Theoretical Frameworks for Routing Problems in the Internet

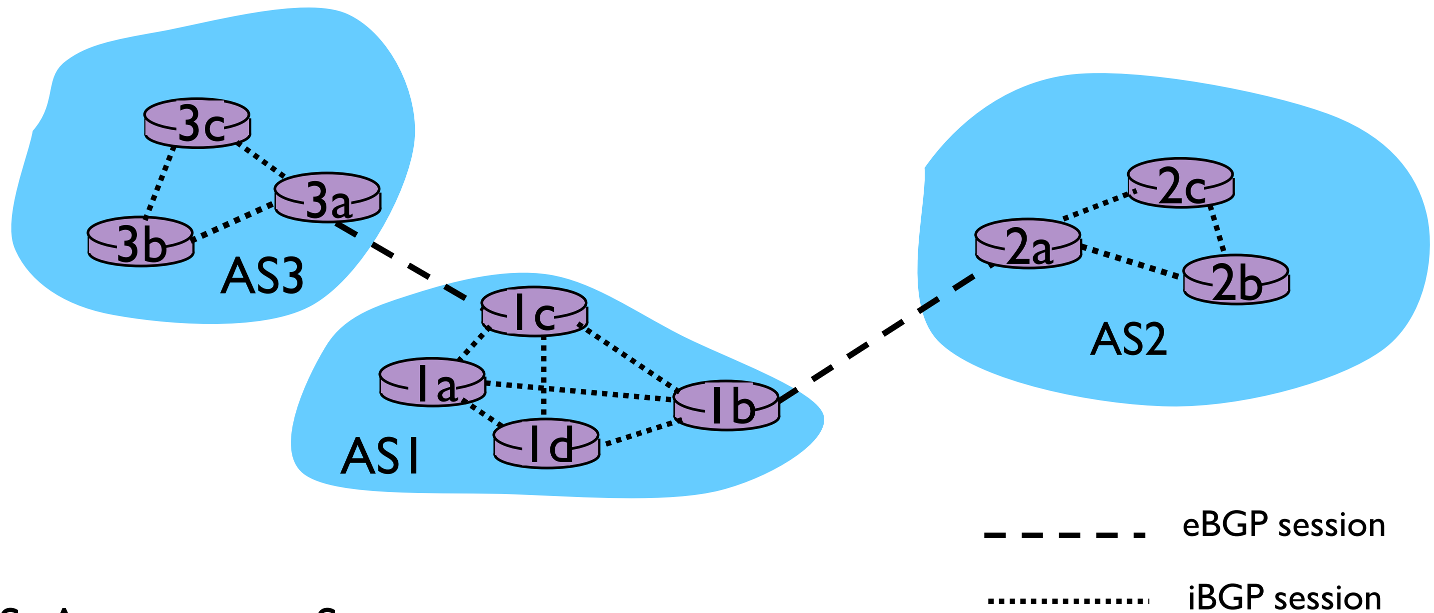
Menglin Liu

Outline

- Background
  - BGP
- Theoretical frameworks
  - Path algebra
  - Routing algebra
  - Stable path problem
  - Policy structure and routing structure
- Metarouting
  - Routing Algebra Meta-Language (RAML)

BGP (Border Gateway Protocol)

- A Network Layer Protocol
- Designed as the core routing protocol of the Internet

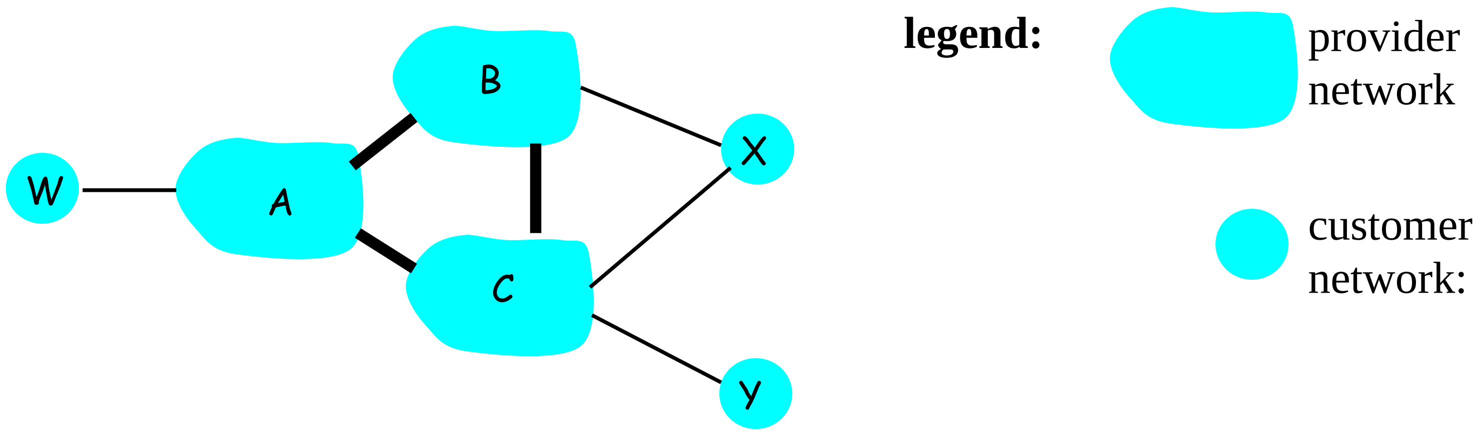


AS: Autonomous System

- From Prof Mukherjee's [ECS 152 lecture notes]

BGP Routing Policy

- An example:



  - X does not want to route from B via X to C
  - .. so X will not advertise to B a route to C

From Prof Mukherjee's ECS 152 lecture notes

BGP Divergence

- BGP is not a pure distance-vector since the routing policies can override distance metrics

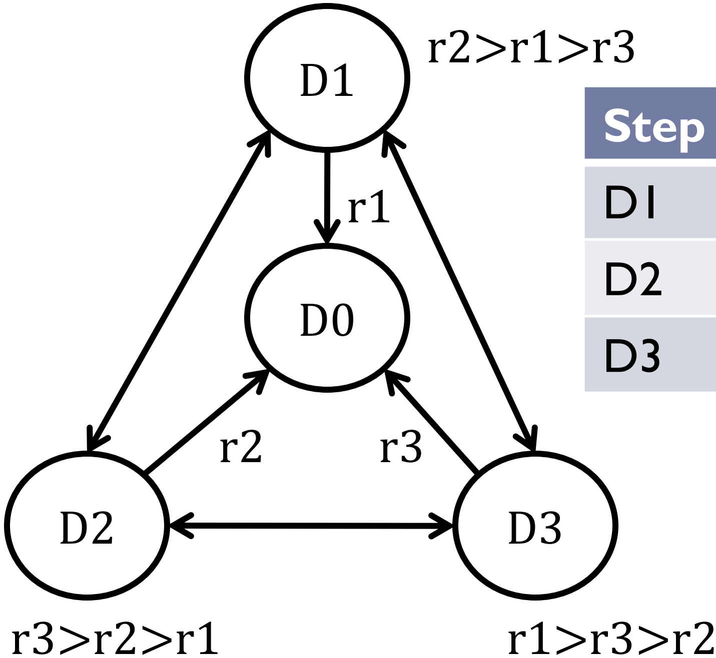
Distance vector:

- From time-to-time, each node sends its own distance vector estimate to neighbors
- When a node x receives new DV estimate from neighbor, it updates its own DV using Bellman-Ford equation:

$$D_x(y) \leftarrow \min_y \{c(x, y) + D_y(y)\} \quad \textit{for each node } y \in \boldsymbol{N}$$

- The routing policies may conflict and cause BGP to diverge.

BGP Divergence: An Example



Step 0		Step 1		Step 2	
D1	r1	D1	r1	D1	r1
D2	r3	D2	r3	D2	r2
D3	r3	D3	r1	D3	r1

Step 3		Step 4	
D1	r2	D1	r2
D2	r2	D2	r2
D3	r1	D3	r3

Step 5		Step 6	
D1	r2	D1	r1
D2	r3	D2	r3
D3	r3	D3	r3

Persistent route oscillations!

Example from [VGE00]

Theoretical Framworks

- How to model the BGP routing problems?

  Policy-based Routing

  - Path Algebras
  - Routing Algebras
  - Stable Path Problem
  - Policy Structure and Routing Structure

- And how to ensure the existence of a solution (stable routing)?
  - Universal condition
  - Instance condition

Path Algebras (Semi-rings)

$(X, \oplus, \otimes, \bar{0}, \bar{1})$

- X : Values that will be associated with routes and edges

$\oplus$

- Commutative: $a \oplus b = b \oplus a$
- Associative: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- Identity: $a \oplus \bar{0} = a$
- Selectivity: $a \oplus b = a \ or \ b$

$\otimes$

- Associativity: $(a \otimes b) \otimes c = a \otimes (b \otimes c)$
- Identity: $\bar{1} \otimes a = a$
- Annihilator: $a \otimes \bar{0} = \bar{0}$

Path Algebra (cont.)

$(X, \oplus, \otimes, \bar{0}, \bar{1})$

$\oplus$ and $\otimes$

- Distributivity: $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

  $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$

$\oplus$	**Path Selection Function**
$\otimes$	Path Computing Function

Path Algebra Examples

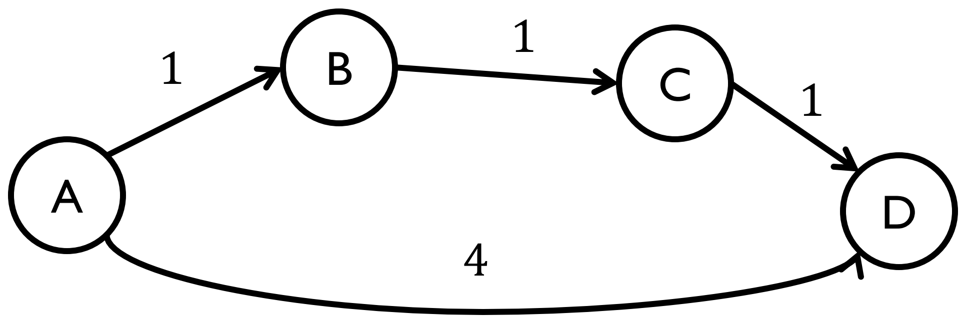
$\mathfrak{B}$	$\mathcal{X}$	$\oplus$,	$\otimes$	$\bar{0}$	$\bar{1}$	description
$\mathfrak{E}$	$\{0, 1\}$	max	min	0	1	usable-path routing (the Boolean semi-ring)
$\mathfrak{M}$	$\mathbb{Z} \cup \{\infty\}$	min	$+$	∞	0	minimum-weight routing
$\mathfrak{M}^+$	$\mathbb{Z}^+ \cup \{\infty\}$	min	$+$	∞	0	minimum-weight routing, non-negative weights
$\mathfrak{R}$	$[0, 1]$	max	$\times$	0	1	most-reliable routing
$\mathfrak{C}$	$\{0, 1, 2, \ldots, k\} \cup \{\infty\}$	max	min	0	∞	greatest-capacity routing

- $P1 : A \rightarrow B \rightarrow C \rightarrow D$
- $P2 : B \rightarrow C \rightarrow D$
- $P3 : A \rightarrow D$

$$P2 = BC \otimes CD$$

$$P1 = AB \otimes P2$$

$$P1 \oplus P3 = P1$$



Path Algebra -- Conditions

- Universal Conditions
  - Super-unitary

    $\bar{1}$: zero weight

    $\bar{1} \oplus x = \bar{1}$: zero weight is best possible – no negative edge
  - Nilpotent

    $\bar{0}$: the worst weight

    $\forall a, \exists q$, s.t. $a^q = \bar{0}$: loop has no benefits for all the instances

- Instance Condition $I = \langle \mathcal{G}, \mathcal{L}_{\mathfrak{B}}, B \rangle$
  - Absorptive

    $\bar{1} \preccurlyeq_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(v_1 v_2 v_3 \cdots v_n v_1)$

    Loop has no benefits for this instance

Routing Algebra

$$\mathcal{A} = \langle \Sigma,\ L,\ \lesssim,\ \otimes \rangle$$

$\mathcal{A} = \langle \Sigma,\ L,\ \lesssim,\ \otimes \rangle$	$(X, \oplus, \otimes, \bar{0}, \bar{1})$
Σ, L	X
$\lesssim$	$\oplus$
$\otimes$	$\otimes$

Routing Algebra -- Conditions

- Universal Condition
  - Monotonicity

    $\sigma \precsim l \otimes \sigma$ for each $l \in L$ and for each $\sigma \in \Sigma$

- Instance Condition $I = \langle \mathcal{G},\ \mathcal{L}_{\mathcal{A}},\ \sigma_0 \rangle$
  - Freeness

    For every $v_1 v_2 \cdots v_n v_1$, and every $\{\sigma_1,\ \sigma_2,\ \cdots,\ \sigma_n\}$, there exists i, such that $\sigma_i < \mathcal{L}_A(v_i, v_{i+1}) \otimes \sigma_{i+1}$

Go to slide 15

Stable Path Problem

- No universal condition for SPP

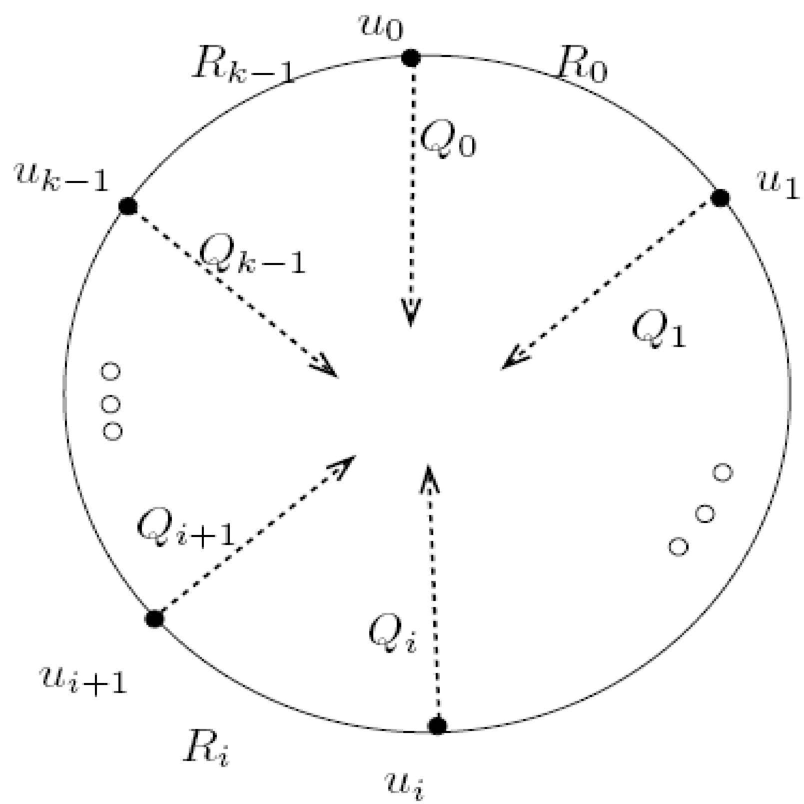
$S_{\mathrm{spp}} = (G, \mathcal{P}, \Lambda)$

- Every node *v* maintains a set of permitted paths $\mathcal{P}^v$ to the destination , and a ranking function λ^v. If $P_1, P_2 \in \mathcal{P}^v$, and $\lambda^v(P_1) < \lambda^v(P_2)$, then P_2 is said to be *preferred over* P_1 .

- The path assignment is a solution if it is stable at each node u
  - The path assignment π maps each node to a path, $\pi(u) \in \mathcal{P}^u$
  - $\mathrm{choices}(\pi, u) = \{(u\ v)\pi(v) \mid \{u,\ v\} \in E\} \cap \mathcal{P}^u$
  - Stable: $\pi(u) = \mathrm{best}(\mathrm{choices}(\pi, u), u)$

A Dispute Wheel

- (1) R_i is a path from u_i to u_{i+1}
- (2) $Q_i \in \mathcal{P}^{u_i}$
- (3) $R_i Q_{i+1} \in \mathcal{P}^{u_i}$
- (4) $\lambda^{u_i}(Q_i) \leq \lambda^{u_i}(R_i Q_{i+1})$



- Which path will u_i choose ?
- $Q_i \leq R_i Q_{i+1} \leq R_i R_{i+1} Q_{i+2}$
  $\leq R_i R_{i+1} R_{i+2} Q_{i+3} \leq \ldots$
  $\leq R_i R_{i+1} \ldots R_{i-1} Q_i \leq \ldots$
- No solution!

Stable Path Problem – Instance Condition

- No dispute wheel
- Revisit instance condition of Routing Algebra

Policy Structure & Routing Structure

- Policy structure

  $S = \langle \mathfrak{X}, \preccurlyeq, \sqsubseteq \rangle$

  - $\mathfrak{X}$: values that will be associated with routes
  - $\preccurlyeq$: $x \preccurlyeq y$ means value x is at least as well-preferred as value y
  - $\sqsubseteq$: $x \sqsubseteq y$ means value y can be constructed from value x

- S-Instance

  $I = \langle \mathcal{G}, \psi, B \rangle$

  - ψ maps paths $P \in \mathcal{P}(v, v_0)$ to elements of $\mathfrak{X}$ such that for all $P \in \mathcal{P}(v, v_0)$ and all $Q \in \mathcal{P}(w, v)$, we have $\psi(P) \sqsubseteq \psi(QP)$

Policy Structure and Routing Structure

- Routing Structure of an S-Instance

  $S_I = \langle \mathcal{P}_\psi^B, \ \preccurlyeq_I, \ \sqsubseteq_I \rangle$

  - Attention!
  - $\preccurlyeq_I$ is the sub-path relation
  - $\sqsubseteq_I$ is the preference relation

$$
\begin{array}{rcl}
\mathcal{P}_\psi^B & = & \{P \in \mathcal{P}(v_0) \mid \psi(P) \notin B\}, \\
P \preccurlyeq_I Q & \Leftrightarrow & \text{there is a path } W \text{ such that } Q = WP, \\
P \equiv_I Q & \Leftrightarrow & \text{head}(P) = \text{head}(Q) \text{ and } \psi(P) = \psi(Q) \\
P \sqsubset_I Q & \Leftrightarrow & \text{head}(P) = \text{head}(Q) \text{ and } \psi(P) \prec \psi(Q) \\
P \sqsubseteq_I Q & \Leftrightarrow & P \equiv_I Q \text{ or } P \sqsubset_I Q.
\end{array}
$$

Policy Structure and Routing Structure

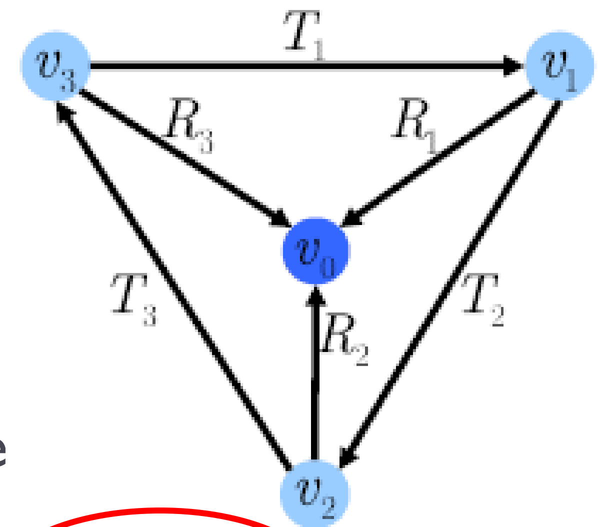
- Then we have :

$$P \preccurlyeq_I Q \quad \Rightarrow \quad \psi(P) \sqsubseteq \psi(Q)$$
$$P \sqsubseteq_I Q \quad \Rightarrow \quad \psi(P) \preccurlyeq \psi(Q)$$
$$P \sqsubset_I Q \quad \Rightarrow \quad \psi(P) \prec \psi(Q)$$

- $R \triangleq R_1 \bowtie R_2$ (join relation):
  - R_1 and R_2 are over the same set X
  - x R z ⇔ There exists $y \in X$ such that $x R_1 y$ and $y R_2 z$.

Policy Structure and Routing Structure

- Instance Condition
  - $\mathscr{R}_I \triangleq (\preccurlyeq_I \bowtie \sqsubset_I)^{tc}$ is anti-reflexive
  - Anti-reflexive: (R is a relation)
    anti-reflexive, if $x \not R x$ for all $x \in X$



  - A bad triangle: an example of dispute whee

$$R_1 \preccurlyeq_I T_1 R_1 \sqsubset_I R_3 \preccurlyeq_I T_3 R_3 \sqsubset_I R_2 \preccurlyeq_I T_2 R_2 \sqsubset_I R_1$$

- Universal Condition
  - $\mathscr{R}_S \triangleq (\sqsubseteq \bowtie \prec)^{tc}$ is anti-reflexive

The Rest of Chau's Paper [CGG06]

- Associate previous frameworks with policy structure and routing structure
  - Path algebras vs. policy/routing structure
  - Routing algebras vs. policy/routing structure
  - Stable path problems vs. routing structure

- Discuss the relation between the universal/instance conditions for all these theoretical frameworks

Routing Algebra Meta-Language (RAML)

- Objective:
  - We can construct more interesting routing protocols
  - All protocols constructed should have a solution

- Motivation
  - Constructing (complex) routing algebras is difficult and tedious
  - Proving monotonicity condition is even worse

- Can we design a meta-language and make thing easier?

$$A = \langle \Sigma, \preceq, L, \oplus, \mathcal{O} \rangle \qquad \mathcal{A} = \langle \Sigma, L, \lesssim, \otimes \rangle$$

RAML (cont'd)

- Technique
  - Design several "natural" operations
  - Define some "basic" routing algebras
  - Construct complex routing algebras from the basic ones by using the operations we define

RAML – Basic Algebras

Algebra	Description	Properties
ADD(n, m)	Natural number addition	Strict monotonicity
MULT(n, m)	Natural number product	Monotonicity
$MULT_r$(n, m)	Real number product	
MAX(n)	Maximum	Monotonicity
MIN(n)	Minimum	
LP(n)	Local preference	
OP(n)	Origin preference	Monotonicity
SEQ(n, m)	Sequences	Strict monotonicity
SIMSEQ(n, m)	Simple sequences	Strict monotonicity
TAGS(T)	Route tags	Monotonicity

RAML – Basic Algebras (cont'd)

- ADD(n, m)
  - $L = \{n, n+1, \dots, m\}$
  - $\Sigma = \{n, n+1, \dots, m\} \cup \{\varphi\}$
  - $i \oplus j = \varphi$, if $i + j \notin \{n, n+1, \dots, m\}$
  - $i \oplus j = i + j$, otherwise
- Multiplications are defined similarly

ADD(1,5)						
$\oplus$	1	2	3	4	5	φ
1	2	3	4	5	φ	φ
2	3	4	5	φ	φ	φ
3	4	5	φ	φ	φ	φ
4	5	φ	φ	φ	φ	φ
5	φ	φ	φ	φ	φ	φ

RAML – Basic Algebras (cont'd)

- MAX(n), MIN(n), LP(n)
  - $L = \{1, 2, \ldots, n\}$
  - $\Sigma = \{1, 2, \ldots, n\} \cup \{\varphi\}$
- OP(n)
  - $L = \{\kappa\}$
  - $\Sigma = \{1, 2, \ldots, n\} \cup \{\varphi\}$

MAX(3)			
$\oplus$	1	2	3
1	1	2	3
2	2	2	3
3	3	3	3

MIN(3)			
$\oplus$	1	2	3
1	1	1	1
2	1	2	2
3	1	2	3

LP(3)			
$\oplus$	1	2	3
1	1	1	1
2	2	2	2
3	3	3	3

OP(3)			
$\oplus$	1	2	3
κ	1	2	3

RAML – Basic Algebras (cont'd)

- SEQ(n, m)
  - $L = \{1, 2, \dots, \mathrm{n}\}$
  - $\Sigma = \{\epsilon\} \cup L^1 \cup L^2 \cup \dots \cup L^{\mathrm{m}} \cup \{\varphi\}$

    (the set of strings over alphabet L with length at most m)
  - $i \oplus \sigma = \varphi$, if $|\sigma| = \mathrm{m}$; $i \oplus \sigma = i :: \sigma$, otherwise
  - $\sigma_1 \preccurlyeq \sigma_2 \Leftrightarrow |\sigma_1| \leq |\sigma_2|$
- SIMSEQ(n, m)
  - $L = \{1, 2, \dots, \mathrm{n}\}$
  - $\Sigma = \{\epsilon\} \cup L^1 \cup L^2 \cup \dots \cup L^{\mathrm{m}} \cup \{\varphi\}$
  - $i \oplus \sigma = \varphi$, if $|\sigma| = \mathrm{m}$ or $i \in \sigma$; $i \oplus \sigma = i :: \sigma$, otherwise
  - $\sigma_1 \preccurlyeq \sigma_2 \Leftrightarrow |\sigma_1| \leq |\sigma_2|$

RAML – Basic Algebras (cont'd)

- TAGS(T)
  - T : type of objects (Integer, String, etc.)
  - $\Sigma = 2^{\mathrm{T}}$ (all finite sets of objects of type T)
  - $L = \{(\mathrm{i}, \sigma) \mid \sigma \in \Sigma\} \cup \{(\mathrm{d}, \sigma) \mid \sigma \in \Sigma\} \cup \{\kappa\}$
  - (i, σ) : insertion of elements
  - (d, σ) : deletion of elements

$\oplus$	σ
(i, σ_1)	$\sigma \cup \sigma_1$
(d, σ_1)	$\sigma \ / \ \sigma_1$
κ	σ

RAML – Lexical Product

- $A \otimes B$
  - Given two routing algebras

$$A = (\Sigma_{\mathrm{A}}, L_{\mathrm{A}}, \preccurlyeq_{\mathrm{A}}, \oplus_{\mathrm{A}}, \varphi_{\mathrm{A}})$$

$$B = (\Sigma_{\mathrm{B}}, \mathrm{L}_{\mathrm{B}}, \preccurlyeq_{\mathrm{B}}, \oplus_{\mathrm{B}}, \varphi_{\mathrm{B}})$$

  - We want to define binary operation $\otimes$ for constructing new routing algebra

$$A \otimes B = (\Sigma, L, \preccurlyeq, \oplus, \varphi)$$

  - Motivation: multiple routing metrics (BGP, OSPF, etc.)

RAML – Lexical Product (cont'd)

- Product Construction I
  - $\Sigma = (\Sigma_\mathrm{A} \,/\, \{\varphi_\mathrm{A}\}) \times (\Sigma_\mathrm{B} \,/\, \{\varphi_\mathrm{B}\}) \cup \{\varphi\}$
  - $(\sigma_{1\mathrm{A}}, \sigma_{1\mathrm{B}}) \preccurlyeq (\sigma_{2\mathrm{A}}, \sigma_{2\mathrm{B}}) \Leftrightarrow \sigma_{1\mathrm{A}} \prec_\mathrm{A} \sigma_{2\mathrm{A}}$ or $\sigma_{1\mathrm{A}} =_\mathrm{A} \sigma_{2\mathrm{A}}, \sigma_{1\mathrm{B}} \preccurlyeq_\mathrm{B} \sigma_{2\mathrm{B}}$
  - $L = L_\mathrm{A} \times L_\mathrm{B}$
  - $(\lambda_\mathrm{A}, \lambda_\mathrm{B}) \oplus (\sigma_\mathrm{A}, \sigma_\mathrm{B}) = (\lambda_\mathrm{A} \oplus_\mathrm{A} \sigma_\mathrm{A}, \lambda_\mathrm{B} \oplus_\mathrm{B} \sigma_\mathrm{B})$ if $\lambda_\mathrm{A} \oplus_\mathrm{A} \sigma_\mathrm{A} \neq \varphi_\mathrm{A}$ and $\lambda_\mathrm{B} \oplus_\mathrm{B} \sigma_\mathrm{B} \neq \varphi_\mathrm{B}$
  - $(\lambda_\mathrm{A}, \lambda_\mathrm{B}) \oplus (\sigma_\mathrm{A}, \sigma_\mathrm{B}) = \varphi$ otherwise

RAML – Scoped Product

- $A \odot B$
  - Given two routing algebras

$$A = (\Sigma_{\mathrm{A}}, L_{\mathrm{A}}, \preccurlyeq_{\mathrm{A}}, \oplus_{\mathrm{A}}, \varphi_{\mathrm{A}})$$

$$B = (\Sigma_{\mathrm{B}}, \mathrm{L}_{\mathrm{B}}, \preccurlyeq_{\mathrm{B}}, \oplus_{\mathrm{B}}, \varphi_{\mathrm{B}})$$

  - We want to define binary operation ⊗ for constructing new routing algebra

$$A \odot B = (\Sigma, L, \preccurlyeq, \oplus, \varphi)$$

  - Motivation: communication inside administrative entities vs. communication between administrative entities

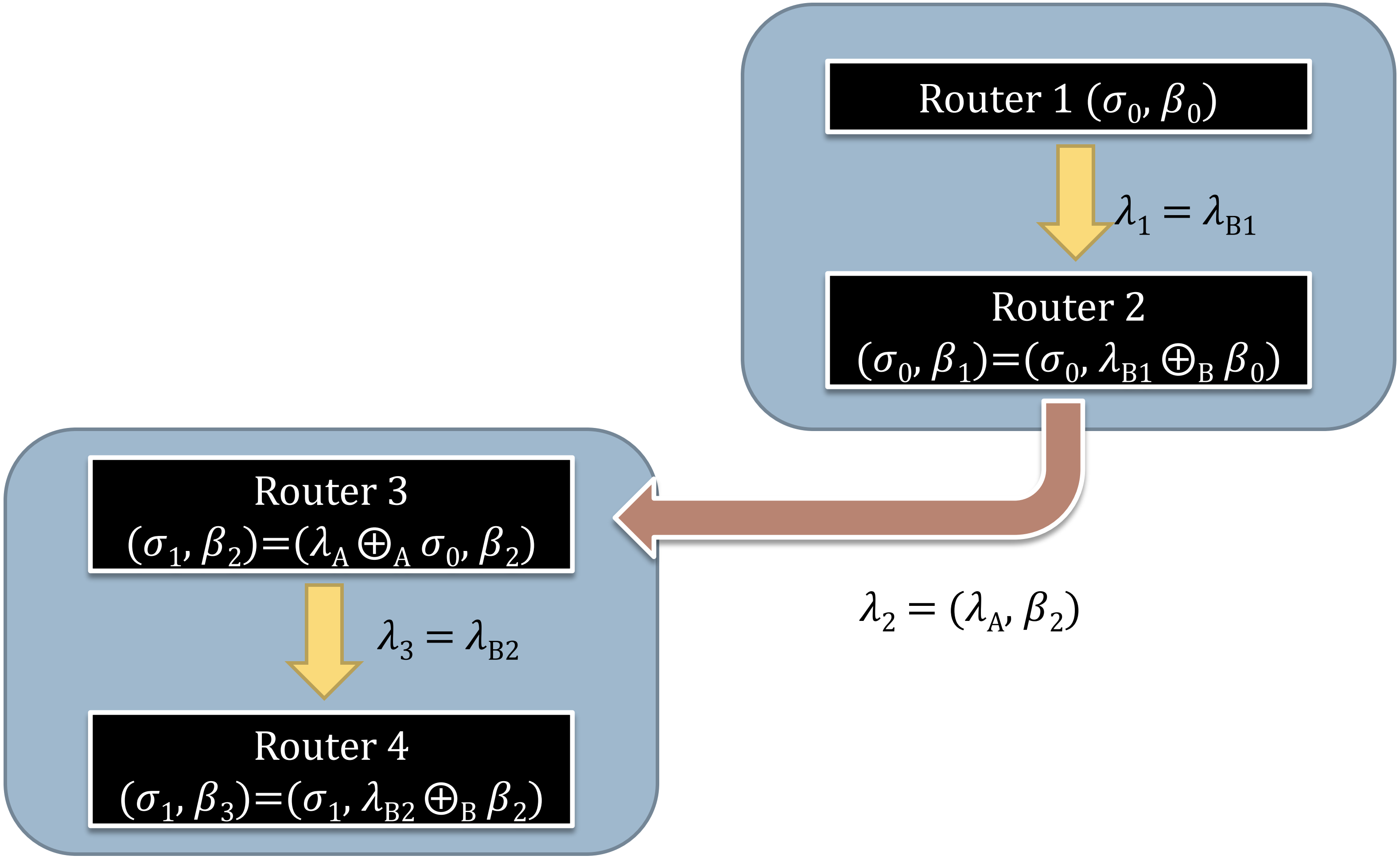
    e.g. BGP = EBGP ⊙ IBGP

RAML – Scoped Product (cont'd)

- Product Construction II
  - $\Sigma = (\Sigma_A / \{\varphi_A\}) \times (\Sigma_B / \{\varphi_B\}) \cup \{\varphi\}$
  - $(\sigma_{1A}, \sigma_{1B}) \preccurlyeq (\sigma_{2A}, \sigma_{2B}) \Leftrightarrow \sigma_{1A} \preccurlyeq_A \sigma_{2A}$ or $\sigma_{1A} =_A \sigma_{2A}, \sigma_{1B} \preccurlyeq_B \sigma_{2B}$
  - $L = (L_A \times \Sigma_B) \cup L_B$
  - Here we assume w.l.o.g that $L_A \times \Sigma_B \cap L_B$ is empty
  - For edges between entities, labels are of the form (λ_A, σ'_B)
  - For edges inside entities, labels are of the form λ_B

$\oplus$	(σ_A , σ_B)
(λ_A, σ'_B)	$(\lambda_A \oplus_A \sigma_A , \sigma'_B)$
λ_B	$(\sigma_A , \lambda_B \oplus_B \sigma_B)$

RAML – Scoped Product (cont'd)



RAML – Disjunction

- $A \lhd B$
  - Given two routing algebras

$$A = (\Sigma_{\mathrm{A}}, L_{\mathrm{A}}, \preccurlyeq_{\mathrm{A}}, \oplus_{\mathrm{A}}, \varphi_{\mathrm{A}})$$

$$B = (\Sigma_{\mathrm{B}}, \mathrm{L}_{\mathrm{B}}, \preccurlyeq_{\mathrm{B}}, \oplus_{\mathrm{B}}, \varphi_{\mathrm{B}})$$

  - We want to define binary operation $\otimes$ for constructing new routing algebra

$$A \lhd B = (\Sigma, L, \preccurlyeq, \oplus, \varphi)$$

  - Motivation: we want to use both A and B in the sense that signatures in Σ_{A} have higher preference than signatures in Σ_{B}

RAML – Disjunction (cont'd)

- Implementation
  - $\Sigma = (\Sigma_A \,/\, \{\varphi_A\}) \cup (\Sigma_B \,/\, \{\varphi_B\}) \cup \{\varphi\}$
  - $\sigma_1 \preccurlyeq \sigma_2 \Leftrightarrow \sigma_1, \sigma_2 \in \Sigma_A, \sigma_1 \preccurlyeq_A \sigma_2$ or
    $\sigma_1, \sigma_2 \in \Sigma_B, \sigma_1 \preccurlyeq_B \sigma_2$ or
    $\sigma_1 \in \Sigma_A\,, \sigma_2 \in \Sigma_B$
  - t : an injection function from Σ_A to Σ_B
  - $L = L_A \cup L_B \cup \{i\}$

$\oplus$	σ_A	σ_B
λ_A	$\lambda_A \oplus \sigma_A$	φ
λ_B	φ	$\lambda_B \oplus \sigma_B$
i	$t(\sigma_A)$	φ

RAML – Monotonicity Preservation

A	B	$A \otimes B$	$A \odot B$	$A \lhd B$
M	M	M	-	M
M	SM	SM	-	M
SM	M	SM	M	M
SM	SM	SM	SM	SM
SM	*	SM	-	-

RAML – Constructing BGP

- Constructing an IGP-like protocol
  - $\mathrm{GN} = \mathrm{ADD}(1, 2^{32}) \oplus \mathrm{SIMEQ}(2^{32}, 30) \oplus \mathrm{TAGS}(\mathrm{String})$
  - $\mathrm{RAN} = \mathrm{ADD}(1, 2^{32}) \oplus \mathrm{SIMEQ}(2^{32}, 30) \oplus \mathrm{TAGS}(\mathrm{String})$
  - $\mathrm{MAN} = \mathrm{ADD}(1, 2^{32}) \oplus \mathrm{SIMEQ}(2^{32}, 30) \oplus \mathrm{TAGS}(\mathrm{String})$
  - $\mathrm{MyIGP} = \mathrm{GN} \odot (\mathrm{RAN} \odot \mathrm{MAN})$

- Constructing the real BGP is more tedious and is omitted here, see Metarouting paper [GS05] for more details

Open Problems and Discussion

- Some of the universal/instance condition seems unnatural
  - The freeness condition for routing algebras is seemingly "translated" from dispute wheel in stable path problem
  - Can we find natural conditions which might reveal more insight of the convergence condition?
- Can we design a theoretical framework that allow security feature?
- Can we design meta-languages for other frameworks?

Thank you!

- Comments Appreciated!