Formalizing Metarouting in PVS

Anduo Wang Boon Thau Loo

University of Pennsylvania

AFM 09
June 27 2009, Grenoble, France

Introduction

- Metarouting, algebraic framework for routing protocol
  - Models BGP systems (today's de facto Internet routing) with convergence guarantee
- Our contribution: Formalize fragment of Metarouting theory in PVS
  - Heavy and interesting use of PVS theory interpretation: mapping and declaration
- Our goal: extend PVS specification logic with metarouting theory
  - Enable network operator to design BGP system in PVS
  - Free network operator from the tedious low-level and trivial theory consistency checking

Outline

Internet Routing

- *Internet*, network of Autonomous Systems (AS) administrated by Internet Service Provider (ISP)
- *Routing Protocol* computes reachability information
  - Given a destination, an router forwards the packet to its immediate neighbor along the best path
- Internet routing is a combination of Internal Gateway Protocol (IGP) and External Gateway Protocol (EGP)
  - ISP runs its own IGP within an AS
  - EGP enables routing across AS administration borders
- A correct routing protocol must converge!

Policy based *Border Gateway Protocol (BGP)*

- BGP: the de facto Internet routing
- BGP is policy based
  - ISP can influence route decision for economical or performance reasons
  - Import policies select routes to accept
  - Export policies decide routes to be advertised
- BGP is NOT ideal: No convergence guarantee
  - Oscillation, convergence delay, and in the worst case: BGP will not converge at all

Metarouting
Timothy G. Griffin and Joao Luis Sobrinho, SIGCOMM'05

- Algebraic framework for modeling BGP systems with convergence guarantee
  - Abstract routing algebra, mathematical model for routing
  - Base algebras, atomic building blocks
  - Lexical product for route selection, composition operator
- Identify and prove sufficient conditions for protocol convergence: Isotonicity and Monotonicity

Metarouting: Abstract Routing Algebra

A: $A = \langle \Sigma, \preceq, \mathcal{L}, \oplus, \mathcal{O}, \phi \rangle$

sorts Σ (paths), $\mathcal{L}$ (links)

opns $\preceq : \Sigma \times \Sigma \rightarrow$ *bool* (preference relation)

$\oplus : \mathcal{L} \times \Sigma \rightarrow \Sigma$ (label application function)

$\mathcal{O}$: *subset of* $\mathcal{L}$ (origination set)

$\phi : \Sigma$ (prohibited path)

Metarouting: Abstract Routing Algebra

A: $A = \langle \Sigma, \preceq, \mathcal{L}, \oplus, \mathcal{O}, \phi \rangle$

sorts Σ (paths), $\mathcal{L}$ (links)

opns $\preceq: \Sigma \times \Sigma \rightarrow$ *bool* (preference relation)

$\oplus : \mathcal{L} \times \Sigma \rightarrow \Sigma$ (label application function)

$\mathcal{O}$: *subset of* $\mathcal{L}$ (origination set)

$\phi : \Sigma$ (prohibited path)

axioms $\forall_{\alpha \in \Sigma - \{\phi\}} \quad \alpha \preceq \phi \quad$ (*Maximality*)

$\forall_{l \in \mathcal{L}} \quad l \oplus \phi = \phi \quad$ (*Absorption*)

$\forall_{l \in \mathcal{L}} \forall_{\alpha \in \Sigma} \quad \alpha \preceq l \oplus \alpha \quad$ (*Monotonicity*)

$\forall_{l \in \mathcal{L}} \forall_{\alpha, \beta \in \Sigma} \quad \alpha \preceq \beta \implies l \oplus \alpha \preceq l \oplus \beta \quad$ (*Isotonicity*)

Metarouting: Abstract Routing Algebra

A: $A = \langle \Sigma, \preceq, \mathcal{L}, \oplus, \mathcal{O}, \phi \rangle$

sorts Σ (paths), $\mathcal{L}$ (links)

opns $\preceq: \Sigma \times \Sigma \to$ *bool* (preference relation)

$\oplus : \mathcal{L} \times \Sigma \to \Sigma$ (label application function)

$\mathcal{O}$: *subset of* $\mathcal{L}$ (origination set)

$\phi : \Sigma$ (prohibited path)

axioms $\forall_{\alpha \in \Sigma - \{\phi\}} \quad \alpha \preceq \phi$ (*Maximality*)

$\forall_{l \in \mathcal{L}} \quad l \oplus \phi = \phi$ (*Absorption*)

$\forall_{l \in \mathcal{L}} \forall_{\alpha \in \Sigma} \quad \alpha \preceq l \oplus \alpha$ (*Monotonicity*)

$\forall_{l \in \mathcal{L}} \forall_{\alpha, \beta \in \Sigma} \quad \alpha \preceq \beta \implies l \oplus \alpha \preceq l \oplus \beta$ (*Isotonicity*)

- Maximality and absorption describe prohibited path
- Isotonicity and monotonicity **guarantee Convergence!**

Introduction

Background: Internet Routing and Metarouting

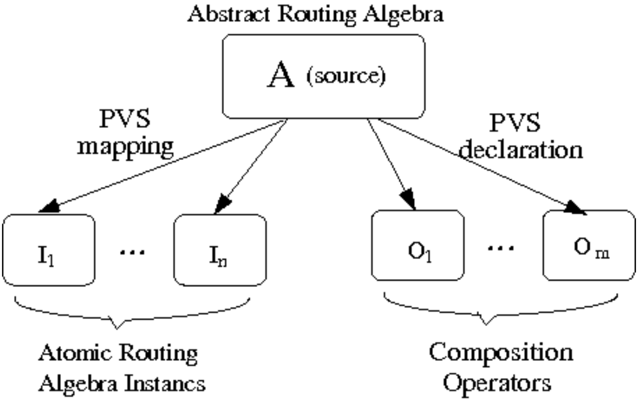
Basic Approach

Compositional Routing Algebra

A Concrete Example

Future Work

Overview of PVS theories



- A: uninterpreted source theory `routeAlgebra`
- I_i: interpreted theory instantiated from A
- O_i: PVS theory taking routing algebra theories as parameters

Abstract Routing Algebra in PVS

```
routeAlgebra: THEORY
 BEGIN
  sig: TYPE+
  label: TYPE+
```

Abstract Routing Algebra in PVS

```
routeAlgebra: THEORY
 BEGIN
  sig: TYPE+
  label: TYPE+
  injected: [label → bool]
  org: TYPE = {l: label | injected(l)}
  prohibitPath: sig
  labelApply: [label, sig → sig]
  prefRel: [sig, sig → bool]
  eqRel(s1, s2: sig): bool = prefRel(s1, s2) ∧ prefRel(s2, s1)
  mono(l: label, s: sig): bool = prefRel(s, labelApply(l, s))
```

Abstract Routing Algebra in PVS

routeAlgebra: THEORY
BEGIN

sig: TYPE+

label: TYPE+

injected: [label → bool]

org: TYPE = {l: label | injected(l)}

prohibitPath: sig

labelApply: [label, sig → sig]

prefRel: [sig, sig → bool]

eqRel(s_1, s_2: sig): bool = prefRel(s_1, s_2) $\wedge$ prefRel(s_2, s_1)

mono(l: label, s: sig): bool = prefRel(s, labelApply(l, s))

pref_complete: AXIOM
$\forall$ (x, y: sig): prefRel(x, y) $\vee$ prefRel(y, x)

absorption: AXIOM
$\forall$ (l: label): labelApply(l, prohibitPath) = prohibitPath

maximality: AXIOM $\forall$ (s: sig): prefRel(s, prohibitPath)

monotonicity: AXIOM $\forall$ (l: label, s: sig): mono(l, s)

isotonicity: AXIOM
$\forall$ (s_1, s_2: sig)(l: label):
prefRel(s_1, s_2) $\Rightarrow$
prefRel(labelApply(l, s_1), labelApply(l, s_2))

END routeAlgebra

Base Algebra for Shortest Path Routing

PVS mapping: Abstract Algebra `routeAlgebra` → Base Algebra `addA`

- PVS mapping makes instantiations of uninterpreted types

$$\begin{aligned} \texttt{sig} &\leftarrow \texttt{upto}(\texttt{m}+1) \\ \texttt{label} &\leftarrow \texttt{upto}(\texttt{n}) \\ \texttt{prohibitPath} &\leftarrow \texttt{m}+1 \\ \texttt{labelApply} &\leftarrow \texttt{APPLY} \\ \texttt{prefRel} &\leftarrow \texttt{PREF} \end{aligned}$$

- PVS mapping generates instances of routeAlgebra axioms as Type Correctness Conditions (*TCCs*)

```
IMP_A_monotonicity_TCC1: OBLIGATION
    FORALL (l: LABEL, s: SIG): mono(l, s)
```

Shortest Path Routing in PVS

Source Theory: Abstract Algebra `routeAlgebra`

Interpreted Theory: Base Algebra `addA`

```
addA: THEORY
 BEGIN
  n: posnat
  m: posnat
  redundant: posnat
  N_M: AXIOM n < m
  LABEL: TYPE = upto(n)
  SIG: TYPE = upto(m+1)
  PREF(s1, s2: SIG): bool = (s1 ≤ s2)
  APPLY(l: LABEL, s: SIG): SIG =
      IF (l+s < m+1)
        THEN (l+s)
      ELSE (m+1)
      ENDIF
  IMPORTING routeAlgebra
              {{sig := SIG, label := LABEL, prohibitPath := m+1,
                labelApply(l: LABEL, s: SIG) := APPLY(l, s),
                prefRel(s1, s2: SIG) := (s1 ≤ s2)}}
 END addA
```

Base Algebra for Provider-Customer, Peer-Peer Guideline

- For economical reasons, ISP reduces use of provider routes, and maximizes availability of customer routes
- $\Sigma(path)$: $C\,/\,R\,/\,P$ (customer/peer/provider path)
- $\mathcal{L}(link)$: $c\,/\,r\,/\,p$ (customer/peer/provider link)
- $\oplus$ (label application):

$\oplus$	C	R	P
c	C	C	C
r	R	R	R
p	P	P	P

- $\preceq$ (preference relation): $C \preceq R, \quad R \preceq P, \quad C \preceq P$

Provider-Customer, Peer-Peer Guideline in PVS

For simplicity, rename labels and signatures:
$c \leftarrow 1, r \leftarrow 2, p \leftarrow 3$ and $C \leftarrow 1, R \leftarrow 2, P \leftarrow 3$

```
lpA: THEORY
 BEGIN

  SIG: TYPE = upto(3)
  LABEL: TYPE = upto(3)
  IMPORTING routeAlgebra
             {{sig := SIG, label := LABEL,
             labelApply(l: LABEL, s: SIG) := l,
             prefRel(s1, s2: SIG) := (s1 <= s2),}}
 END lpA
```

Lexical Product $\otimes$ and Route Selection

- Lexicographic comparison models route selection
  - Most important attribute of each route is compared first, if no decision is reached, the next attribute is considered
- Lexical Product $A \otimes B$ built from existing algebras: A, B
  - Models a routing protocol with multiple attributes
  - More important attributes are handled by A, and the less important by B

Lexical Product $A \otimes B$ in PVS

PVS *declaration* and *mapping* ensures resulting algebra $A \otimes B$ is a valid routing algebra, i.e. $\otimes$ is closed under abstract routing algebra

```
lexProduct[A: THEORY routeAlgebra, B: THEORY routeAlgebra]: THEORY
 BEGIN

  SIG: TYPE = [A.sig, B.sig]

  LABEL: TYPE = [A.label, B.label]

  APPLY(l: LABEL, s: SIG): SIG =
      (A.labelApply(l`1, s`1), B.labelApply(l`2, s`2))

  PREF(s1, s2: SIG): bool =
      A.prefRel(s1`1, s2`1) ∨
       (A.eqRel(s1`1, s2`1) ∧ B.prefRel(s1`2, s2`2))

  IMPORTING routeAlgebra
                 {{sig := SIG, label := LABEL,
                   labelApply(l: LABEL, s: SIG) := APPLY(l, s),
                   prefRel(s1, s2: SIG) := PREF(s1, s2)}}

 END lexProduct
```

A Concrete BGP system

- Route paths are measured in terms of customer-provider relationship and distance cost
  - Customer-Provider Peer-Peer guideline must be enforced
  - Once customer-provider policy is satisfied, ISP wants least-cost (shortest) paths
- Decompose this BGP system into two sub-components
  - Sub-component A for customer-provider guideline
  - Sub-component B for shortest-path
  - Check the sub-component A first, and only use B to break tie

Simple BGP system in PVS
Top Level Algebra: *BGPsystem*

```
simpleBGP: THEORY
 BEGIN

  IMPORTING AlgebraInstance, lexProduct

  BGPsystem: THEORY = lexProduct[A_2, B_2]

 END simpleBGP
```

Simple BGP system in PVS
Sub-Component Algebras: A_2, B_2

```
AlgebraInstance: THEORY
 BEGIN

  IMPORTING addA{{n := 16, m := 16}}

  IMPORTING lpA{{c := 3}}

  A2: THEORY =
        routeAlgebra
             {{sig = lpA.SIG, label = lpA.LABEL,
               labelApply(l: lpA.LABEL, s: lpA.SIG) = l + s, prohibitPath = 4,
               prefRel(s1, s2: int) = (s1 ≤ s2)}}

  B2: THEORY =
        routeAlgebra
             {{sig = addA.SIG, label = addA.LABEL,
               labelApply(l: addA.LABEL, s: addA.SIG) = mod(l + s, 16),
               prohibitPath = 17,
               prefRel(s1, s2: addA.SIG) = (s1 ≤ s2)}}


 END AlgebraInstance
```

Conclusion, Recap

- Our contribution: Formalize fragment of Metarouting theory in PVS
  - Heavy and interesting use of PVS theory interpretation: mapping and declaration
- Our goal: extend PVS specification logic with metarouting theory
  - Enable network operator to design BGP system in PVS
  - Free network operator from the tedious low-level and trivial theory consistency checking

Future Work

- Provide full support for modeling complex BGP systems via metarouting
  - Encode more base algebras and composition operators presented in recent metarouting development
- Relaxed algebra for BGP systems with non-monotonic attributes
  - MULTI-EXIT-DISCRIMINATOR (`MED`) expresses router's preference regarding which neighbor to use
  - NON monotonic attribute: $a \preceq b$, $b \preceq c$, $c \preceq a$
  - Routers in an AS cannot express a monotonic ranking

Thank you!

Questions?