Metarouting

Timothy G. Griffin
Computer Laboratory
University of Cambridge
Cambridge, UK

João Luís Sobrinho
Instituto deTelecomunicações
Instituto Superior Técnico
Lisbon, Portugal

SIGCOMM

August 23, 2005

Computer Science is Largely about Abstractions

Instances	Capturing (almost) all instances
Parsers	Yacc
Data Management Systems	SQL-based systems + application code
Routing Protocols	Metarouting (??)

Why do this?

- No one-size-fits-all IGP
  - BGP is now a widely used IGP!
- Hard to define, standardize, and deploy new routing protocols (or minor modifications to existing protocols)
  - Just standardize Metarouting language and leave it up to operator community to standardize protocols using high-level specs…
- It’s fun!

Idea #1

Let’s try something radical -- keep these separate!

Protocol = Mechanism + Policy + …. ??? …

Mechanism

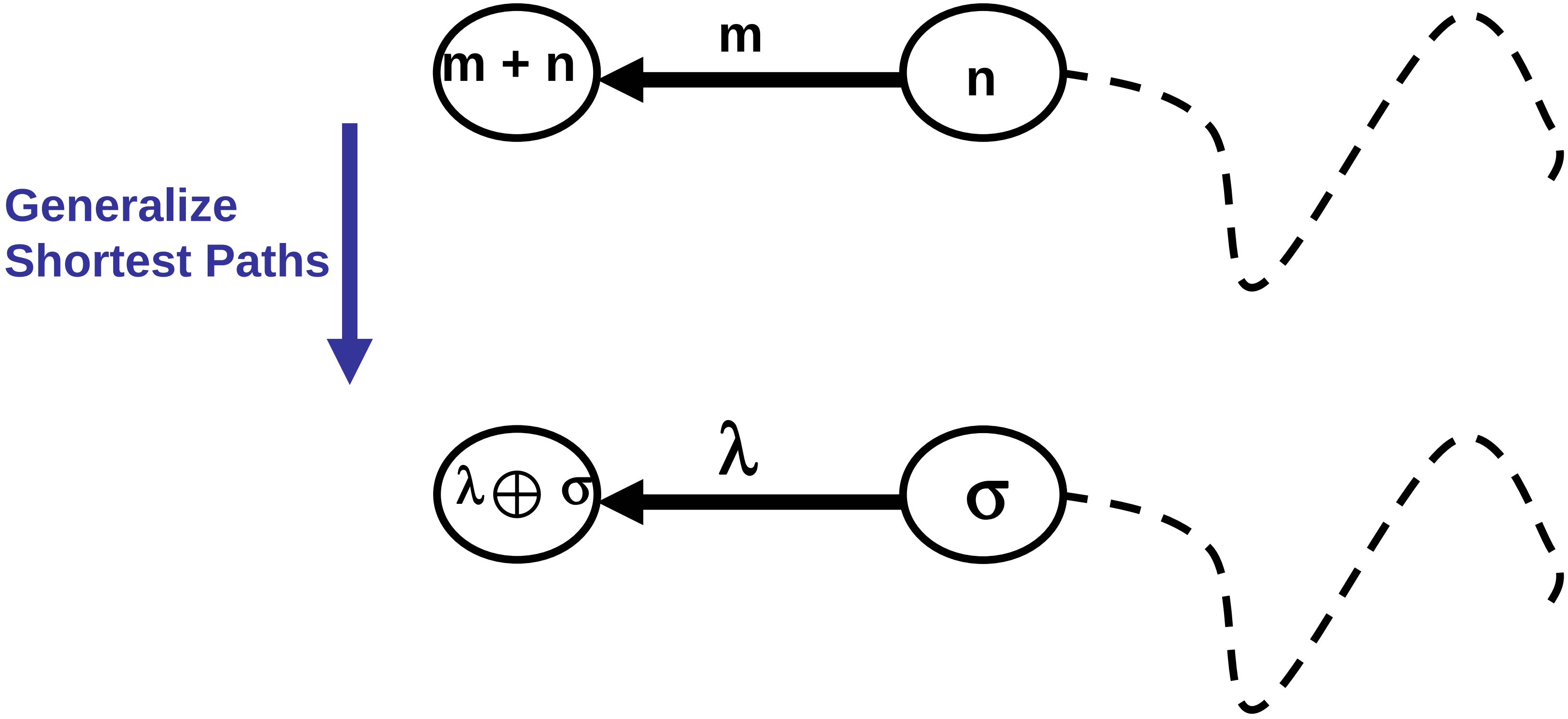
- How are routing messages exchanged and propagated? (example: Link-State, Path Vector)
- How are adjacencies established?
- …

Policy

- How are the attributes of a route described?
- How does configuration attach path characteristics?
- How are best routes selected?
- …

Idea #2

- Use “Routing Policy Algebras” as basis for Policy Component
  - “Network Routing with Path Vector Protocols: Theory and Applications” João Sobrinho. SIGCOMM 2003 (to appear in ToN Oct. 2005)



Routing Policy Algebras

How paths are transformed by application of labels

L : link labels

$\oplus$: $\mathbf{L \times \Sigma \longrightarrow \Sigma}$

$$A = (\underbrace{\Sigma, <=}, \overbrace{L, \oplus}, \underbrace{O})$$

How paths are described and compared

Σ : path signatures

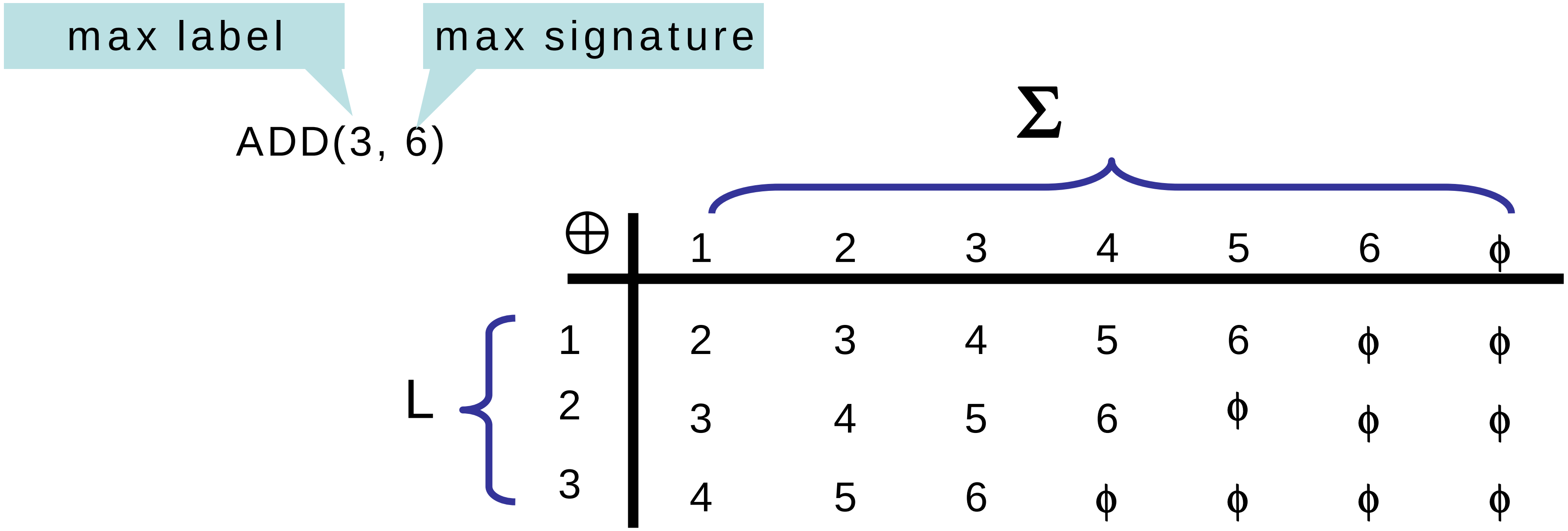
<= is a preference relation over Σ :

(complete) $\forall$ x, y in Σ, x <= y or y <= x (or both)

(transitive) $\forall$x, y, z in Σ, if x <= y and y <= z then x <= z

A subset of signatures that can be associated with originated routes.

Example --- Addition (ADD)



ADD(3, 6)

⊕	1	2	3	4	5	6	ϕ
1	2	3	4	5	6	ϕ	ϕ
2	3	4	5	6	ϕ	ϕ	ϕ
3	4	5	6	ϕ	ϕ	ϕ	ϕ

ADD(n, m) is SM if $0 < n <= m$

Guarantees?

We want protocols that are <u>nice</u>!

→ **always** converge, for **every** network state

→ unique solution (perhaps modulo some ≅ class)

→ no forwarding loops (after convergence)

Correctness

Monotonicity (M): $\forall \; \sigma \,\varepsilon\, \Sigma/\phi \,, \lambda \,\varepsilon\, L \quad \sigma <= \lambda \oplus \sigma$

Strict Monotonicity (SM): $\forall \; \sigma \,\varepsilon\, \Sigma/\phi \,, \lambda \,\varepsilon\, L \quad \sigma < \lambda \oplus \sigma$

Isotonicity (I): $\forall \; \sigma,\beta \,\varepsilon\, \Sigma/\phi \,, \lambda \,\varepsilon\, L \quad \sigma <= \beta \rightarrow \lambda \oplus \sigma <= \lambda \oplus \beta$

	SM	**I**	**Assoc. $\oplus$**
vectoring	★		
Link-state with generalized Dijkstra	★	★	★
Link-state with local vector simulation	★		

An algebra for OSPF?

(hand-coded from **careful** reading of RFC 2328)

$\oplus$	ε	$(1, \varepsilon, \sigma)$	$(1, (1, \mathbf{v}), \sigma)$	$(1, (2, \mathbf{v}), \sigma)$	$(2, \varepsilon, \sigma)$	$(2, (1, \mathbf{v}), \sigma)$	$(2, (2, \mathbf{v}), \sigma)$
$(1, \lambda)$	$(1, \varepsilon, \lambda \oplus \varepsilon)$	$(1, \varepsilon, \lambda \oplus \sigma)$	$(1, (1, \mathbf{v}), \lambda \oplus \sigma)$	$(1, (2, \mathbf{v}), \lambda \oplus \sigma)$	$(2, \varepsilon, \lambda \oplus \sigma)$	$(2, (1, \mathbf{v}), \lambda \oplus \sigma)$	$(2, (2, \mathbf{v}), \lambda \oplus \sigma)$
$(1, (1, \mathbf{v}), \lambda)$	$(1, (1, \mathbf{v}), \lambda \oplus \varepsilon)$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
$(1, (2, \mathbf{v}), \lambda)$	$(1, (2, \mathbf{v}), \lambda \oplus \varepsilon)$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
$(2, \lambda)$	$(2, \varepsilon, \lambda \oplus \varepsilon)$	$(2, \varepsilon, \lambda \oplus \sigma)$	$(2, (1, \mathbf{v}), \lambda \oplus \sigma)$	$(2, (2, \mathbf{v}), \lambda \oplus \sigma)$	ϕ	ϕ	ϕ
$(2, (1, \mathbf{v}), \lambda)$	$(2, (1, \mathbf{v}), \lambda \oplus \varepsilon)$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
$(2, (2, \mathbf{v}), \lambda)$	$(2, (2, \mathbf{v}), \lambda \oplus \varepsilon)$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

<1, …> = intra-area route   <2, …> = inter-area route   <{1,2}, λ> = “normal” route

<{1,2}, <1, v>, λ> = type I external   <{1,2}, <2, v>, λ> = type II external

Routing Algebras are a good start, but…

- The algebraic framework does not, by itself, provide a way of constructing new and complex algebras.
  - Algebra definition is hard…
  - Proofs are tedious…
  - Modifications to an algebra's definitions are difficult to manage…

Idea #3
Routing Algebra Meta-Language (RAML)

A ::= B (base algebras)

| Op(A) (unary operator)

| A Op A (binary operators)

- "Abstract syntax" for generating new Algebras
- Goals
  - Want to automatically derive properties (M, SM, ...) of the algebra represented by an RAML expression from properties of base algebras and preservation properties of operators
  - Simplicity
  - Expressiveness

Lexical Product

$A \otimes B$

$$\begin{array}{c|c} \oplus & (\sigma 1, \sigma 2) \\ \hline (\lambda 1, \lambda 2) & (\lambda 1 \oplus \sigma 1, \lambda 2 \oplus \sigma 2) \end{array}$$

$(\phi, \_) = (\_, \phi) = \phi$

Preference is Lexical order

Preservation properties

A	B	$A \otimes B$
M	M	M
SM		SM
M	SM	SM

This suggests a design pattern for SM:

$$\underbrace{\underbrace{A1 \otimes A2 \otimes \dots Ai}_{\text{all M}} \otimes \underbrace{A(i+1)}_{\text{SM}} \otimes \underbrace{\dots \otimes An}_{\text{don't care}}}_{\text{SM}}$$

Point-wise application?

A ★ B

$\oplus$	$(\sigma 1, \sigma 2)$
$\lambda 1$	$(\lambda 1 \oplus \sigma 1, \sigma 2)$
$\lambda 2$	$(\sigma 1, \lambda 2 \oplus \sigma 2)$

Preservation properties

A	B	A ★ B
SM	SM	M
SM	M	M
M	SM	M
M	M	M

$\kappa(A)$

$\oplus$	$\sigma 1$
λ	$\lambda 1 \oplus \sigma 1$
κ	$\sigma 1$

A	$\kappa(A)$
M	M
SM	M

$$A \bigstar B = \kappa(A) \otimes \kappa(B)$$

Scoped Product

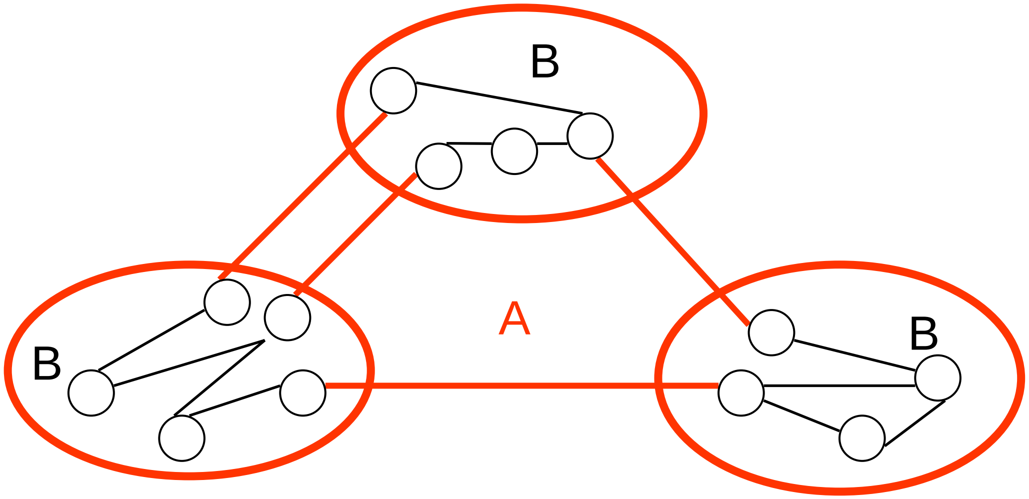
A ⊙ B

⊕	(σ1, σ2)
(λ1, σ3)	(λ1⊕ σ1, σ3)
λ2	(σ1, λ2 ⊕σ2)

Preservation properties

A	B	A ⊙ B
SM	SM	SM
SM	M	M

Can be used to implement
IBGP/EBGP-like “information hiding”



Programmatic Labels

$A = (\Sigma, \mathbf{8}, L, \oplus, \mathit{0})$ $\quad$ $prog(A) = (\Sigma, \mathbf{8}, L, \oplus, \mathit{0})$

$\lambda ::= \lambda \mid \lambda 1; \lambda 2 \mid \text{reject} \mid \text{if } \pi \text{ then } \lambda 1 \text{ else } \lambda 2$

$$\lambda \oplus \sigma = \lambda \oplus \sigma$$

$$(\lambda 1; \lambda 2) \oplus \sigma = \lambda 1 \oplus (\lambda 2 \oplus \sigma)$$

$$\text{reject} \oplus \sigma = \phi$$

$$(\text{if } \pi \text{ then } \lambda 1 \text{ else } \lambda 2) \oplus \sigma = \begin{cases} \lambda 1 \oplus \sigma & \text{if } \pi(\sigma) \\ \lambda 2 \oplus \sigma & \text{o.w.} \end{cases}$$

A	prog(A)
M	M
SM	SM

MyFirstIGP

prog(distance : ADD(2000, 2000)

⊗

router-path: SimpleSeq(1000, 30)

⊗

bandwidth: WIDTH(10000)

⊗

tags: TAGS(string[100]))

This is SM

```
metapolicy MyIGP {
  programatic {
     lex {
          attribute distance {
              label link-weight {
                 mode local;
                 default: 1
              }
              data {
                 addition 2000 2000 ;
              }

          } attribute router-path {
              label router-id {
                  mode local nodal;
               }
               data {
                  Sequence IPv4;
               }

          }
          attribute bandwidth {
              label link-bandwidth {
                 mode local;
              }
              data {
                  WIDTH 1000;
              }
          }
          attribute tags {
             data {
                TAGS string[100];
             }
          }
     }
   }
}
```

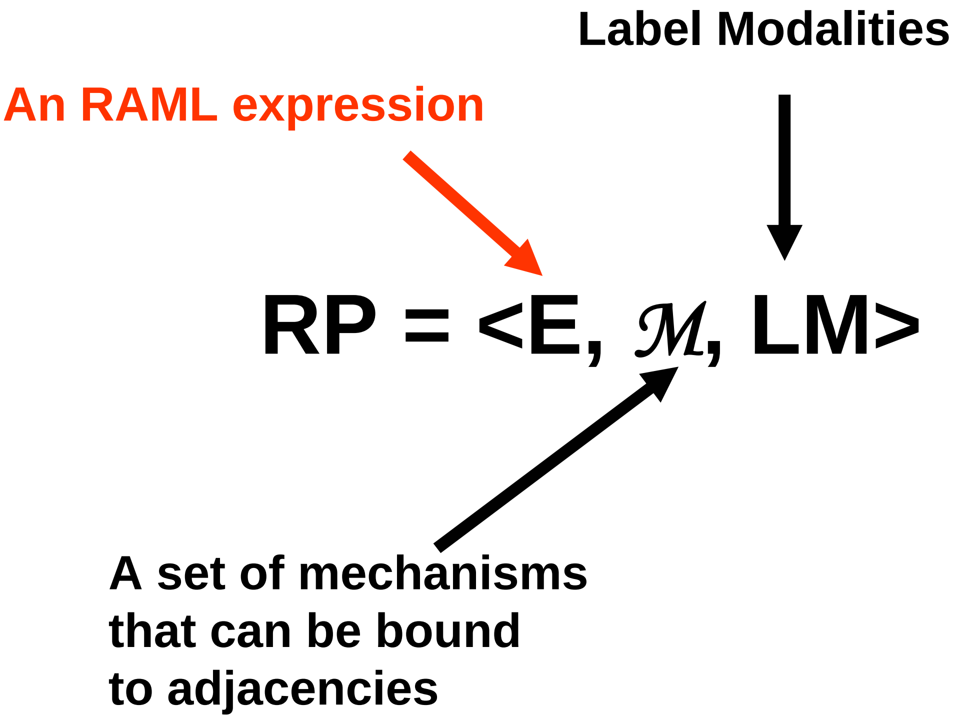
Using MyIGP...

```
policy my-export {
   metapolicy MyIGP;
   if (empty(router-path)) {
      insert tags “sales center NE”
   }
}
```

```
BGP-17 {
   router-id 10.20.20.11;
   mechanism link-state hard-state
   neighbor 10.10.10.10 {
      metapolicy MyIGP;
      import [set link-bandwidth OC-12;
              my-import];
      export my-export;
   }
}
```

```
policy my-import {
   metapolicy MyIGP;
   if (“data center” in tags) {
      if (bandwidth < OC-12) {
       set link-weight 100;
      } else {
        set link-weight 10
      }
   } else {
       set link-weight 20;
   }
}
```

A Metarouting Specification,



Label Modalities

An RAML expression

$$RP = <E, \mathcal{M}, LM>$$

A set of mechanisms
that can be bound
to adjacencies

Ongoing, Future Work

- RAML
  - More operators…
    - At the “protocol level”
    - Inter-operation operators
- Implementation
  - Using XORP (www.xorp.org)
    - With Mark Handley (UCL) and others
  - Hijack BGP
  - Routing Metaprotocol

Disjuntion ("Injection")

$A \underset{\tau}{\blacktriangleleft} B$

$\oplus$	$\sigma 1$	$\sigma 2$
$\lambda 1$	$\lambda 1 \oplus \sigma 1$	ϕ
$\lambda 2$	ϕ	$\lambda 2 \oplus \sigma 2$
ι	$\tau(\sigma)$	ϕ

Preservation properties

A	B	$A \underset{\tau}{\blacktriangleleft} B$
SM	SM	SM
SM	M	M
M	SM	M
M	M	M

… at the "protocol level" ?

$$\langle A, \mathcal{M}1, \mathrm{LM1}\rangle \underset{\tau}{\blacktriangleleft} \langle B, \mathcal{M}2, \mathrm{LM2}\rangle = \langle A \underset{\tau}{\blacktriangleleft} B, \mathcal{M}1 + \mathcal{M}2, \mathrm{LM1} + \mathrm{LM2}\rangle$$

Not sure what "+" means

Nice way of thinking about "administrative distance" …

Perhaps OSPF is really something like

$$\langle \mathrm{AREAS}, \mathit{Path\text{-}Vector}, \mathrm{LM1}\rangle \odot \langle \mathrm{ADD}, \mathit{Link\text{-}State}, \mathrm{LM2}\rangle$$

Migration operators