CS434/534: Topics in Network Systems

Topic 3: Scalable, Programmable (Software-Defined) Networking: BGP Analysis; 5GC Networking, Network LB

Nov. 9, 2021

Y. Richard Yang
Computer Science Department, Yale University
Email: yry@cs.yale.edu

http://zoo.cs.yale.edu/classes/cs434/
Admin

- Project 1 to be return by this week

- Final Project planning
  - Nov. 11: Initial teaming/potential topics
  - Nov. 18: Iteration of topics
  - Dec. 3: Checkpoint
  - Dec. 16: Due
Recap: Scaling Distributed Link State Networking

- **OSPF scaling**
  - Introduce in the graph a network node representing each multi-access network
  - Each network elects a designed router (DR) and a backup designed router (BDR)
    - Differentiate between neighboring relation (who can hear each other) and adjacency relationship (who synchronizes with each other)
  - Backbone-area hierarchical model: summary based abstraction

- **OSLRv2 scaling**
  - Flooding reduction: forwarding only by Flooding MPR
  - Topology reduction (reducing the number of links distributed): routers declare link state information for their routing MPR selectors, if any.
  - Equation-driven distributed protocol design
    \[
    N_1(n) = \{ x \in N_{1\_rcv}(n) : n \in N_{1\_rcv}(x) \} \\
    N_2(n) = \text{union}\{N_1(x): x \in N_1(n)\} \setminus N_1(n)
    \]
Recap: (OSPF) Link State Protocol Analysis

- Consider a single OSPF area (network)
  - After a link change, propagation is whole area
  - The nodes in the same area must use exactly the same algorithm when computing the paths

- Backbone+area hierarchy correctness and flexibility
  - The protocol ensures convergence, despite processing loops (area -> backbone -> area)
  - But, if the areas form a general topology, there can be routing loops
Recap: BGP Autonomous System, Path Vector Networking

- **Main goal:** flexibility/autonomy
  - allow flexible abstraction of networks: the abstracted networks can form **flexible inter-connection topology** (vs simple spoke-leaf OSPF)
  - allow individual networks autonomous control of both **internal routing and external routing** (e.g., the upstream who can use the network, and how the network chooses downstream)

- **BGP solution:**
  - flexible abstraction of networks into autonomous systems (AS)
  - Path vector based policy routing
  - BGP is the de facto networking routing protocol of the global Internet
Recap: BGP Data Structures and Workflow

Routing cache

Adj-RIB-In

1 3 0
1 0

select best path

Loc-RIB

export path to neighbors

Adj-RIB-Out

route selection policy: rank paths

export policy: which paths export to which neighbors

Yale

AT&T

Qwest

Internet2

1 3 0
1 0
BGP Example

- Ingress control: Control exports to determine (upstream) ingress
- Egress control: selects the preferred downstream
Demo: BGP Looking Glass for the Global Internet

- BGP routing:
  - BGP looking glass server (http://www.bgp4.as/looking-glasses),
    - e.g., CERN (http://lg.cern.ch/)

- Using one of the looking glass servers:
  http://www.bgp4.as/looking-glasses
Routing Table Size of BGP in Global Internet (number of globally advertised, aggregated entries)

Src: https://bgp.potaroo.net/as1221/bgp-active.html
Example: Data Center Networking (Design 1)

- OSPF based
  - OSPF to collect network state
  - Equal cost multipath (ECMP)
    - Compute shortest paths to each dest.
    - If two neighbors have the same cost to the dest, hash on flow address and equal split

Exercise: How many EC paths src -> dst1?
Exercise: How many EC paths src -> dst2?
Example: Data Center Networking Plane (Design 2; BGP)
Data Center Network Routing: BGP Benefits

- Simple protocol
- Simpler protocol design concepts
  - Less finite state machines, data structures
- Troubleshooting BGP is simpler
  - BGP RIB structure is simpler compared to link-state LSDB
  - Clear picture of what sent where (RIBIn, RIBOut)
- Event propagation is more constrained in BGP
  - E.g. link failures have limited propagation scope
  - More stability due to reduced event “flooding” domains
- BGP allows for per-hop traffic engineering
  - Use for unequal-cost Anycast load balancing solution
Outline

- Admin and recap
- Scalable, programmable (software-defined) networking
  - Overview and roadmap
  - Scaling networking from a switch to the whole Internet
    - Flooding + learning networking protocol
    - Scalable, distributed link state networking protocol (OSPF)
    - Scaling link state protocol in MANET (OLSRv2)
    - Autonomous system, path vector networking protocol (BGP)
      - BGP model, protocol and processing
      - BGP convergence analysis
BGP Convergence

- A decentralized policy routing system as BGP can be considered as a system to aggregate individual preferences, but aggregation may not be always successful.
  
  The BAD GADGET example:
  - 0 is the destination
  - the route selection policy of each AS is to prefer its counter clock-wise neighbor

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>210</td>
<td>20</td>
<td>320</td>
</tr>
<tr>
<td>3</td>
<td>130</td>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>
```

- preferred
- less preferred
BGP Policy Analysis as Dependency Analysis

- **Observation:** BGP decisions have dependencies
  - The “closer” a node to the destination, the more “powerful” it may be

- **Note:** this captures egress routing (only on paths starting from itself) only
  - BGP handling ingress is not well understood and a good project
Complete Dependency: P-Graph

- Complete dependency can be captured by a structure called P-graph.
- Nodes in P-graph are feasible paths.
- Edges represent dependency priority (low to high):
  - A directed edge from path $N_1 P_1$ to $P_1$
    - Intuition: to let $N_1$ choose $N_1 P_1$, $P_1$ must be chosen and exported to $N_1$
  - A directed edge from a lower ranked path to a higher ranked path
    - Intuition: the higher ranked path should be considered first

Any observation on the P-graph?
P-Graph and BGP Convergence

- If the P-graph of the networks has no loop, then policy routing converges.
- Example: suppose we swap the order of 30 and 320

Intuition (general case): BGP converges despite asynchronous BGP protocol, if P-graph has no loop.

Exercise: What are the final paths?
Outline

- Admin and recap
- Scalable, programmable (software-defined) networking
  - Overview and roadmap
  - Scaling networking from a switch to the whole Internet
    - Flooding + learning networking protocol
    - Scalable, distributed link state networking protocol (OSPF)
    - Scaling link state protocol in MANET (OLSRv2)
    - Autonomous system, path vector networking protocol (BGP)
      - BGP model, protocol and processing
      - BGP convergence analysis
        - Dependency using P-graph
        - Dependency from economics
Internet Economy (The Invisible Hand Controlling Internet): Two Types of Business Relationship

**Customer provider relationship**
- a provider is an AS that connects the customer to the rest of the Internet
- customer pays the provider for the transit service
- e.g., Yale is a customer of AT&T and QWEST

**Peer-to-peer relationship**
- mutually agree to exchange traffic between their respective customers only
- there is no payment between peers
Route Selection Policies Driven by Economics

- Route selection (ranking) policy:
  - the typical route selection policy is to prefer customers over peers/providers to reach a destination, i.e., Customer > Peer/Provider (why?)
Export Policies Driven by Economics

**case 1:** routes learned from customer

Routes learned from a customer are sent to all other neighbors.

**case 2:** routes learned from provider

Routes learned from a provider are sent only to customers.

**case 3:** routes learned from peer

Routes learned from a peer are sent only to customers.
Example: Typical Export -> No-Valley Routing

Suppose $P_1$ and $P_2$ are providers of $A$; $A$ is a provider of $C$
**Typical Export Policies Route Patterns**

- Assume a BGP path SABCD to destination AS D. Consider the business relationship between each pair:

![Diagram of route relationships]

- Three types of business relationships:
  - PC (provider-customer)
  - CP (customer-provider)
  - PP (peer-peer)
Invariant 1 of valid BGP routes (with labels representing business relationship)

Reasoning: only route learned from customer is sent to provider; thus after a PC, it is always PC to the destination
Invariant 2 of valid BGP routes (with labels representing business relationship)

Reasoning: routes learned from peer or provider are sent to only customers; thus all relationship before is CP.
Stability of BGP Policy Routing

- Suppose
  1. there is no loop formed by provider-customer relationship in the Internet
  2. each AS uses typical route selection policy: $C > E/P$
  3. each AS uses the typical export policies

- Then decentralized BGP policy routing converges (i.e., is stable).
Case 1: A Link is PC

Proof by contradiction. Assume a loop in P-graph. Consider a fixed link in the loop.
Case 2: Link is CP/PP
Outline

- Admin and recap
- Scalable, programmable (software-defined) networking
  - Overview and roadmap
  - Scaling networking from a switch to the whole Internet
    - Flooding + learning networking protocol
    - Scalable, distributed link state networking protocol (OSPF)
    - Scaling link state protocol in MANET (OLSRv2)
    - Autonomous system, path vector networking protocol (BGP)
      - BGP model, protocol and processing
      - BGP convergence analysis
      - Decentralized networking analysis
Theory framework

- Given individual preferences, define a framework (called constitution in social choice; protocol in network systems) to aggregate individual preferences:
  - A set of choices: a, b, c, ...
  - A set of voters 1, 2, ...
    - Each voter has a preference (ranking) of all choices, e.g.,
      » voter 1: a > b > c
      » voter 2: a > c > b
      » voter 3: a > c > b
  - A well-specified aggregation rule (protocol) computes an aggregation of ranking, e.g.,
    - Society (network): a > c > b
Example: Aggregation of Global Preference

- **Choices (for S→D route):**
  - SAD, SBD, SABD, SBAD

- **Voters:**
  - S, A, B, D

- **Each voter has a preference, e.g.,**
  - S: SAD > SBD > SABD > SBAD
  - A: SAD > SABD > SBD > SBAD
  - ...
Global Aggregation Framework/Protocol

- Axioms:
  - Transitivity
    - if $a > b$ & $b > c$, then $a > c$
  - Unanimity:
    - If all participants prefer $a$ over $b$ ($a > b$) => $a > b$
  - Independence of irrelevant alternatives (IIA)
    - Global ranking of $a$ and $b$ depends only on the relative ranking of $a$ and $b$ among all participants

- Result:
  - Arrow’s Theorem: Any constitution (protocol) that respects transitivity, unanimity and IIA must be a dictatorship.
Proofs of Arrow’s Theorem

- There are quite a few proofs, and the six-page paper linked on the Schedule page gives three simple proofs.

- Below, we give the key insight of the proof using approach 1.
The Extremal Lemma

- Let choice \( b \) be chosen arbitrarily. Assume that every voter puts \( b \) at the very top or the very bottom of his ranking. Then constitution protocol must as well (even if half voters put \( b \) at the top and half at the bottom)

- Proof: by contradiction.
  - Assume there exist \( a \) and \( c \) such that constitution protocol has \( a \geq b \); \( b \geq c \).
  - We can move \( c \) above \( a \) w/o changing \( a-b \) or \( b-c \) votes
Step 1: Existence of Pivotal Voter

- Let choice b be chosen arbitrarily. There exists a voter n* = n(b) who is extremely pivotal for b in the sense that by changing his vote at some profile, he can move b from the very bottom to the very top in the global ranking.

Proof:
- Consider an extreme profile where b is at the bottom of each voter.
- Consider voter from 1 to n, and we move b from bottom to top one-by-one.
- The first voter whose change causes b to move to the top is n*(b)
Step 2: $n^* = n(b)$ is dictator of any pair $ac$ not involving $b$

Proof

- Consider a from $ac$ pair. We show that if $a > n^* c$, then constitution protocol has $a > c$
- Let profile before $n^*$ moves $b$ to top as profile I
- Let profile after $n^*$ moves $b$ to top as profile II
- Construct profile III from II by letting $n^*$ move $a$ above $b$; all others can arrange $ac$ as they want, but leave $b$ in extreme position

<table>
<thead>
<tr>
<th>Profile</th>
<th>1</th>
<th>2</th>
<th>n*</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile I</td>
<td>b</td>
<td>b</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>b</td>
<td>b</td>
</tr>
</tbody>
</table>

Constitution protocol: b bottom

<table>
<thead>
<tr>
<th>Profile</th>
<th>1</th>
<th>2</th>
<th>n*</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile II</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>b</td>
</tr>
</tbody>
</table>

Constitution protocol: b top

<table>
<thead>
<tr>
<th>Profile</th>
<th>1</th>
<th>2</th>
<th>n*</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile III</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>b</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>c</td>
<td>b</td>
</tr>
</tbody>
</table>

Constitution protocol:
- $a > b$ since ab same as I
- $b > c$ since bc same as II
Step 3: $n^*$ is dictator for every pair $ab$

- Consider $c$ not equal to $a$ or $b$
- There exists $n(c)$ who is a dictator of any pair not involving $c$, such as the pair $ab$, i.e.,
  - For any profile, if $a >_{n(c)} b$, $a > b$ in constitution protocol
- $n(c)$ must be $n^*$
  - Assume not.
    - Consider Profile I and Profile II.
    - Since $n(c)$ is not $n^*$, $n(c)$ ranking of $ab$ does not change in Profile I and Profile II.
    - When $n^*$ changes $ab$ ranking between Profile I and Profile II, the global ranking of $ab$ changes.
    - Contradiction.
Summary: BGP Routing Architecture Design

- **Scalability:**
  - introduces the abstraction of AS to hide network details

- **Privacy:**
  - route carries only path vector, not internal network path

- **Autonomy**
  - Autonomous systems have flexibility to choose their own internal and (egress, and influence ingress) external routing
  - Policy dispute is a major potential issue in a decentralized setting but the Internet economy drives to stability so far
  - Other use of policy should consider stability
Outline

- Admin and recap
- Scalable, programmable (software-defined) networking
  - Overview and roadmap
  - Scaling networking from a switch to the whole Internet
    - Flooding + learning networking protocol
    - Scalable, distributed link state networking protocol (OSPF)
    - Scaling link state protocol in MANET (OLSRv2)
    - Autonomous system, path vector networking protocol (BGP)
    - Scaling from fixed networks to cellular networks (5G core networking)
Basic 5G Core Structure

Discussion: What key challenges do you see in 5G cellular routing (networking)?

Focus on:
- Roaming (UE location can change)
- QoS (beyond best effort data traffic)
- Scaling

UE: User Equipment  gNB: next-generation NodeB
UPF: User Plane Function  DN: Data Network, e.g., Internet
Outline

- Admin and recap
- Scalable, programmable (software-defined) networking
  - Overview and roadmap
  - Scaling networking from a switch to the whole Internet
    - Flooding + learning networking protocol
    - Scalable, distributed link state networking protocol (OSPF)
    - Scaling link state protocol in MANET (OLSRv2)
    - Autonomous system, path vector networking protocol (BGP)
    - Scaling from fixed networks to cellular networks (5G core networking)
      - User (data) plane (path)
Using of routing update to handle roaming (mobility) is not scalable in large-scale networks (why?)

Exercise: do you have any proposal?

5G basic idea:
- Shield UE mobility from all except ingress/egress nodes
- => routing using tunneling from ingress to egress
**5GC Data Path Basic Idea: Downlink**

Q: What does a UPF need to know to forward to UE?

Access Node (AN) Tunnel: tunnel at gNB, for UPF to forward downlink traffic to UE.

Identification: gNB IP+UE Tunnel Endpoint ID TEID_an
Q: What does a gNB need to know to forward to the right UPF?

Core Node (CN) Tunnel
(for gNB to forward uplink traffic to the right UPF):

Identification: UPF IP+UE Tunnel Endpoint ID TEID_cn
Exercise: What may a packet from gNB->UPF for UE to Internet (DN) look like?