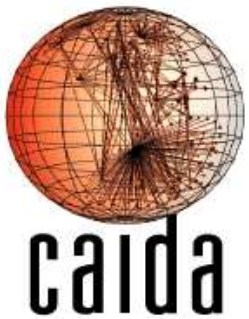
Atomised Routing

Patrick Verkaik, Andre Broido, kc claffy

CAIDA / NLnet Labs / RIPE NCC

http://www.caida.org/projects/routing/atoms/

work in progress



Motivation

- Observation: many prefixes share AS path in all RouteViews peers
- BGP policy atom: set of prefixes that share AS path
- Routed the same (to a large degree)
- 1 March 2003 RouteViews data:
  - around 31000 atoms
  - covering around 117000 prefixes
  - (and 15000 ASes)
- 2002 study by Tel Aviv University (RIPE data):
  - in 8 hours only 2-3% of prefixes change atom membership
  - in 1 week 14% change atom membership

Apply to today's routing?

- Summarise prefixes of atom into one routing entry
- Incorporate into BGP

Possible benefits:

- Perform routing computations per atom, not per prefix
- Shrink routing table and FIB size in default-free routers
- Hide updates to prefixes (abstraction, compare: CIDR aggregation)

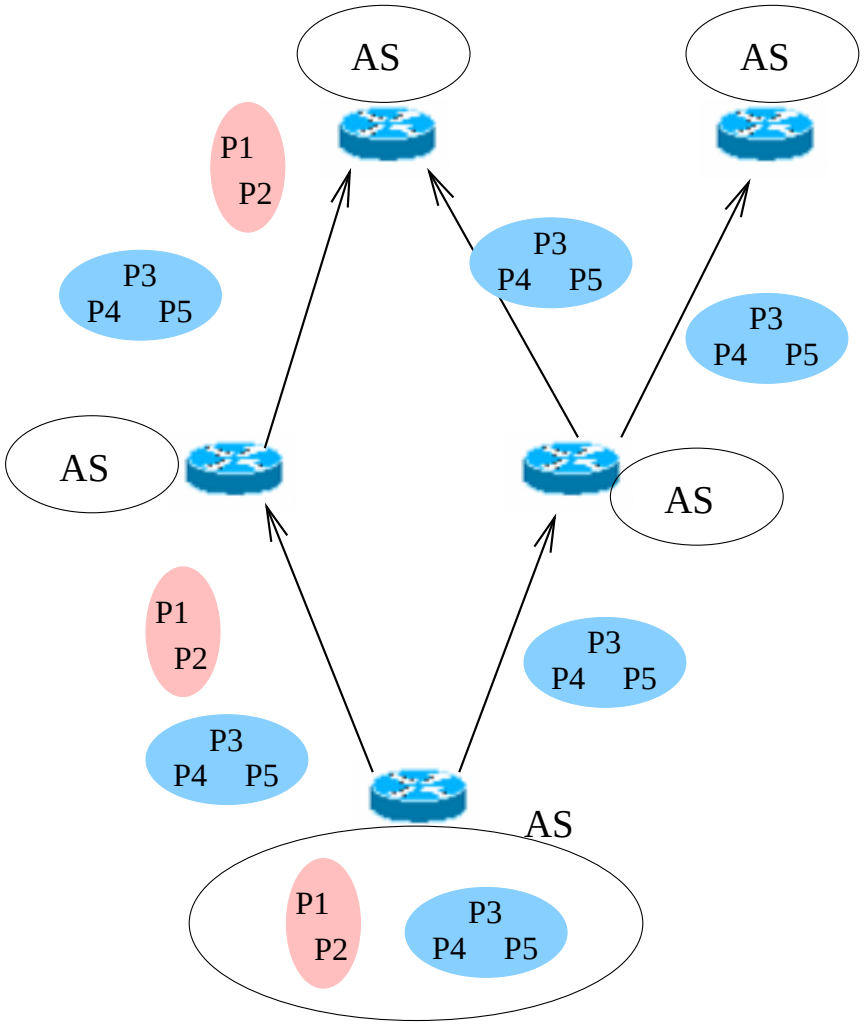
Architecture

- Group prefixes into *atoms*
- Route and distribute atoms in modified BGP
- Deployment

Architecture — Create Atoms

- To be declared by origin ASes
- These ASes partition prefixes into atoms and announce
- Other ASes must *agree* to route prefixes the same
- Prefixes can be IPv4 or IPv6

Architecture — What is an atom?

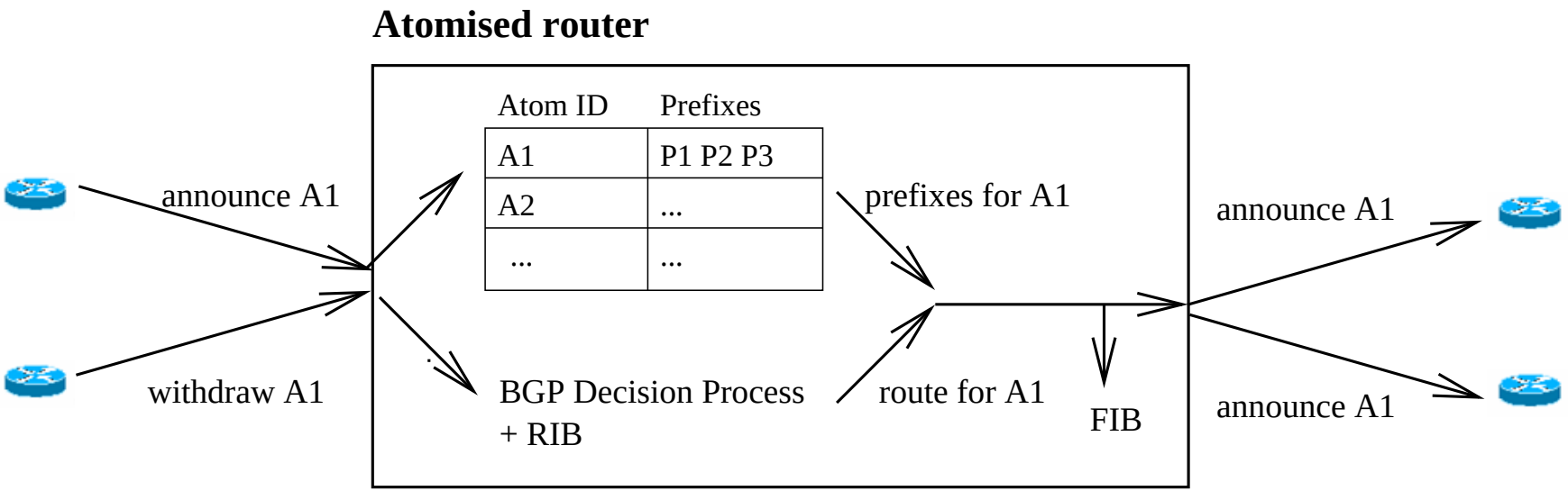


Architecture — Routing and Distribution

Protocol has two functions:

- Atom routing
  - Atom is represented by an atom ID (syntactically a prefix)
  - BGP routing computations on these atom IDs
- Atom distribution
  - Distributes mapping of atom ID <-> prefix
  - BGP extension (or another protocol)
  - Light-weight: no BGP routing computations
  - No *delayed* convergence for withdrawals?

Architecture — Routing and Distribution



BGP Update Messages

Announce atom

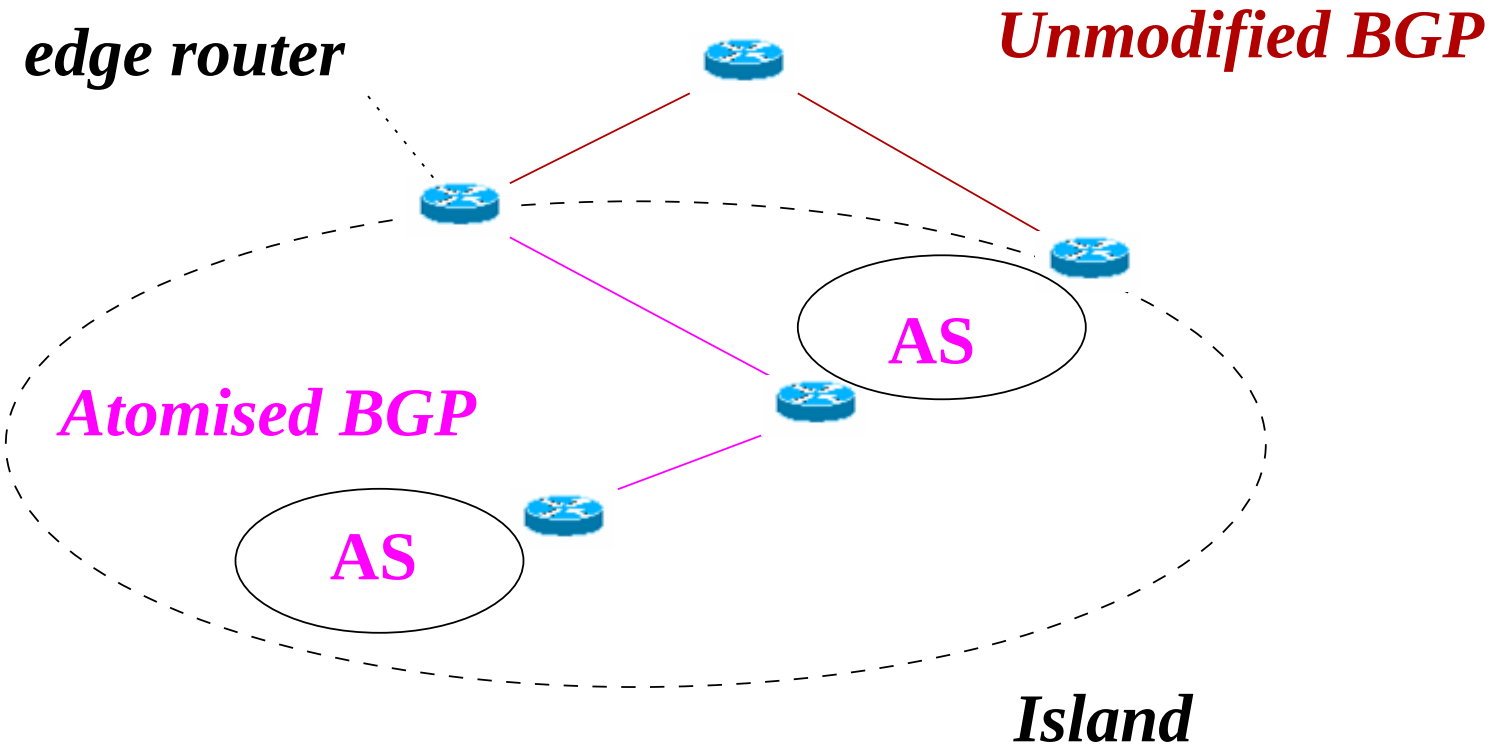
Announce	A1
Withdraw	
Atom attrib	P1 P2 P3

Withdraw atom

Announce	
Withdraw	A1
Atom attrib	

Architecture — Deployment

- Testing and incremental deployment: islands
  - Confine atomised routing to an island
  - Incremental deployment: grow the island



Implementation

Preliminary implementation of atomised routers

- In Zebra: free routing software (GNU license)
- Atoms declared using router configuration language
- Slightly different version of attributes

Unresolved issues

- Many policies not in AS path!
- Handling link failures
  - Atom splits, *or*
  - Use reachability bits
- Atom distribution convergence
  - During convergence mappings of neighbours inconsistent
  - Router needs mapping per neighbour
  - Decision process to resolve mapping conflicts
- Scalable atom computation possible?

Questions we have

- Importance of table size?
  - Entries vs bytes vs dynamics
  - FIB / RIB
- Do routers intelligently handle ‘equivalent’ prefixes?
- Encapsulation: how inefficient?

Questions?

Acknowledgements

Andrew Partan
Bill Woodcock
Cengiz Alaettinoglu
Daniel Karrenberg
Dave Meyer
Evi Nemeth
Frances Brazier
Frank Kastenholz
Geoff Huston
Henk Uijterwaal
Maarten van Steen
Nevil Brownlee
Omer Ben-Shalom
Ronald van der Pol
Ted Lindgreen
Teus Hagen
Wytze van der Raay

http://www.caida.org/projects/routing/atoms/

Shrinking Table Sizes in Default-Free Routers

- Edges of island:
  - carry all prefixes
  - contain atom ID <-> prefix mapping
  - encapsulate IP packets
    outer address is based on atom ID
- Routers inside island:
  - only carry atom IDs
  - can be unmodified BGP implementations
    (since atom ID looks like a prefix)