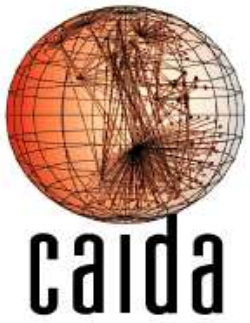
# Beyond CIDR Aggregation

Patrick Verkaik, Andre Broido, Young Hyun, kc claffy

CAIDA / NLnet Labs / RIPE NCC

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





## Outline of talk

- Introduction
- Atoms architecture
- Incremental deployment
- Prototype
- Analysis and simulation
- Future work

## Motivation

- Observation: many prefixes share AS path in all RouteViews / RIPE peers
- BGP policy atom: set of prefixes that share AS path
- Equivalent in terms of routing

## Number and stability

1 Nov 2003 RouteViews data:

- around 35K atoms
- covering around 127K prefixes
- (16K ASes)

Stability over 8 hours:

- 4.9% of atoms undergo prefix membership change (8 May 2003 RouteViews data)
- 2-3% of prefixes change atom membership (Tel Aviv University, 2002)

## Apply to routing?

- Summarise prefixes of atom into one routed object
- Incorporate into BGP

Reduce number of routed objects in Default-Free Zone (DFZ):

- Shrink routing tables and forwarding tables
- Perform routing updates per atom, not per prefix

## Current BGP techniques and their limitations

- CIDR aggregation. *But* [BGP-GROWTH]:
  - Multihoming and inbound traffic $\longrightarrow$ deaggregation
  - Fragmented address space cannot be aggregated
  - Failure to aggregate
- Rate limiting and dampening
- Pack multiple prefixes in single BGP update message. *But*:
  - No effect on number of routes
  - Per-prefix update processing remains
  - Only for prefixes with identical attributes (e.g. different origin AS $\longrightarrow$ separate update message)

Currently coping, but what about future?

- IPv6
  - raises upper bound on number of routes
  - multihoming practices operationally disabled [RFC2772]
- 32-bit AS numbers: increased multihoming by small sites?

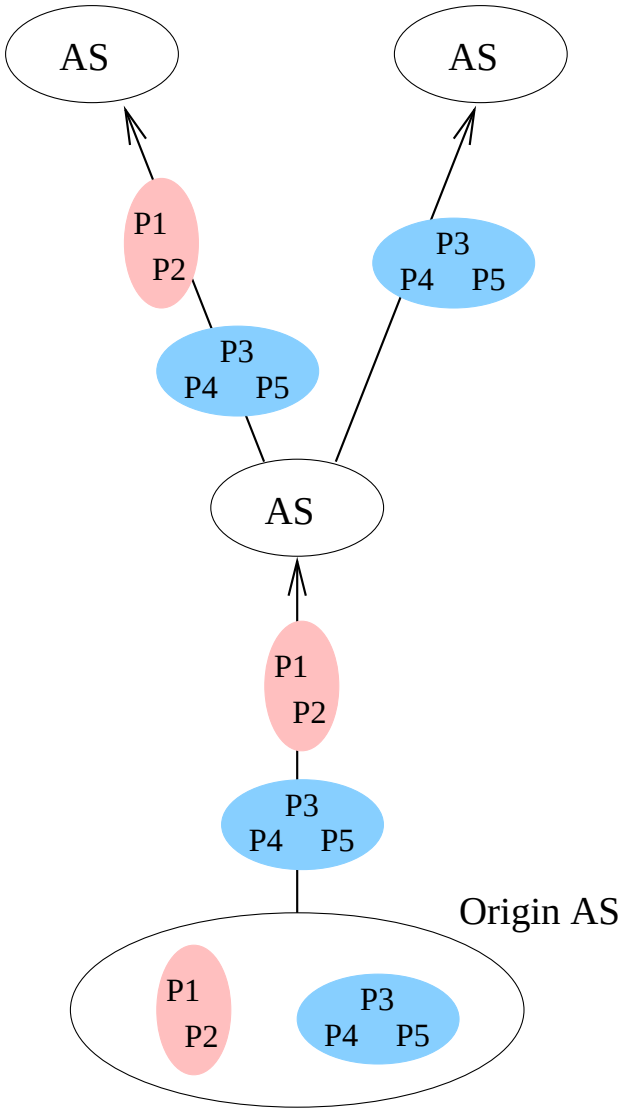
## Outline of talk

- Introduction
- Atoms architecture
- Incremental deployment
- Prototype
- Analysis and simulation
- Future work

## What is an atom?

- Group of *atomised prefixes* to be routed together
- To be *declared* by origin ASes
- These ASes partition prefixes into atoms and announce
- Atomised prefixes can be IPv4 or IPv6
- Only globally routed prefixes are atomised
  - Default-free zone (DFZ)
  - Not CIDR-aggregated into other prefixes

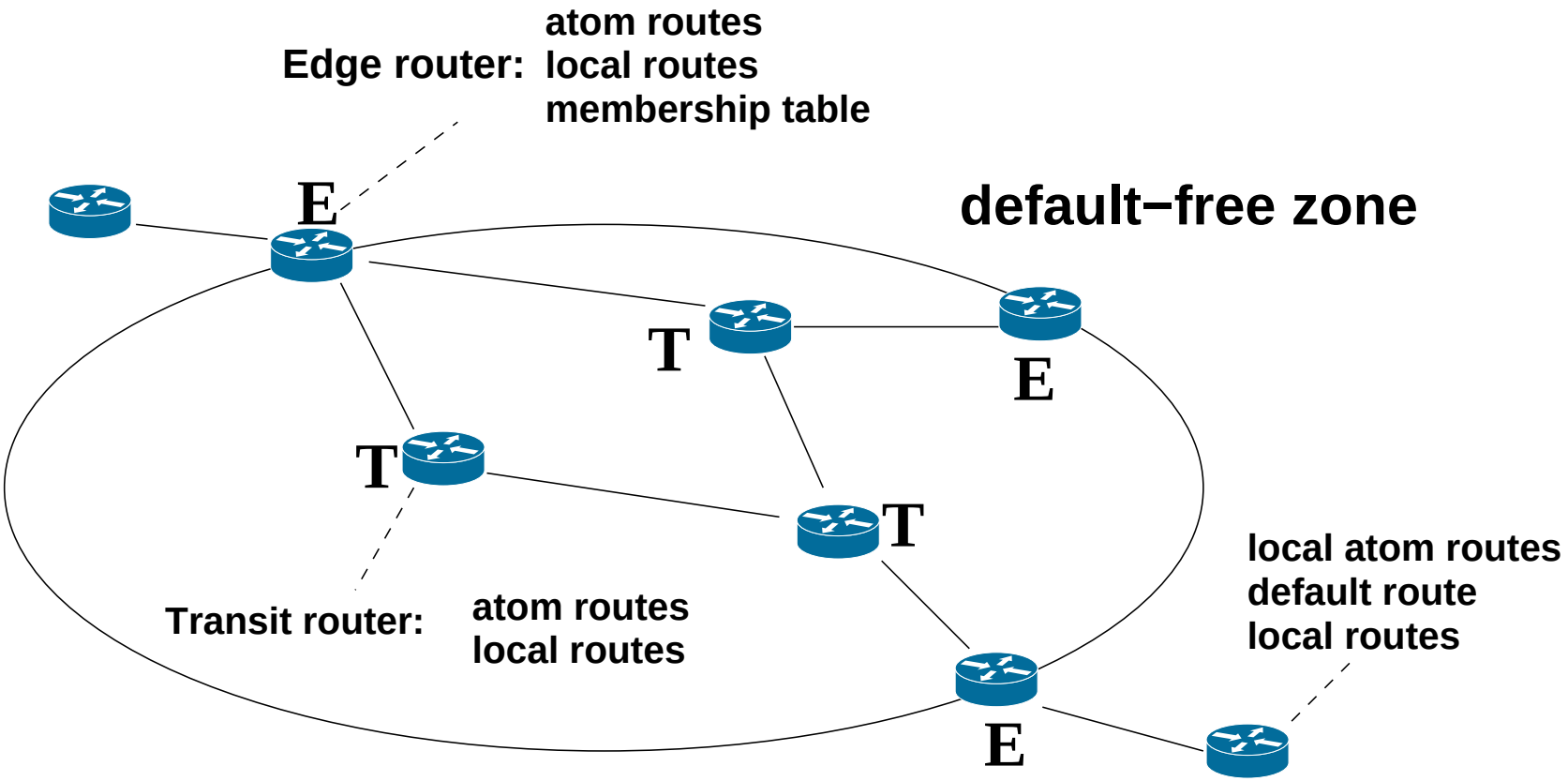
# What is an atom?



## Applying the atom concept

- BGP updates govern entire atoms
- Atoms replace prefixes in core routers:
  - Reduce table size in these routers
  - Reduce update load on these routers
- Maintain grouping of atoms outside of BGP
- Perhaps: improved convergence behaviour?

# Architecture — Overview



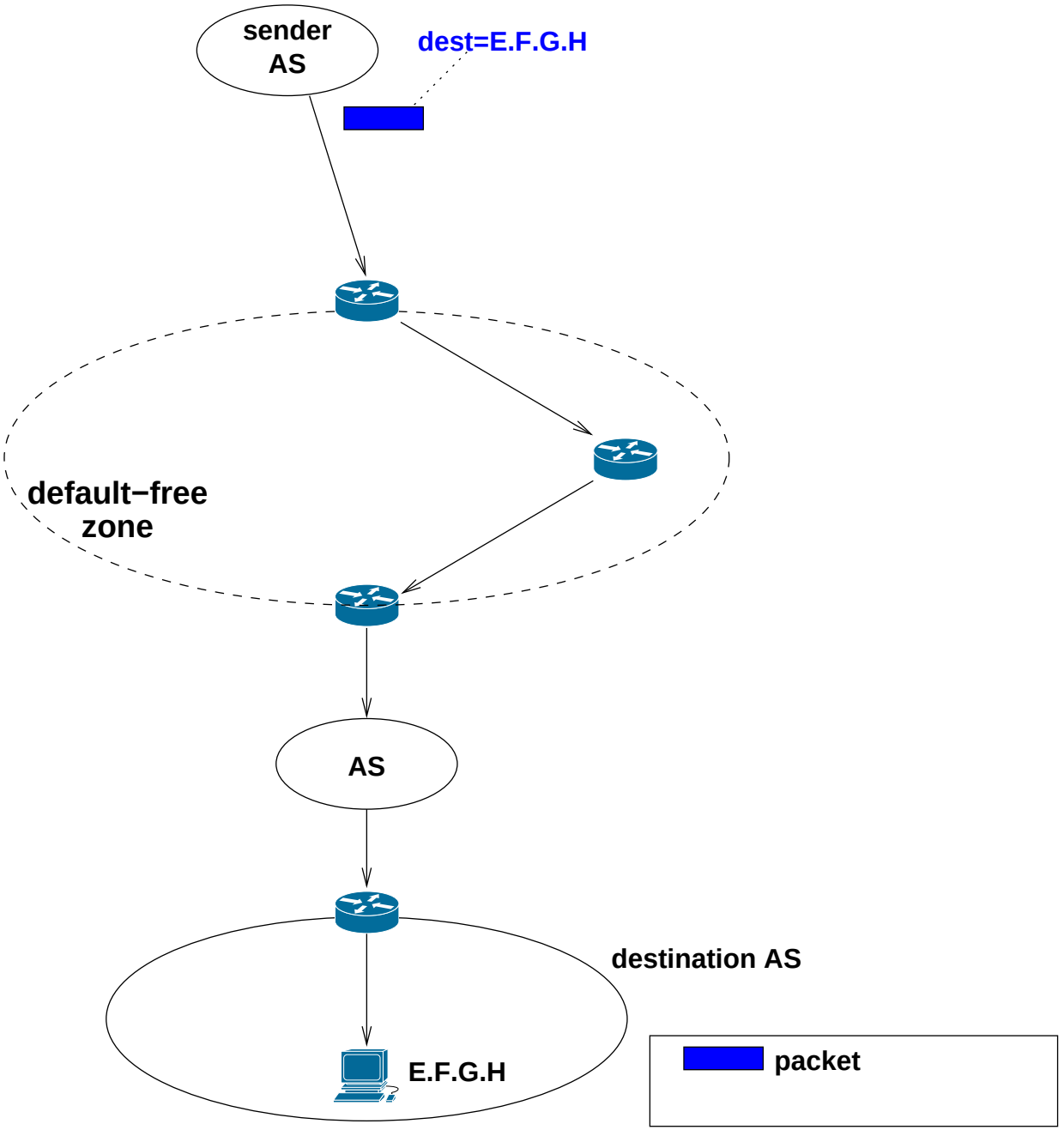
## Architecture — Components

- Tunneling / MPLS for forwarding
- BGP on atoms
- *Membership protocol* binds atomised prefixes to atoms

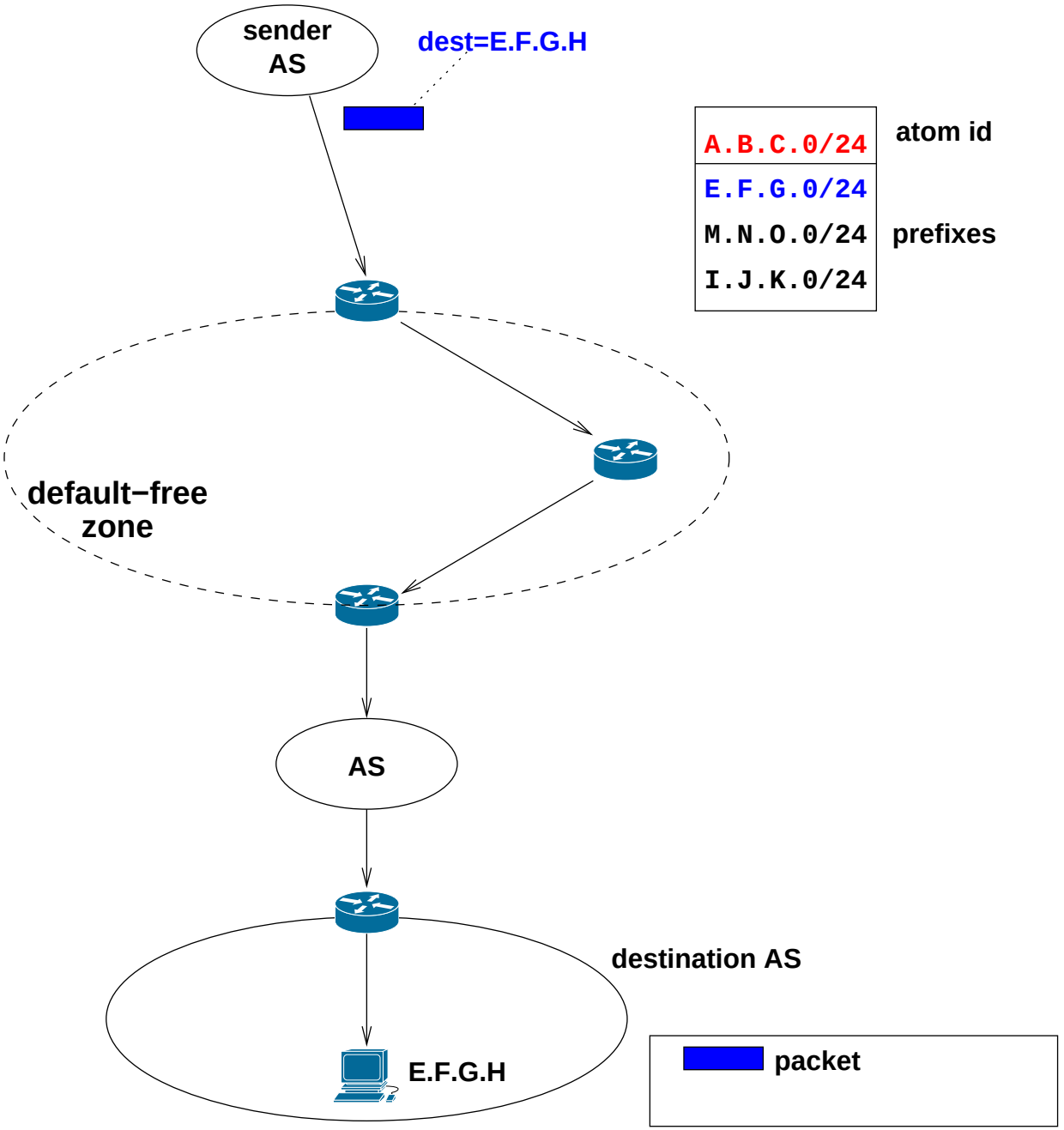
## Atom representation

- Atom is represented by an *atom ID*
- Atom ID syntactically a prefix (unrelated to prefixes in atom)
  - Reason: BGP can route atom IDs
- Atom IDs are a flat namespace
  - No further aggregation

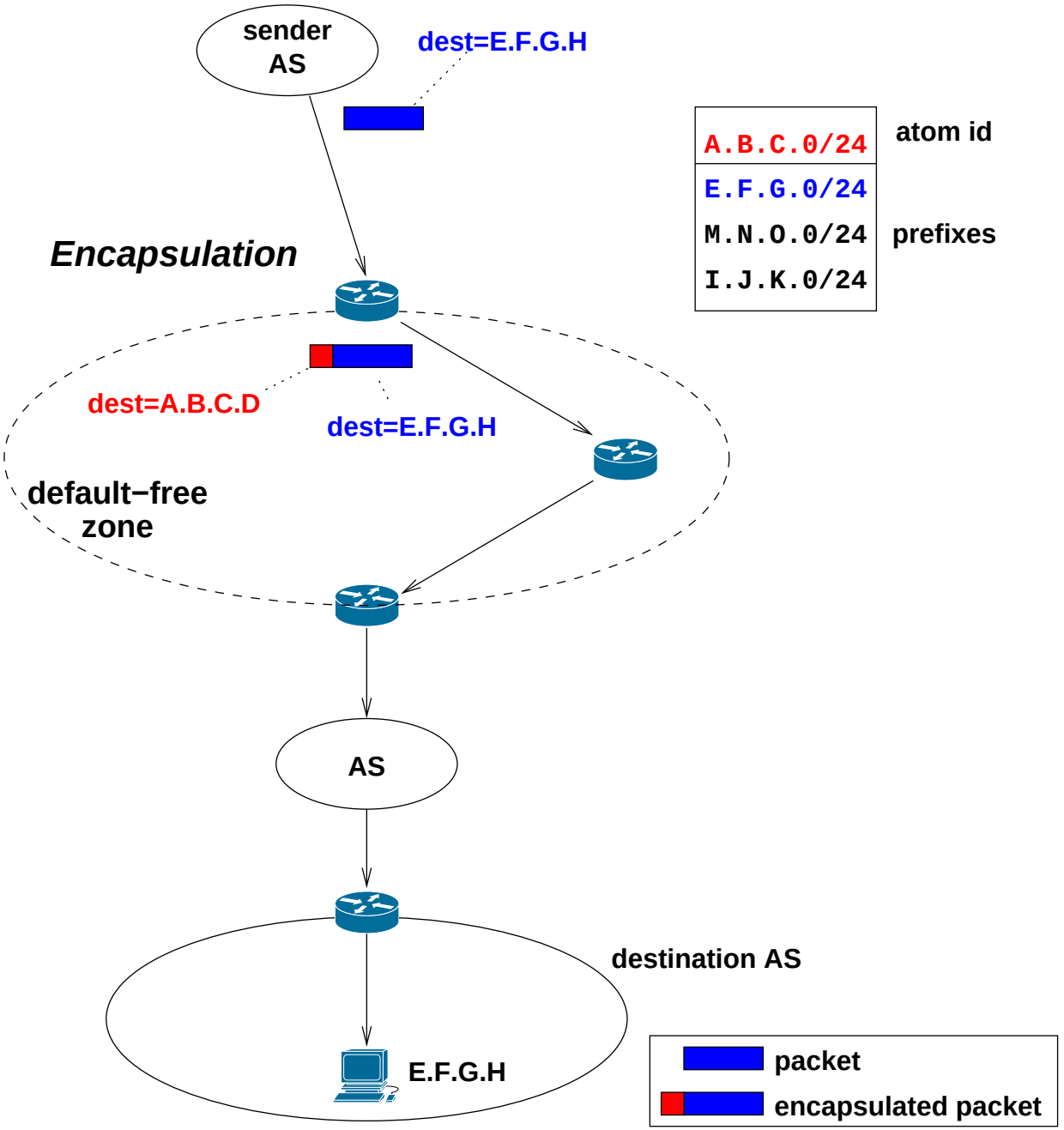
# Forwarding


sender
AS
dest=E.F.G.H
default−free
zone
AS
destination AS
E.F.G.H
packet

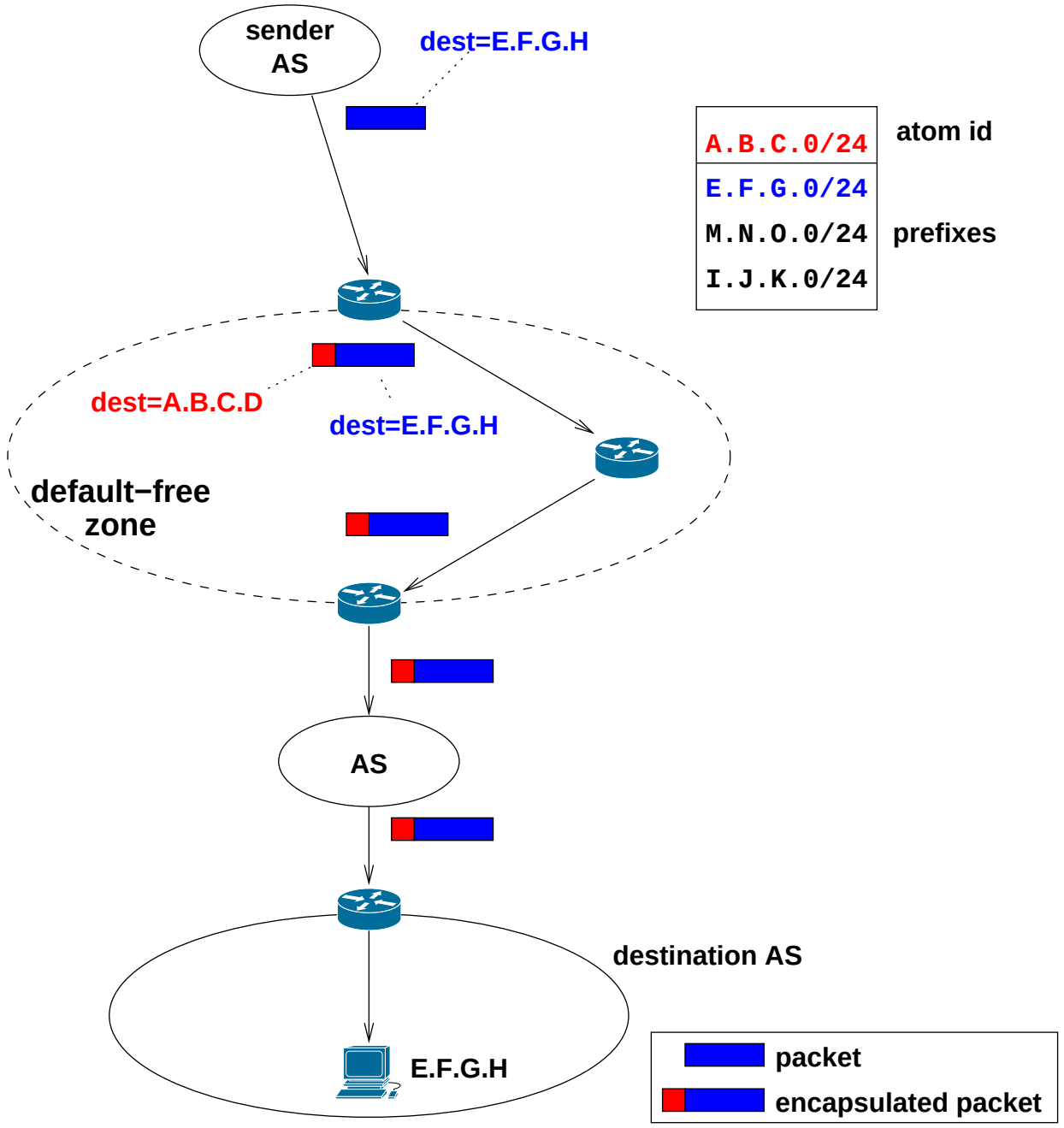
# Forwarding


sender
AS
dest=E.F.G.H
A.B.C.0/24
E.F.G.0/24
M.N.O.0/24
I.J.K.0/24
atom id
prefixes
default−free
zone
AS
destination AS
E.F.G.H
packet

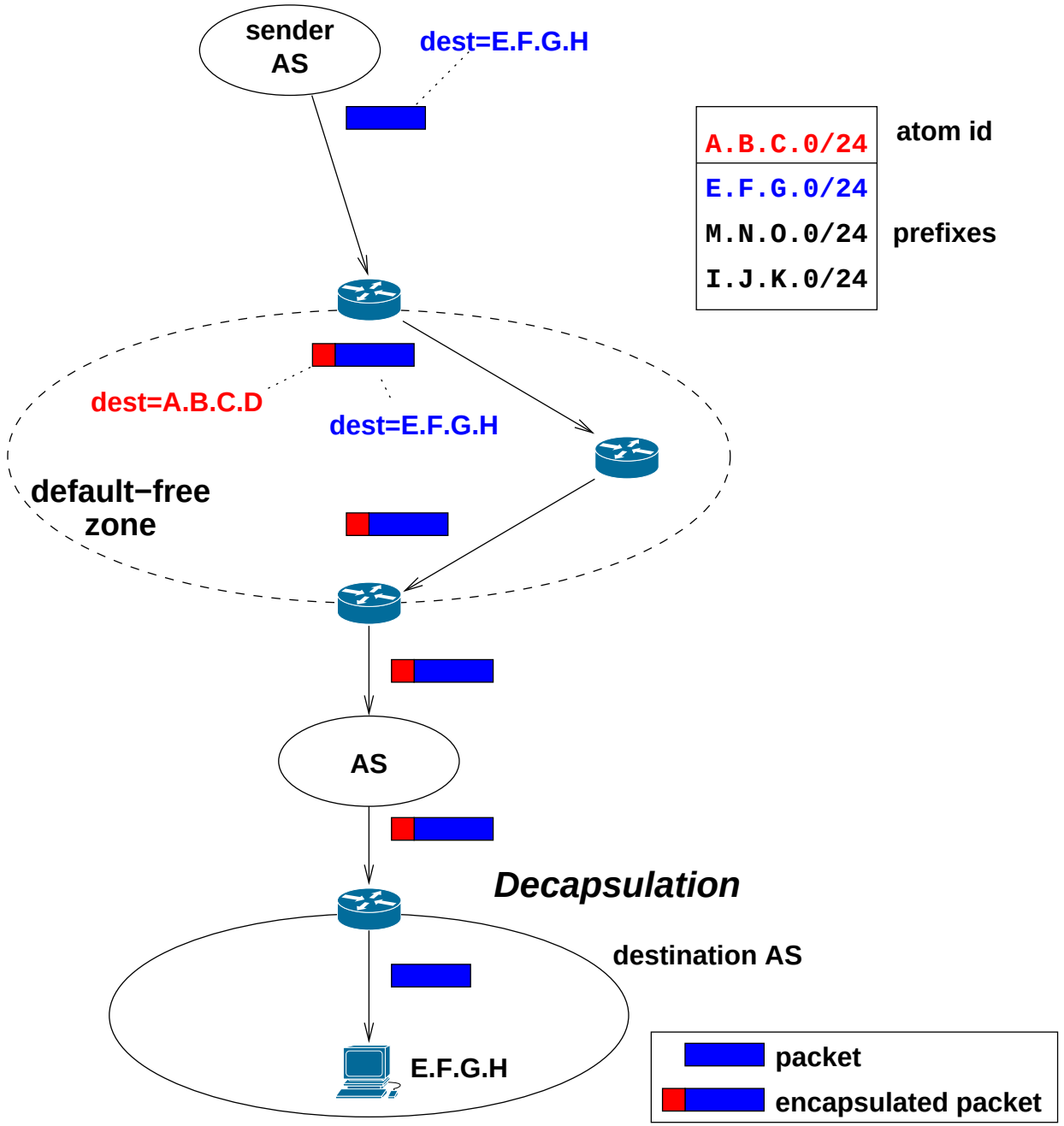
# Forwarding


sender AS
dest=E.F.G.H
A.B.C.0/24
E.F.G.0/24
M.N.O.0/24
I.J.K.0/24
atom id
prefixes
Encapsulation
dest=A.B.C.D
dest=E.F.G.H
default−free zone
AS
destination AS
E.F.G.H
packet
encapsulated packet

# Forwarding


sender AS
dest=E.F.G.H
A.B.C.0/24
E.F.G.0/24
M.N.O.0/24
I.J.K.0/24
atom id
prefixes
dest=A.B.C.D
dest=E.F.G.H
default−free zone
AS
destination AS
E.F.G.H
packet
encapsulated packet

# Forwarding


sender AS
dest=E.F.G.H
A.B.C.0/24
E.F.G.0/24
M.N.O.0/24
I.J.K.0/24
atom id
prefixes
dest=A.B.C.D
dest=E.F.G.H
default−free zone
AS
Decapsulation
destination AS
E.F.G.H
packet
encapsulated packet

## Forwarding

- Encapsulation to traverse DFZ
- Ingress edge router encapsulates
- Destination AS decapsulates

## Encapsulation vs MPLS

Similar techniques:

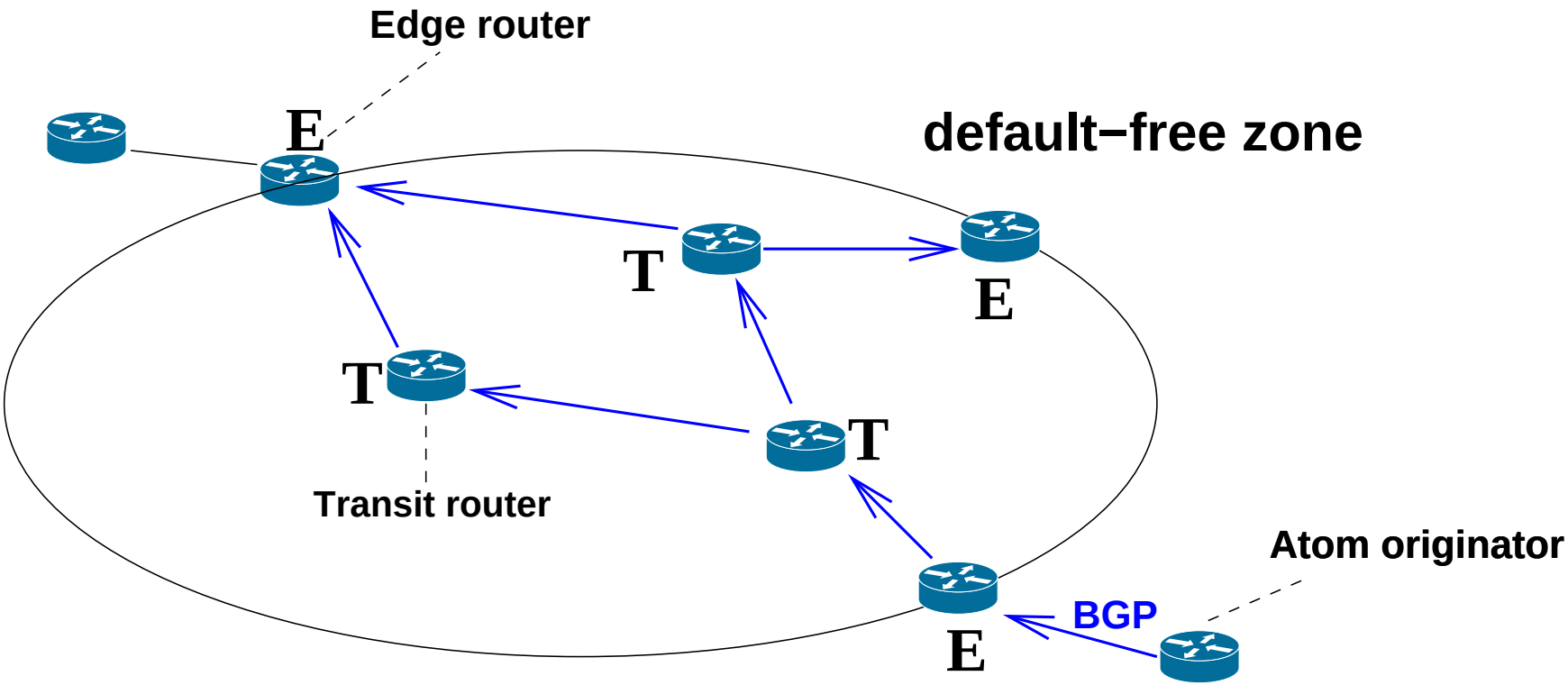
- Both do tunneling
- Atom ID $\leftrightarrow$ Forwarding Equivalence Class
- Encapsulation $\leftrightarrow$ Labeling
- IP forwarding $\leftrightarrow$ Label swapping

Disadvantages of encapsulation:

- Encapsulation reduces MTU
- Encapsulation complicates path MTU discovery and ICMP

*But:* MPLS not used interdomain

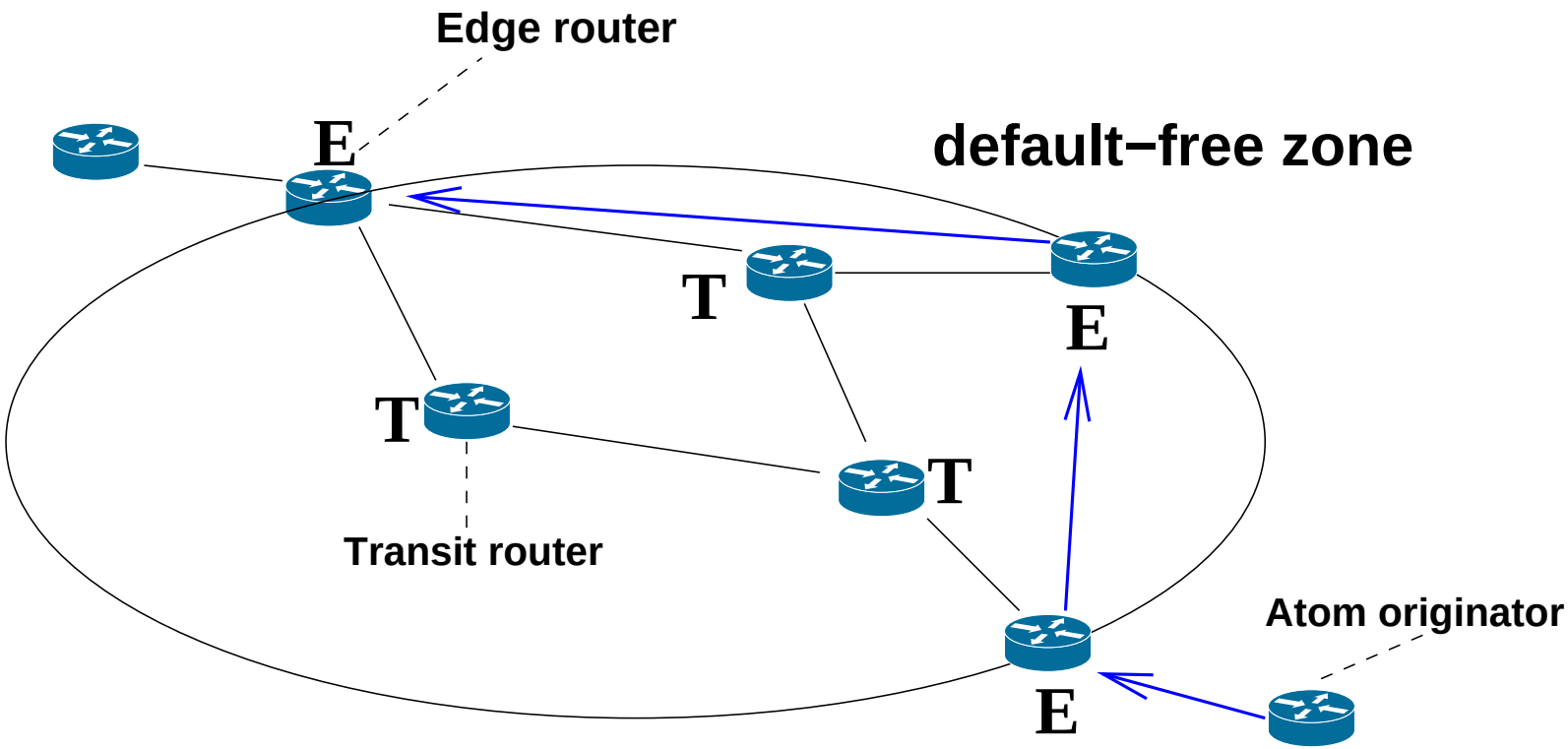
# BGP routes atom IDs



## Atom membership

- Atom originator partitions prefixes over atoms
- Sends $\{$ atom ID $\leftrightarrow$ atomised prefixes $\}$ to edge router(s)
- Edge routers propagate to neighbouring edge routers

# Atom membership



## Atom membership

Structured propagation limits unnecessary messages:

- Propagation based on current practices of business relationships:
  - Customers $\leftrightarrow$ all neighbours
  - Providers and peers $\nrightarrow$ providers and peers
- Additional rules avoid sending multiple times in most cases

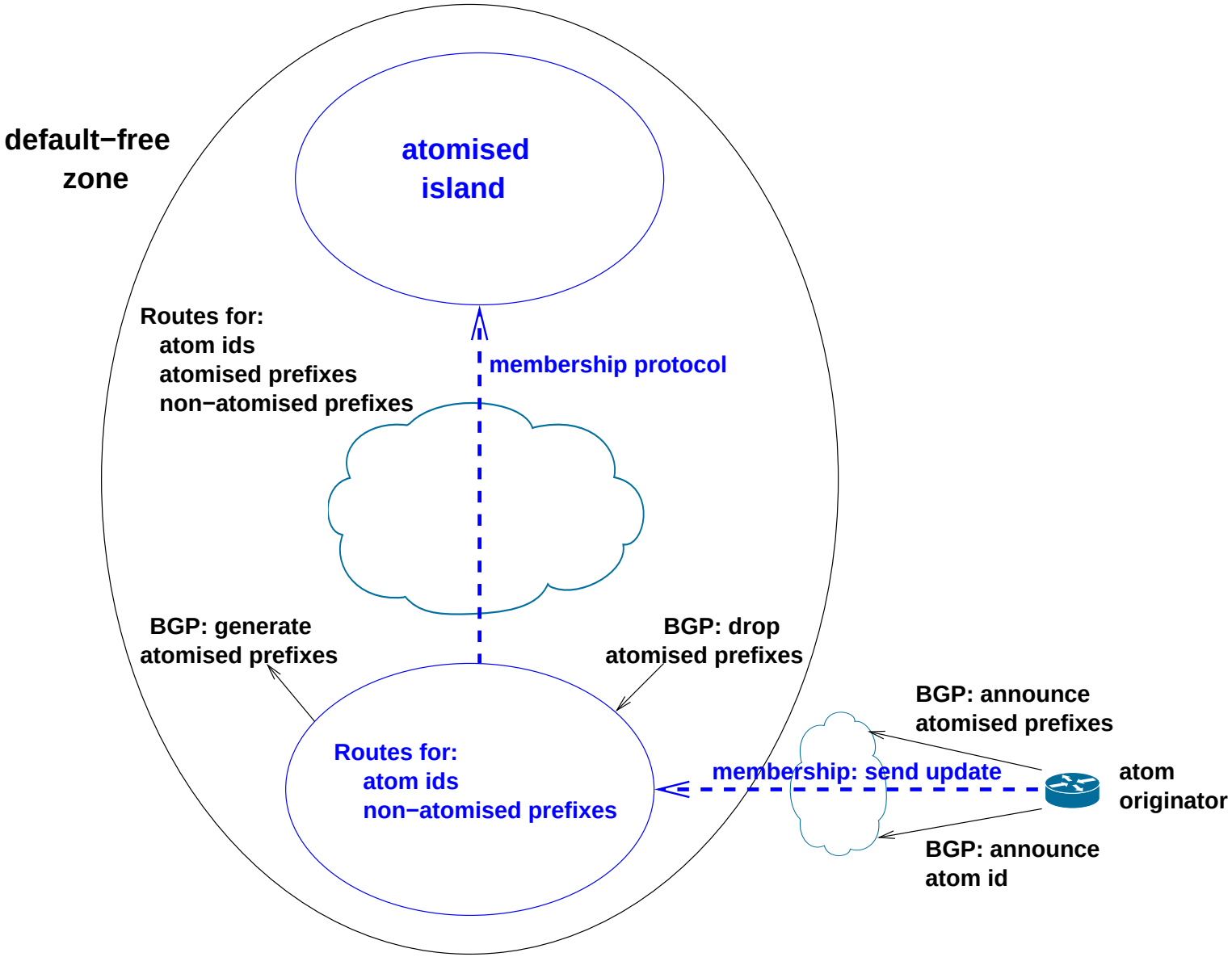
## Membership vs BGP

- Some differences between membership protocol and BGP. Membership protocol:
  - Maps atom ID $\leftrightarrow$ atomised prefixes
  - Has no route selection
  - Semantics of update independent of which neighbour sent it
  - Identical state in all edge routers after convergence
- Membership protocol is multihop:
  - Spoken only by edge routers and atom originators
  - Dynamics affect only subset of routers
  - Multihop but: session resets relatively harmless

## Outline of talk

- Introduction
- Atoms architecture
- Incremental deployment
- Prototype
- Analysis and simulation
- Future work

# Incremental deployment — Routing



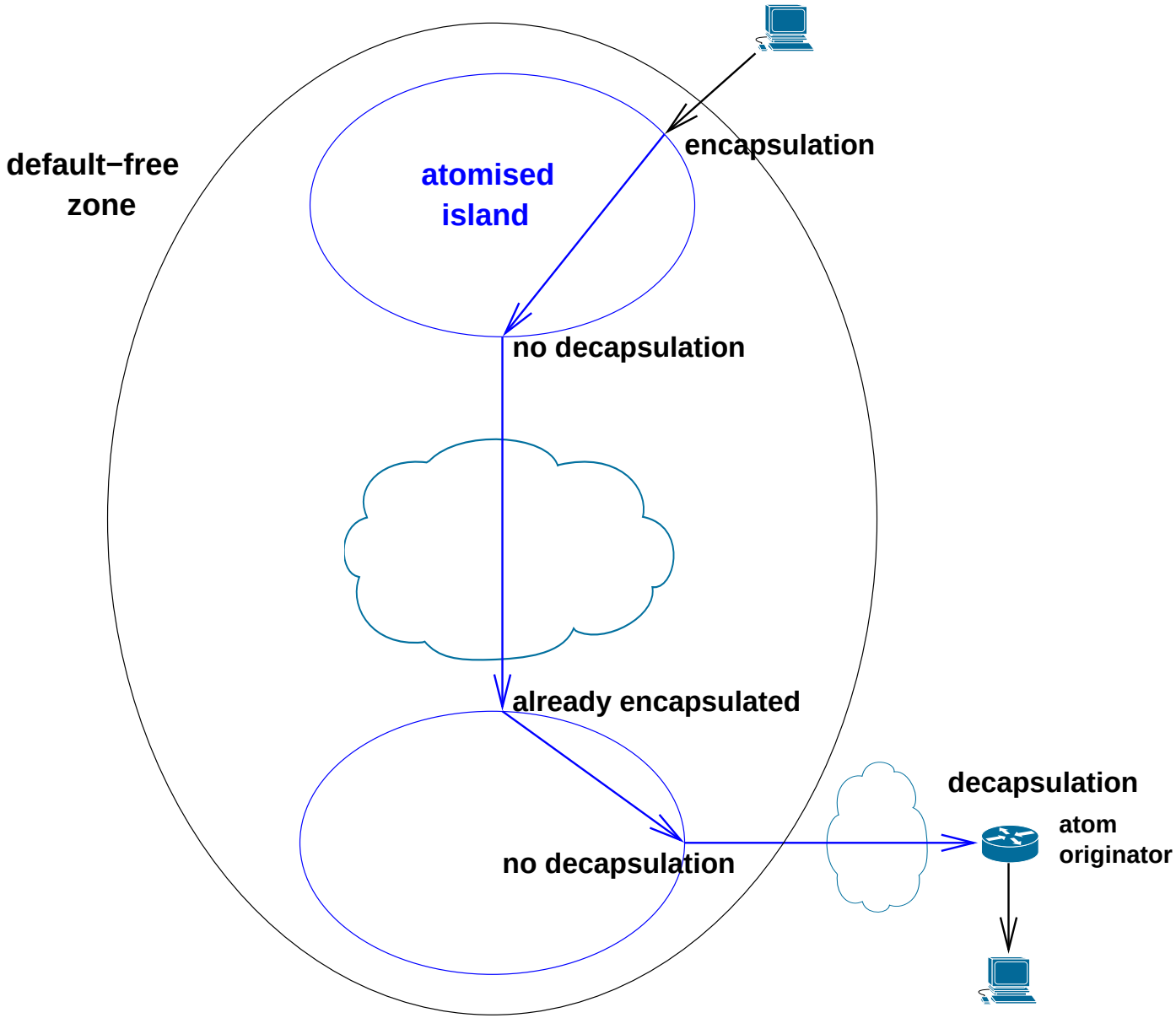
# Incremental deployment — Routing

- Non-atomised prefixes
  - All routers capable of routing non-atomised prefixes
- DFZ ASes that are not atomised
  - Multiple atomised *islands*
  - Islands connected through membership protocol (multihop)
  - Islands generate atomised prefix routes into non-atomised DFZ
  - ...and drop these routes when received from DFZ
- ASes outside DFZ that are not atomised
  - Atom originator speaks membership with edge router (multihop)
  - Announces route for atom ID in BGP
  - Announces routes for atomised prefixes in BGP (dropped by islands)

## Incremental deployment — Forwarding

- Encapsulate once: on first entry into an island
- Decapsulate once: at origin AS
- Routes for atom IDs exist on the entire forwarding path

# Incremental deployment — Forwarding



## Outline of talk

- Introduction
- Atoms architecture
- Incremental deployment
- Prototype
- Analysis and simulation
- Future work

## Prototype

- Implemented for IPv4 in Zebra (GNU license)
- Atoms *declared* manually in router configuration language

```
ip prefix-list A1 permit E.F.G.0/24
ip prefix-list A1 permit I.J.K.0/24
ip prefix-list A1 permit M.N.O.0/24
atom declare A.B.C.0/24 A1
```

- Zec's virtual network stack for testing [VIMAGE]
- It pings!

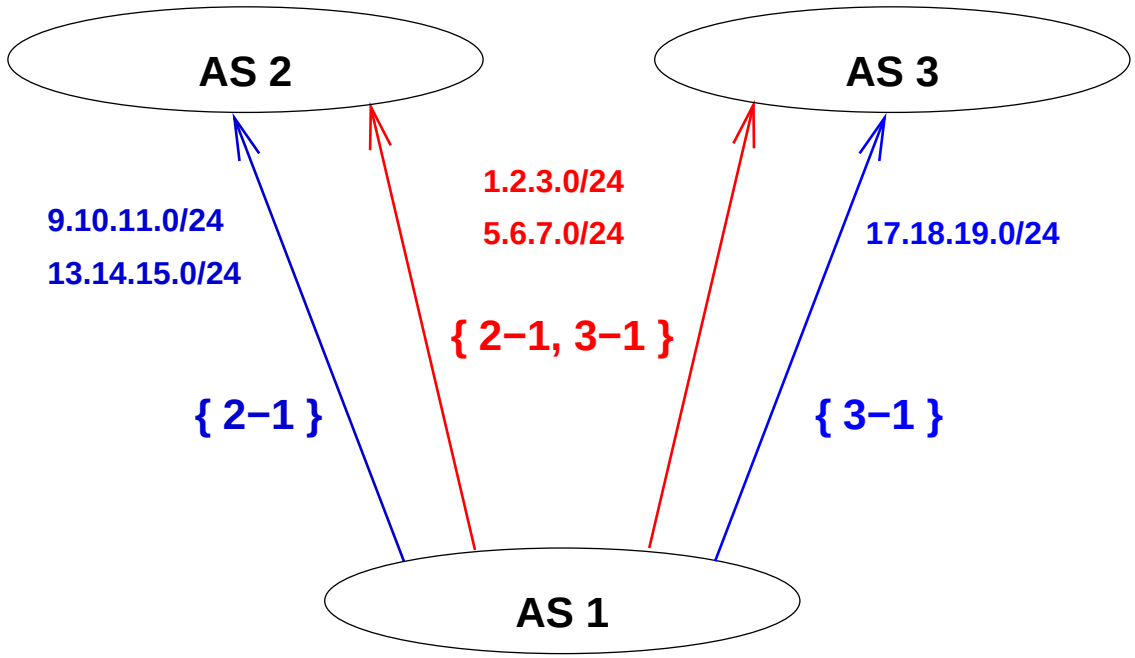
## Outline of talk

- Introduction
- Atoms architecture
- Incremental deployment
- Prototype
- Analysis and simulation
- Future work

# Analysis of origin link sets

Define: *origin link* == First two ASes in AS path

- Observe origin links sets of prefixes in RouteViews
- Estimate number of atoms as number of origin link sets
- RouteViews Nov 1 snapshot:
  - 24K unique origin link sets
  - covering 127K prefixes
  - 16K ASes



## Simulation (in progress)

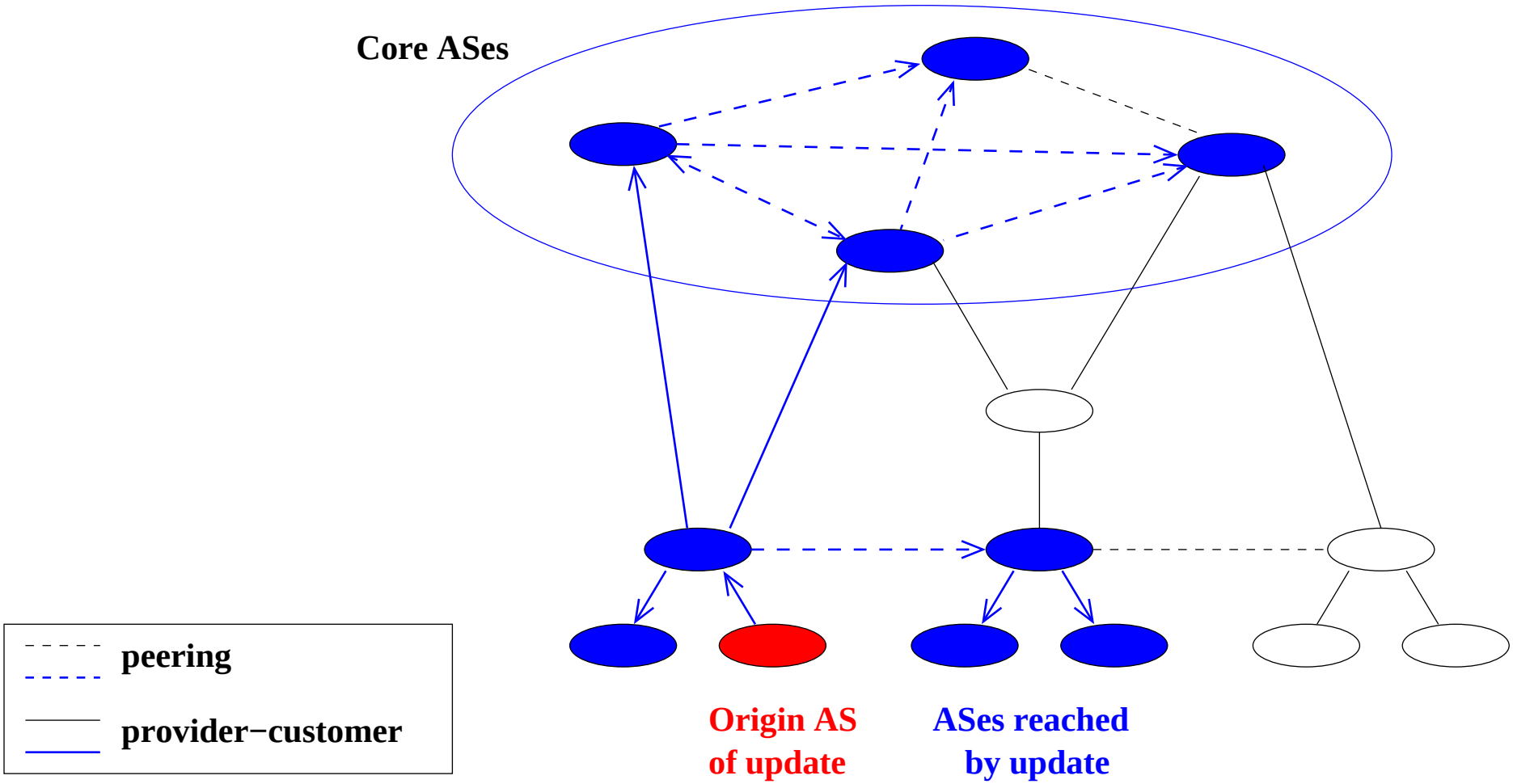
GeorgiaTech's BGP extension to ns-2 [BGP++]

- Simulate various kinds of updates:
  - BGP updates
  - Membership updates
- Determine cost of these updates:
  - Number of messages
  - Convergence time
- Compare with non-atoms routing
- Analyse ratio of membership updates to BGP updates
  - #membership $>$ #BGP but same order of magnitude

# Simulation topology

- Subset of RouteViews AS topology
- Business relationships from [VantagePoints]
- Method:
  - Start with an origin AS
  - Include any ASes potentially reached by updates from that AS
    - Update from customer AS reaches all neighbour ASes
    - Update from any neighbour AS reaches all customer ASes
    - Update providers or peers do not reach other providers or peers
  - To limit size, don’t traverse core
- Add fixed number of routers to each AS
- Form DFZ from core and multihomed ASes

# Selecting subset of AS topology



# Future work

- Configurability
  - Complexity
  - Multiple atom originators per AS
- Security: SBGP or soBGP
- Provider-originated atoms:
  - Immediate providers of stub ASes collectively originate atoms from multiple customers
  - May require only limited cooperation from providers
  - Reduces lower bound on number of atoms from 24K to 12K
- IPv6
- Interdomain MPLS
- Measure overhead of encapsulation/decapsulation

# Questions?

*Acknowledgements*

Andrew Lange
Andrew Partan
Bill Woodcock
Bradley Huffaker
CAIDA folks
Cengiz Alaettinoglu
Daniel Karrenberg
Dave Meyer
Evi Nemeth
Fontas Dimitropoulos
Frances Brazier
Frank Kastenholz
Geoff Huston
Henk Uijterwaal
Jeffrey Haas
Maarten van Steen
Nevil Brownlee
Mike Lloyd
Omer Ben-Shalom
Pedro Roque Marques
Ronald van der Pol
Ruomei Gao
Sean Finn
Senthilkumar Ayyasamy
Ted Lindgreen
Teus Hagen
Vijay Gill
Wytze van der Raay

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

## References


[BGP-GROWTH] T. Bu, L. Gao and D. Towsley, ”On characterizing BGP routing table growth,” Proc. IEEE Global Telecommunications Conf. (GLOBECOMM), pp. 2197-2201, Nov. 2002

[BGP++] Christos Xenofontas Dimitropoulos, ”BGP implementation for ns-2,” http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/

[RFC2772] R. Rockell and R. Fink, ”6Bone Backbone Routing Guidelines, RFC 2772,” Feb. 2000

## References


[VantagePoints] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz, ”Characterizing the Internet Hierarchy from Multiple Vantage Points,” in Proc. of IEEE INFOCOM 2002, New York, NY, Jun 2002

[VIMAGE] Marko Zec, ”Network stack cloning / virtualization extensions to the FreeBSD kernel,” http://www.tel.fer.hr/zec/vimage/

# Message layout

**BGP: announce atom**

| | |
|---|---|
| Withdraw | |
| Announce | `A.B.C.0/24` |
| Attributes | `••••••••••` |

*atom id*

**BGP: withdraw atom**

| | |
|---|---|
| Withdraw | `A.B.C.0/24` |
| Announce | |
| Attributes | |

*atom id*

**Atom membership update**

| | |
|---|---|
| Withdraw | |
| Announce | `A.B.C.0/24` |
| Membership attribute | `E.F.G.0/24`<br>`I.J.K.0/24`<br>`M.N.O.0/24`<br><br>`7382576` |

*atom id*

*atomised prefixes*

*timestamp*

# Analysis of origin link sets

Infer updates to atoms by origin ASes, time-out to allow for BGP noise

- Routing change: all prefixes of S1 move to new S2
- Split: some prefixes in S1 move to new S2
- Join: all prefixes in S1 move to existing S2
- Shift: some prefixes in S1 move to existing S2
- Announcement: newly routed prefixes form a new S1
- Announce membership: newly routed prefixes join existing S1
- Withdrawal: all prefixes in S1 become unreachable
- Withdraw membership: some prefixes in S1 become unreachable

## Analysis of origin link sets

Each such update:

- Zero or more BGP updates
- Zero or more membership updates

# Analysis of origin link sets