Introduction to Routing Algebras

Alberto García

References

Specific to Routing Algebras

- [Grif05] Metarouting. Timothy G. Griffin, Joao Luis Sobrinho, SIGCOMM 2005.
- [Sobr03] Network routing with path vector protocols: theory and applications. João Luis Sobrinho. Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. 2003.

To go further

- [Chau06] Towards a Unified Theory of Policy-Based Routing. Chi-kin Chau, Richard Gibbens, Timothy G. Griffin. INFOCOM 2006.

Related work on safety (not Routing Algebras)

- [Gao01] Stable Internet Routing Without Global Coordination. Lixin Gao and Jennifer Rexford. Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2000.
- [Grif02] The Stable Paths Problem and Interdomain Routing. Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. IEEE Transactions on Networking. Volume 10, Issue 2 (April 2002). Pages 232-243.
- [Bert92] Data Networks, D. Bertsekas, Gallagher. Prentice Hall, 2nd Edition. 1992.

Why Routing Algebras?

- Routing protocols need to assure (= mathematical proofs) safety, i.e. that
  - A unique solution for route assignment exists
  - The protocol proposed (the way information is exchanged by the protocol) always reaches (eventually) the unique solution
- Safety may be a general property of a protocol, or may apply to certain scenarios (certain topologies, restrictions in the initial conditions, etc.)
  - Eg.: A distance vector protocol minimizing distance to a destination is safe (for any topology, any initial conditions
- To prove safety, you can follow ad-hoc approaches. For example, for BGP:
  - Restrict the configuration (valley-free) and use graph theory to prove safety [Gao01]
  - Have the intuition that dispute wheels are relevant for this problem, and use graph theory to prove that in fact are [Grif02]
- Look for a more systematic approach based on a general theory for routing, which may provide more results than what we already know:
  - Game theory
  - Routing Algebras, derived from path algebras

Context for BGP routing analysis

- **Just consider the routing problem for a single prefix**
  - **Since each prefix is processed independently, multi-prefix operation is assured by safety for a single prefix**
- **Focus on policy elements:**
  - **Information propagated for a prefix**
    - ✓ **For a positive advertisement, the attributes (AS_PATH, LOCAL_PREF, NEXT_HOP…)**
    - ✓ **The prefix is no longer reachable**
  - **Decision process on each node**
    - ✓ **For BGP, selection process (discard routes with lower LOCAL_PREF…)**
  - **Propagation process on each node**
    - ✓ **For BGP, does a node filters-out a prefix for a given neighbor?**
    - ✓ **Egress processing: for EBGP, include your own AS number, change NEXT_HOP…**
- **Fortunately, we do NOT need to consider 'mechanisms' how detecting if your neighbor is down, which are the particular messages…)**
  - **Lost messages, or reordered messages coming from a neighbor router (remember BGP is deployed over TCP)**
  - ***Mechanisms* (the particular mechanisms used to detect if a neighbor is down, the particular messages to exchange routing information…)**
- **Remember BGP is a 'Bellman-Ford' protocol**
  - **"Each node chooses at any time a local-optimal path with respect to the paths learned from each of its out-neighbors to reach the destination."**

What [the hell] is a Routing Algebra

- Algebra in general, deals with sets and operations over the elements of the sets
- A routing algebra A is a tuple $A = (\Sigma, \preceq, L, \oplus, \vartheta)$

Σ is a set of {signatures describing a path}

- σ is ONE signature, a route.
- σ is an element of Σ
- σ is not THE PATH, but HOW THE PROTOCOL DESCRIBES the path
  - ✓ Eg. in distance vector (with hop count): σ is distance to the origin

$\preceq$ is a preference relation over the signature

- $\sigma_A \preceq \sigma_B$ is read as 'σ_A is preferred to σ_B'
- It represents the selection process being made among all the possible routes available to a destination.
  - ✓ Eg. In distance vector: $\preceq$ is 'prefer shorter distance'
  - ✓ Practical eg.: $2 \preceq 4$

What [the hell] is a Routing Algebra

L is the set of labels which can be used to process routing information sent to the next node (λ is a label, an element of *L*)

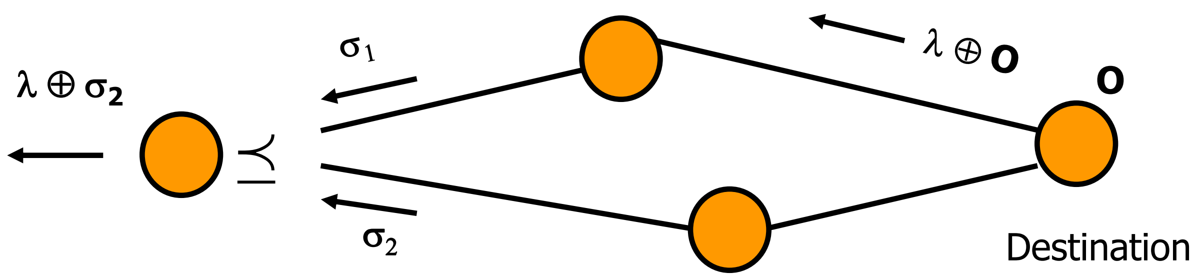
- Eg. In distance vector: the only processing being made is adding 1 (one hop), so $L=\{ 1 \}$ and $\lambda = 1$

$\oplus$ is how you operate the labels with the signature (route) preferred when it is being sent to the next node, i.e. a label application, that maps $L \times \Sigma$ to Σ

- Distance vector: $\oplus$ is the 'natural number addition' operation
- Each time a signature is processed to be sent to another node, the route propagated is $\sigma_{out} = 1 + \sigma_{selected\_in}$

ϑ is the origination set describing the signatures that can be associated with originated routes. $\vartheta \subseteq \Sigma$

- Distance vector: The only possible originating signatures is 0 (when the initial node generates a route, the distance is 0)



Routing Algebra requirements and properties

For an an algebra to be a Routing Algebra, we require

- $\preceq$ to be a total order, i.e. being transitive, anti-symmetric and complete; i.e. every element is ordered compared to any other
- If $\varnothing$ is one of the possible signatures (this is optional), meaning that a route does not exist, then it must hold that
  - Any route is preferred to $\varnothing$ so $\alpha \prec \varnothing$ (*maximality*)
  - For every $l \in L$, $l \oplus \varnothing = \varnothing$ (*absortiveness*)
    - ✓ Both are reasonable for routing protocols

In addition, a Routing Algebra can be

- **Strict Monotonic (SM)**: $\forall\ \sigma \in \Sigma$ and for all $\lambda \in L$, $\sigma \prec \lambda \oplus \sigma$
  - For routing algebras, read this as 'as the route propagates, it (always) becomes strictly less preferred'
  - Other way: is it possible for a node who were receiving σ and $\lambda \oplus \sigma$, to prefer the last one?
- **Monotonic (M)**: $\forall\ \sigma \in \Sigma$ and for all $\lambda \in L$, $\sigma \preceq \lambda \oplus \sigma$

Note that SM => M

A Routing Algebra for EBGP

- **Route description**
  - **Σ: list of attributes of a route (AS_PATH, NEXT_HOP,…)**
- **Route selection**
  - **$\preceq$: rules stated for BGP route selection**
  - **It is a complex process: First apply policies to routes (eg. set LOCAL_PREF), then discard routes not having the highest LOCAL_PREF, then …**
- **Outgoing processing**
  - ***L* and ⊕: concatenate own AS# (or many, if AS_PATH prepending is being used) in AS_PATH, change NEXT_HOP, delete LOCAL_PREF (we are assuming EBGP operation), etc.**
    - ***L* can be ∅, which means that this route cannot be propagated (i.e. the route is filtered-out)**
      - For every $\ell \in$ L, $\ell \oplus \varnothing = \varnothing$ (it is said that this ⊕ is *absortive*)
- **The origination signatures $\vartheta \subseteq \Sigma$ is a restriction on the signatures that can be legally attached to routes injected in the protocol:**
  - **Assume the node originating a route sends to its neighbors $L \oplus \vartheta$**
  - **ϑ states that the AS_PATH is originally empty**

Safety conditions for Bellman-Ford protocols modelled as Routing Algebras

Now we have routing algebras, but… can we derive from this nice properties for routing protocols?

Note I'm showing results, not demonstrations (refer to [Sobr03] for this)

Statements

- Bellman-Ford protocol with SM routing algebra => safe
  - If { as the route propagates it becomes strictly less preferred }, then { a Bellman-Ford protocol will eventually converge to a solution }
  - Link with prior knowledge: SM implies Freeness (property in algebra equivalent to absence of Dispute Wheel)
    - ✓ So dispute wheels are behind, again !
- Bellman-Ford protocol with M routing algebra => exists at least one solution
  - But the protocol may not be able to find it
  - In any case, M does not guarantee safety ☹
- Bellman-Ford protocol with routing algebra that is not M => it doesn't converges

Safety conditions for Bellman-Ford protocols modelled as Routing Algebras

- **Scope for the conditions**
  - **We are not assuming any particular topology, so it works for ALL topologies**
  - **We are not assuming any particular definition of $\prec$ or of $\oplus$, as long as they fulfill the conditions stated**
- **However, it may be that a protocol is not SM, but certain configuration restrictions may make the result (a given protocol under certain configuration conditions, etc.) to be SM**
  - **Note that configuration restrictions affect to $\prec$ and $\oplus$, so SM may now hold**

Safety of distance vector protocols

- **What about Distance vector (hop count) protocols)?**
  - $\sigma \prec \lambda \oplus \sigma$?
  - **While a route propagates (*positively*), $\sigma \prec 1+ \sigma$ is true. So it is safe**
  - **However, it is not true when a route disappears (a route a node propagated earlier is now preferred to a $\varnothing$ route recently received), so in this case it counts-to-infinite**
    - **If we include mechanisms to suppress/limit count-to-infinite, theory says it converges!**

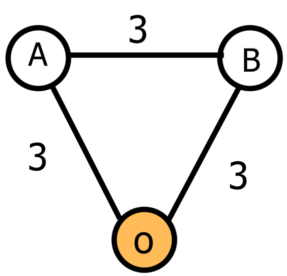
- **Disclaimer: this assumes that messages are not lost, reordered…**
  - **See [Berts] to see a prove of this for a general scenario – not using Routing Algebras**

Safety of (max, min) Bellman-Ford protocols

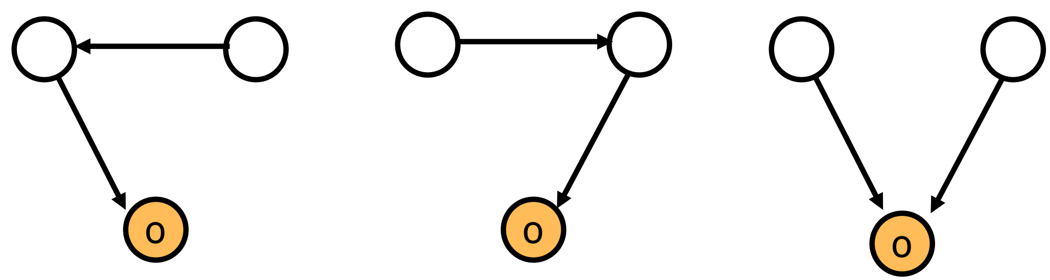
We call this algebra WIDEST PATH or GREATEST CAPACITY ROUTING – seeks for the maximum assured bandwidth to destination

- **Σ : bandwidth to destination**
- **$\preceq$: 'select route with highest bandwidth to destination'**
  - **This is the 'max'**
- **⊕ : take the minimum of selected route and outgoing link**
  - **This is the 'min'**
- **ϑ, the origination set describing the signatures that can be associated with originated routes: ∞**

- **This is M, but not SM**
  - **In general, many solutions**
  - **Can you think of an activation in which oscillates? (note: think that A and B prefer the path through each other)**



SOLUTIONS:



Safety for BGP?

- $\preceq$: for BGP is a lexicographic preference relation
  - *Lexicographic* means 'use rule 1 and compare, if rule 1 does not decide, then use rule 2 and compare…'
  - First compare LOCAL_PREF and discard lower values, etc.
- $\sigma \prec \lambda \oplus \sigma$? (is BGP SM?)
  - NO! First reason for this is that LOCAL_PREF is not SM: a node can prefer any route… just based on the neighbor, not on how the signature of the route varies, i.e. not on $\lambda \oplus \sigma$
  - This means that there may be cases in which configuration (i.e. $\prec$ and/or $\oplus$) can make the protocol to not converge
    - ✓ We already know: dispute wheel configuration
  - Note that it is not even M

Can we force BGP to be SM?

- **Maybe:**
  - **Let's assume that LOCAL_PREF is the only problem**
  - **Then, require that all routers are not going to configure LOCAL_PREF**
  - **This means that LOCAL_PREF is in fact not applied. If the rest of the rules make the routing algebra being SM, then we are done!**
- **Problem: removing LOCAL_PREF may not be reasonable for network operators**
  - **Business relationships …**

Can we force BGP to be SM? (second try)

We need to understand better what do we need to change (from LOCAL_PREF, any other …)

- **[Grif05] To 'make SM' a lexicographic preference, in which rules A_1, A_2, A_3,…A_n are applied sequentially, we need to find a k for which**

  A_1… A_{k-1} are all M at least, and A_k is SM

  Then, we don't need to care about A_{k+1} … A_n!

- **So for BGP**
  - **LOCAL_PREF: we already know it is not even M**
  - **Origin: (assume it is not changed by intermediate nodes) => it is M**
  - **AS_PATH length: SM**
    - ✓ **Search for shortest distances is always a safe strategy!**

  - **Note: MED is problematic, because it is not a preference order, so we assume it is not applied**

Can we force BGP to be SM? (second try)

Strategy:

- **AS_PATH is SM**
- **We need to 'promote' LOCAL_PREF at least to be M, by stating some (reasonable) configuration requirements:**
  - **Force the use of (typical) customer/provider relationships, using the following LOCAL_PREF table**

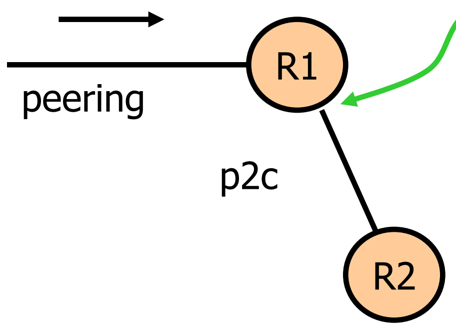
Type of link from which a node receives a route

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

Type of link to which the node will propagate the route

LOCAL_PREF 'Treatment' that the route will receive in the next node ($C \prec R \prec P$)

Can we force BGP to be SM? (second try)



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

- **C=1, R=2, P=3 (simplification, but assume always C < R < P)**
  - **When applying preferences, C < R < P (customer more preferred)**
- **R1 receives a route from a peer, and sends it to R2, a customer of R1 (so R2 sees the route as coming from a Provider)**
  - **LOCAL_PREF evaluation at R1: R(2)**
  - **LOCAL_PREF evaluation at R2: P(3)**
    - **In this case route evaluated at R1 is preferred (strictly) to route evaluated at R2, i.e. according to LOCAL_PREF $\sigma \preceq \lambda \oplus \sigma$**
    - **This case is monotonic (some combinations are equally preferred)**
      - Check all the cases, and you will see that the whole selection process + egress processing is M !
- **To have this, you need all routers implementing this table for LOCAL_PREF selection and label assignment**
- **This is similar to [Gao01]**
  - **Restricts configuration, assuring in this case safety**
  - **Although it says that transferring routes from a peer to a peer does not result in any safety trouble**

Conclusions

- Routing Algebra can be used to reason about stability of complex routing protocols/scenarios…
  - IBGP [SIGCOMM 09]
  - Multipath routing [Chi-kin]
- Some results are tied to Routing Algebra theory, so we only need to concentrate on some parts of the analysis (are fulfilled M and/or SM properties?)
- Analysis confirms results obtained using other means (graph theory, game theory)
  - So far, not 'extraordinary' results, but provides simple intuitions to reason about routing protocols