destination
BGP Policies give operators control over routes selection

Policies are mainly used to

- **filter incoming routes**
  - *ignore* routes you don’t want to consider
- **modify routes’ attributes**
  - *influence* path selection
  - *modify* the way routes are perceived
- **filter outgoing routes**
  - *enforce* business relationships
BGP Policies are defined at different *abstraction* levels
Some BGP Policies are defined on *all* sessions
Some BGP Policies are defined on *groups* of sessions.

- A
- B
- C
- D
- E

- *provider*
- *customer*
Some BGP Policies are defined on AS sessions.
Some BGP Policies are defined on *individual* sessions.
Some BGP Policies are defined on *prefixes*
However, policies are often defined at low level

neighbor 206.196.178.45 {
    description "Mid-Atlantic Crossroads (MAX)";
    import [SANITY-IN SET-CONNECTOR-PREF MAX-IN CONNECTOR-IN ];
    peer-as 10886;
}
neighbor 192.88.192.137 {
    description OSCnet;
    import [SANITY-IN SET-CONNECTOR-PREF OARNET-IN CONNECTOR-IN ];
    peer-as 3112;
}
neighbor 204.238.76.5 {
    description "Drexel University";
    import [SANITY-IN SET-CONNECTOR-PREF DREXEL-IN CONNECTOR-IN ];
    peer-as 36412;
}
neighbor 192.88.115.24 {
    description 3ROX;
    import [SANITY-IN SET-CONNECTOR-PREF PSC-IN CONNECTOR-IN ];
    peer-as 5050;
}
...
neighbor 199.18.156.241 {
    description "OSCnet mcast-only for their non-I2 customers";
    import [SANITY-IN SET-CONNECTOR-PREF CONNECTOR-IN ];
    peer-as 600;
}
A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

Conclusion
Towards a *hierarchical* model of routing policies

Our model aims to

- express a policy at the appropriate level
- represent *network-wide* policies
- ease policy addition and modification
- be vendor agnostic
Our model at a glance

BGP Session’s abstraction

Chains of routing filters

BGP Session’s abstraction associated to templates
Our model is structured around *chains* of filters

Policies are modeled by chains

- A node is a sequence of *rules*.
- A *rule* is a couple \((\text{predicate}, \text{template})\).
- A *predicate* conditions the association of the template to the session’s filters.
- A *template* is a sequence of routing filters statements.
Our model is structured around *chains* of filters

\[ \text{true}, (r.pfx \in \text{BOGONS}) \Rightarrow \text{reject} \]

**Predicate**  
**Template**  
**Rule**

\[ S : \text{session} \]
\[ r : \text{route} \]
Our model is structured around *chains* of filters

\[
\begin{align*}
  s.group &= \text{CUST}, \ r\text{.comm} \uplus \{\text{CUST}\} \\
  s.group &= \text{PEER}, \ r\text{.comm} \uplus \{\text{PEER}\} \\
  s.group &= \text{PROV}, \ r\text{.comm} \uplus \{\text{PROV}\}
\end{align*}
\]

\( s \): session  
\( r \): route
A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

Conclusion
How is it implemented?

- Each BGP session is specified with a *textual representation*
  
  BXL:CUST:2611:<130.104.0.2>:backup

- Predicates are modeled by *regular expressions*
  
  s.type=backup modeled *.backup$

- Templates are represented by using *StringTemplate*
  
  ```
  policy-statement BACKUP-PREF {
    term down-pref {
      then {
        local-preference subtract $value$;
        accept;
      }
    }
    ...
  }
  ```
How does it work?

BXL:CUST:2611:<130.104.0.2>:backup
How does it work?

BXL: CUST: 2611:<130.104.0.2>: backup

\[(r.pfx \in BOGONS) \Rightarrow reject\]
How does it work?

BXL: **CUST**: 2611: <130.104.0.2> : backup

(r.pfx ∈ BOGONS) ⇒ reject
r.lp = 5000
How does it work?

BXL:CUST:2611:<130.104.0.2>:backup

(r.pfx ∈ BOGONS) ⇒ reject
r.lp = 5000
r.lp = r.lp − 500
How does it work?

BXL: **CUST**:2611:<130.104.0.2>:backup

(r.pfx ∈ BOGONS) ⇒ reject
r.lp = 5000
r.lp = r.lp − 500
r.comm ⊔ {CUST}
How does it work?

BXL: \textbf{CUST}:2611:<130.104.0.2>:backup

\[(r.pfx \in BOGONS) \Rightarrow reject\]
\[r.lp = 5000\]
\[r.lp = r.lp - 500\]
\[r.comm \uplus \{CUST\}\]
\[(r.pfx \notin RIR\_PFX(s.asn)) \Rightarrow reject\]
How does it work?

BXL: **CUST**:2611:<130.104.0.2>:backup

- $(r.pfx \in \text{BOGONS}) \Rightarrow \text{reject}$
- $r.lp = 5000$
- $r.lp = r.lp - 500$
- $r.comm \uplus \{\text{CUST}\}$
- $(r.pfx \notin \text{RIR-PFX}(s.asn)) \Rightarrow \text{reject}$

*announce default route*
How does it work?

BXL:PROVIDER:2611:<130.104.0.2>
How does it work?

\[(r.pfx \in BOGONS) \Rightarrow \text{reject}\]
How does it work?

BXL: **PROVIDER**: 2611: <130.104.0.2>

\[(r.pfx \in BOGONS) \Rightarrow reject\]
\[r.lp = 3000\]
How does it work?

(r.pfx ∈ BOGONS) ⇒ reject
r.lp = 3000
r.comm ⊢ {PROV}
How does it work?

BXL: **PROVIDER**:2611:<130.104.0.2>

\[
(r.pfx \in BOGONS) \Rightarrow reject
\]
\[
r.lp = 3000
\]
\[
r.comm \uplus \{PROV\}
\]

\[
((r.comm \ni CUST) \lor (r.pfx \in INTERNAL)) \Rightarrow accept
\]
A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation and Evaluation

Conclusion
To Conclude

Our model offers

- a *network-wide* and *vendor-agnostic* way of configuring routing policies
- detailed documentation
- quick and safe modifications/additions
A Hierarchical Model for BGP Routing Policies

Laurent Vanbever, Bruno Quoitin and Olivier Bonaventure

UCL, Belgium

Questions?