SWINBURNE UNIVERSITY OF TECHNOLOGY

# A Technique for Reducing BGP Update Announcements through Path Exploration Damping

Geoff Huston, Mattia Rossi, Grenville Armitage

mrossi@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

## Terminology and BGP recap

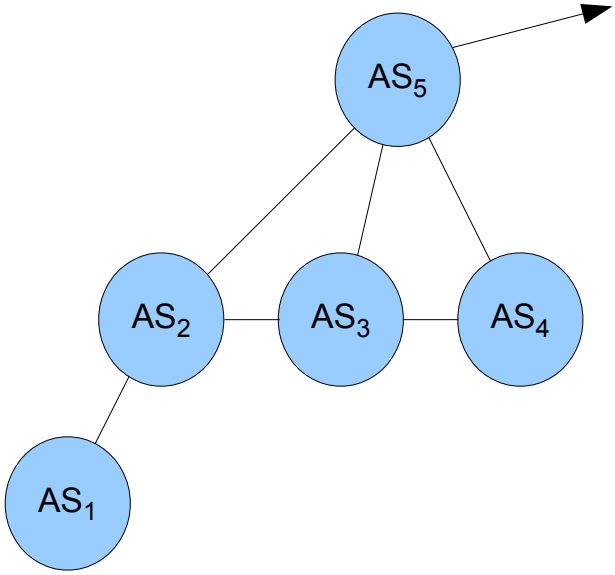
- BGP messages between BGP speakers (peers):
  - OPEN
  - KEEP-ALIVE
  - UPDATE
  - NOTIFICATION
  - ROUTE-REFRESH
- UPDATE messages:
  - carry announcements or withdrawals or both for the same AS-path
  - announcements or withdrawals are a list of prefixes
  - “update packing”
- AS-path:
  - list of ASes to be traversed to reach the originator/owner of the prefix
  - AS-path length is the main criteria for deciding the best path

# Terminology and BGP recap

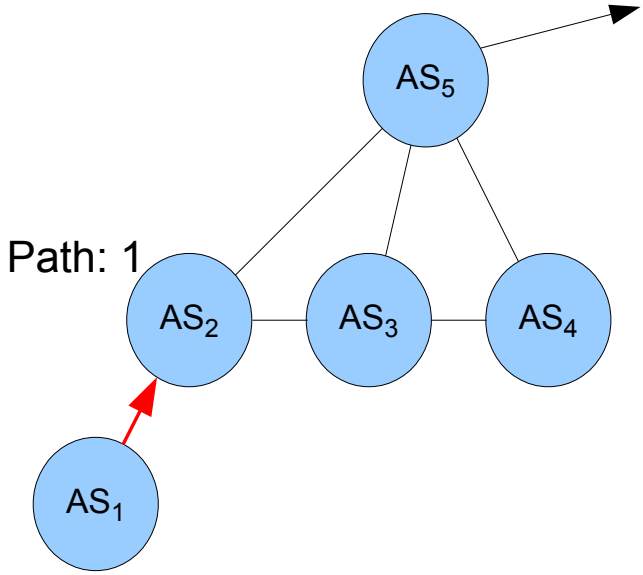
- BGP speaker uses 3+1 Tables:
  - ADJ-RIB-IN (per peer)
  - RIB (routing table)
  - ADJ-RIB-OUT (per peer)
  - decision making takes place in RIB using information from ADJ-RIB-IN
  - ADJ-RIB-OUT used for UPDATEs per peer
  - FIB, forwarding table
- 2 Layers:
  - Control plane (routing plane) – exchange of routing information
  - Data plane (forwarding plane) – packet forwarding



# BGP dynamics

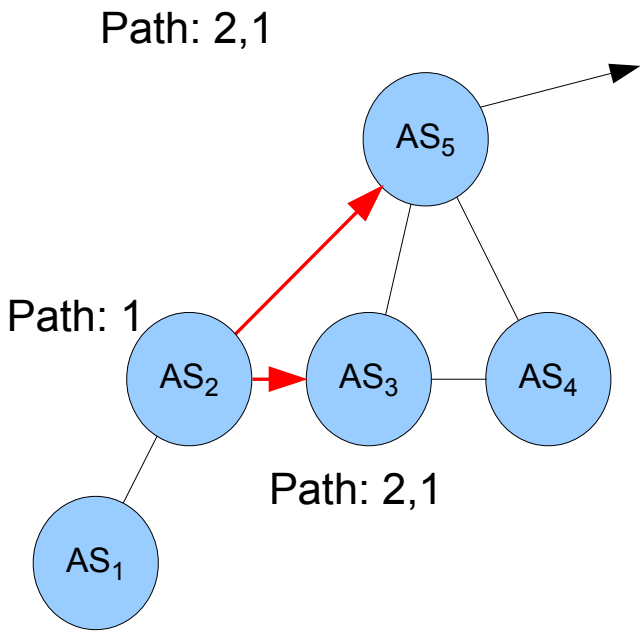


# BGP dynamics

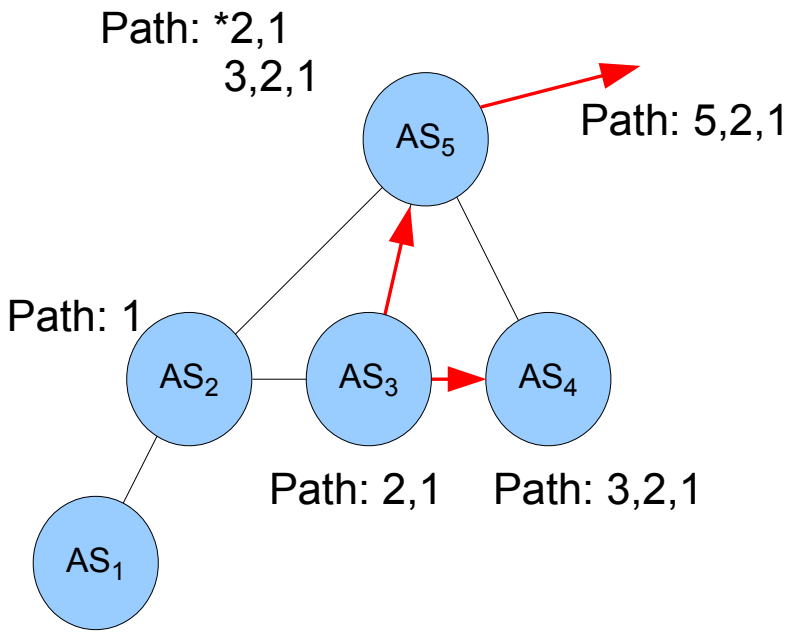





# BGP dynamics

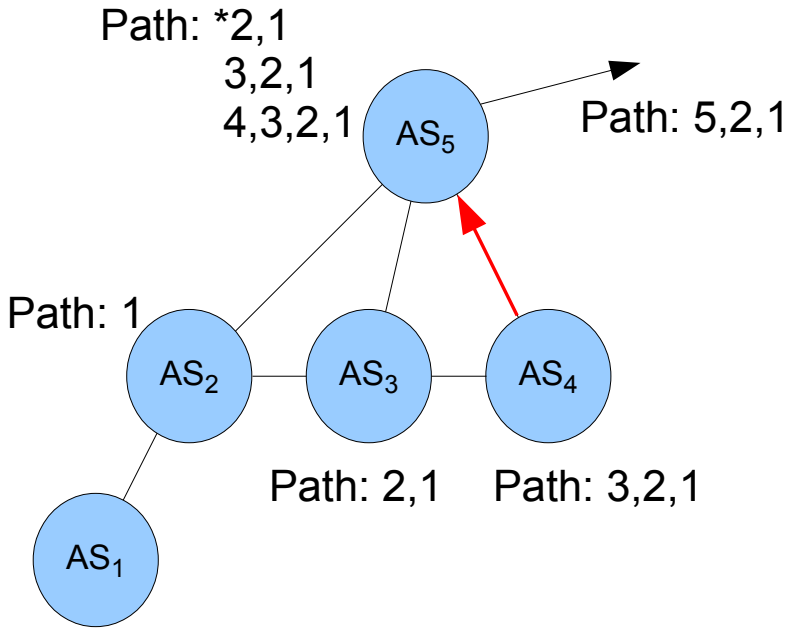


# BGP dynamics

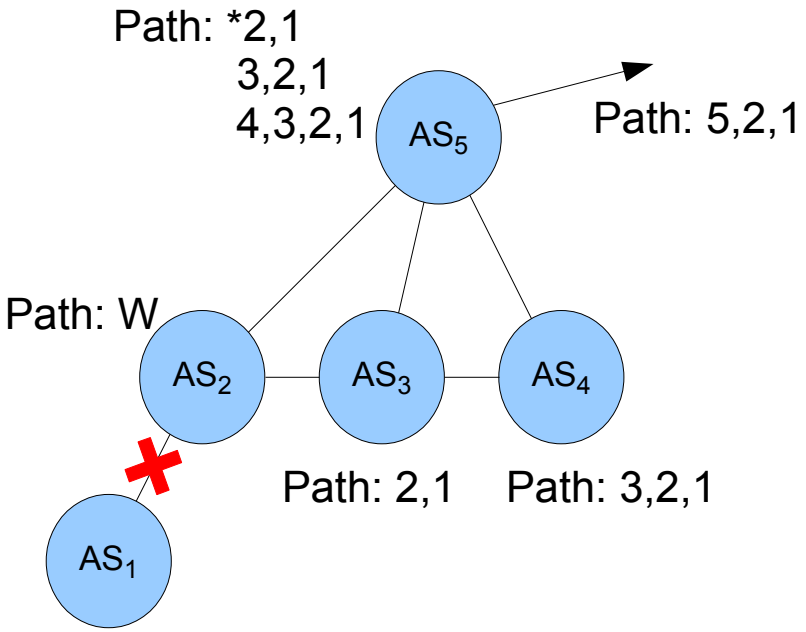





# BGP dynamics

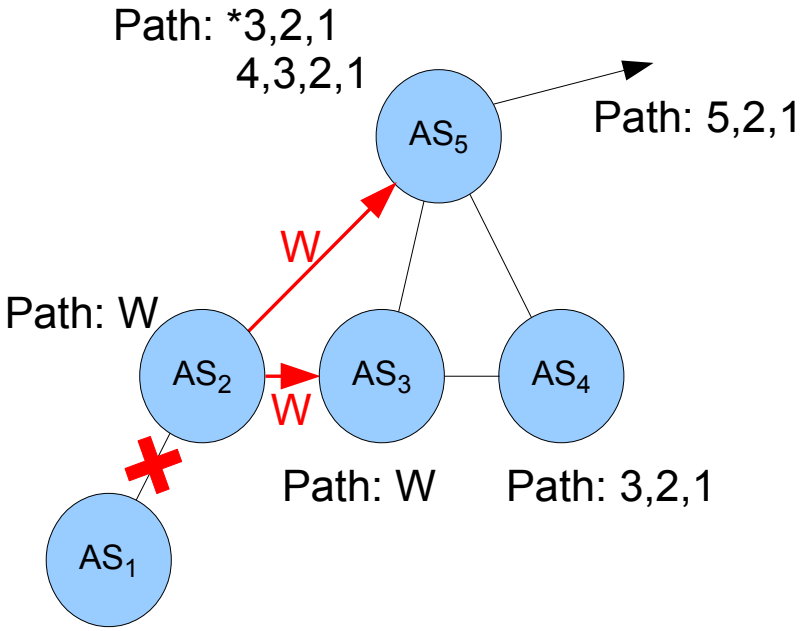


# What is Path Exploration?

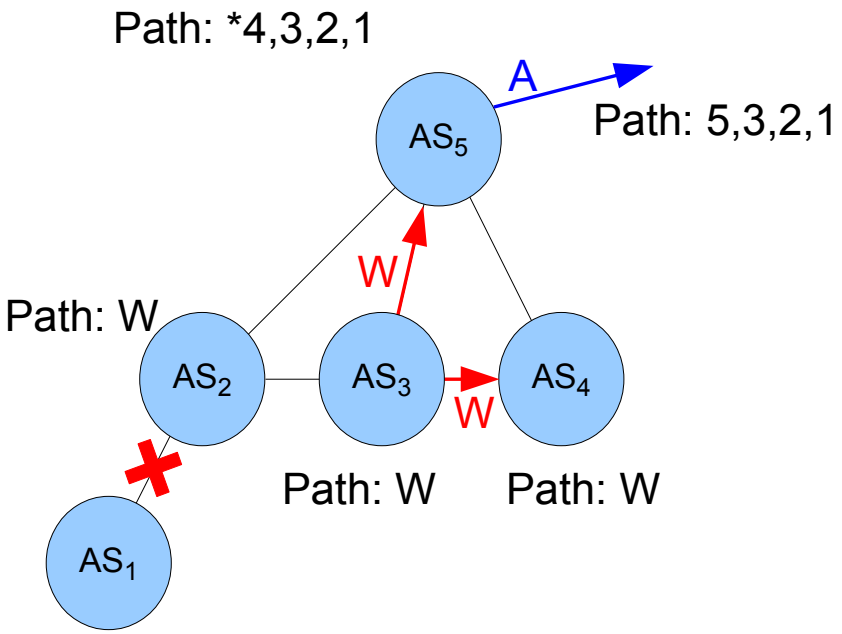





# What is Path Exploration?

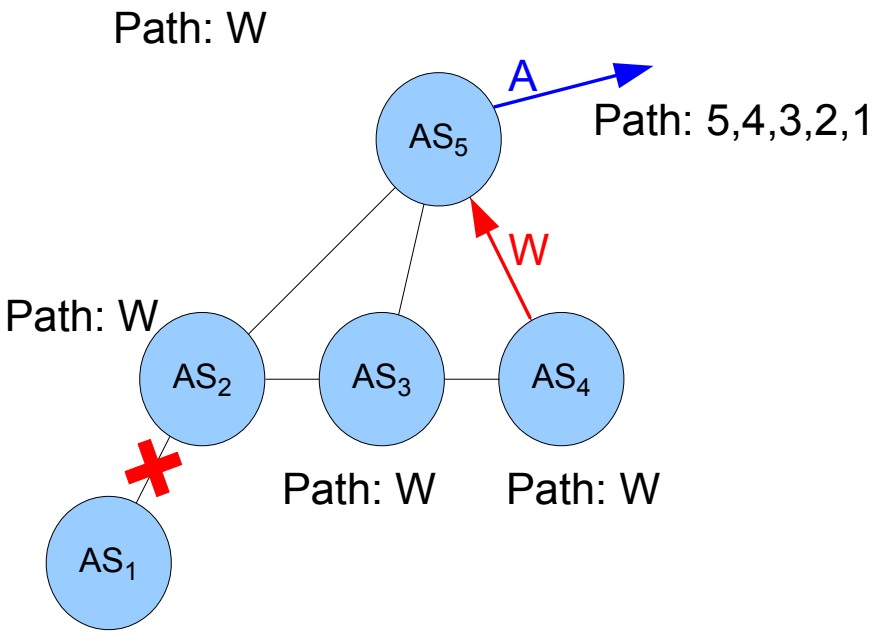


# What is Path Exploration?

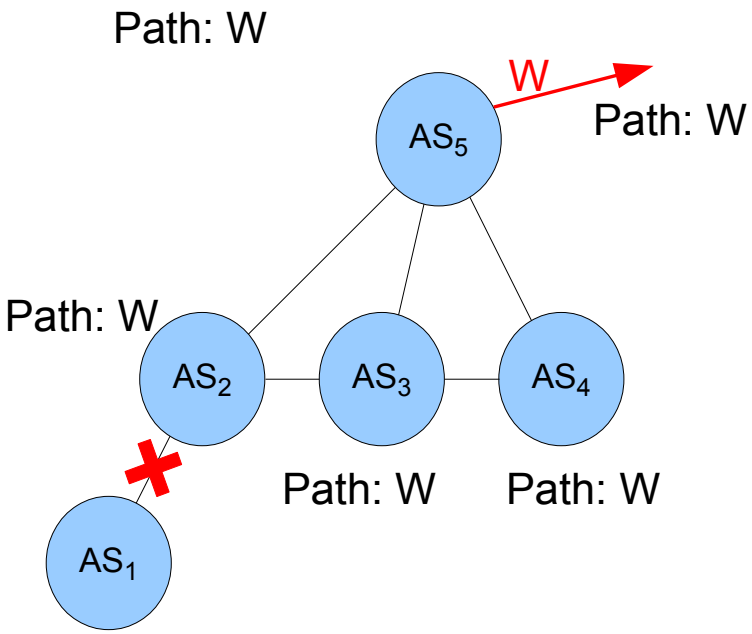





# What is Path Exploration?



# What is Path Exploration?






# Path Exploration consequences

Unnecessary load!

- Every update triggers one or more decision processes ⇒ ↑CPU load
- Every peer sends updates
- ↑peers ⇒ ↑CPU load
- Path Exploration ≡ unnecessary updates ≡ unnecessary CPU load
- Unnecessary network load

# Current methods to avoid unnecessary updates

- Old knowledge - countermeasures exist
- Outgoing updates depend on arrival time of incoming updates
- Three methods available in current BGP systems
- All three based on timing:
  - MRAI - Minimum Route advertisement Interval
    - Delays announcements
    - Goal: Prevention of interim announcements
  - RFD - Route Flap Damping
    - Delays withdrawals and announcements for unstable peers for 24 hrs
    - Goal: Prevention of updates caused by flapping routes
  - WRATE - Withdrawal rate limiting
    - Delays withdrawals and announcements
    - Goal: Allow peer to stabilize before sending updates



# The Problem

- Current status is not ideal:
- RFD is strongly discouraged
- WRATE likely disrupts data delivery
- MRAI widely deployed with a timer of 30 seconds + random jitter for eBGP
- often not deployed on a per-prefix basis → random behavior
- We need a better solution!

# Update categorization

| Code | Description |
|---|---|
| AA+ | Announcement of an already announced prefix with a longer AS Path (update to longer path) |
| AA- | Announcement of an announced prefix with a shorter AS Path (update to shorter path) |
| AA0 | Announcement of an announced prefix with a different path of the same length (update to a different AS Path of same length) |
| AA* | Announcement of an announced prefix with the same path but different attributes (update of attributes) |
| AA | Announcement of an announced prefix with no change in path or attributes (possible BGP error or data collection error) |
| AW | Withdrawal of an announced prefix |
| NA | Announcement of a previously unknown or withdrawn prefix |



# Update sequences

- Path Exploration example by update sequence:
  - {NA, AA+, AA+, AW}
- We define Path Exploration:
  - An update sequence lengthening the AS-path gradually until stability is reached

# PED – Getting rid of unnecessary updates the smart way

- Path Exploration Damping – PED Algorithm:
  - Delay updates announcing a longer (or equal-length) AS-path (AA+,AA0,AA*,AA)
  - Immediately send announcements of a shorter AS-path or withdrawals (AA-,AW)
  - Do not delay initial announcements (NA)
- Perform output queue compression
- Introduction of the Path Exploration Damping Timer – PEDI
- PED does not change the BGP protocol
- PED can be deployed incrementally

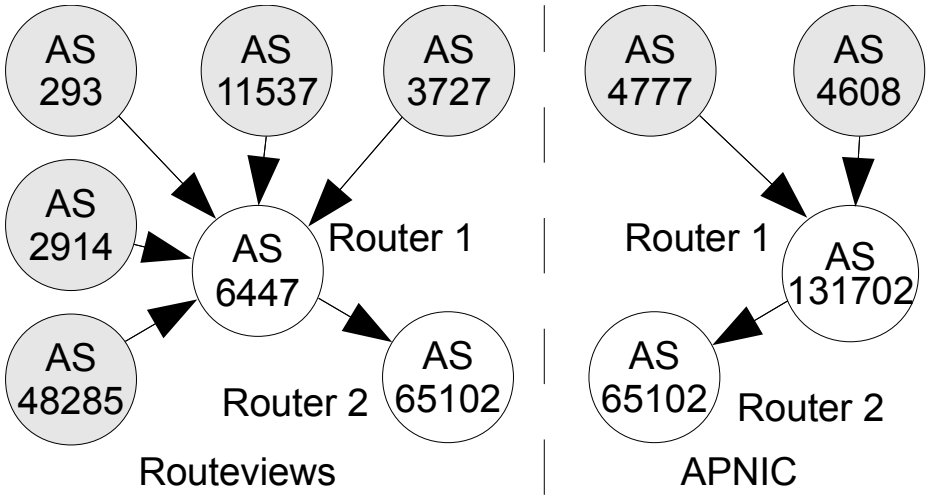


# PED – Implementation

- Quagga 0.99.13
- Implemented in ADJ-RIB-OUT (does not affect decision process)
- Per-peer and per-prefix basis
- Added jitter like MRAI does

# PED – Reduction in update load

- Experiments using 24 hours of real BGP updates
- Two datasets:
  1. APNIC (2 peers)
  2. Routeviews (5 peers)
- Replayed using the *Quagga-Accelerator*

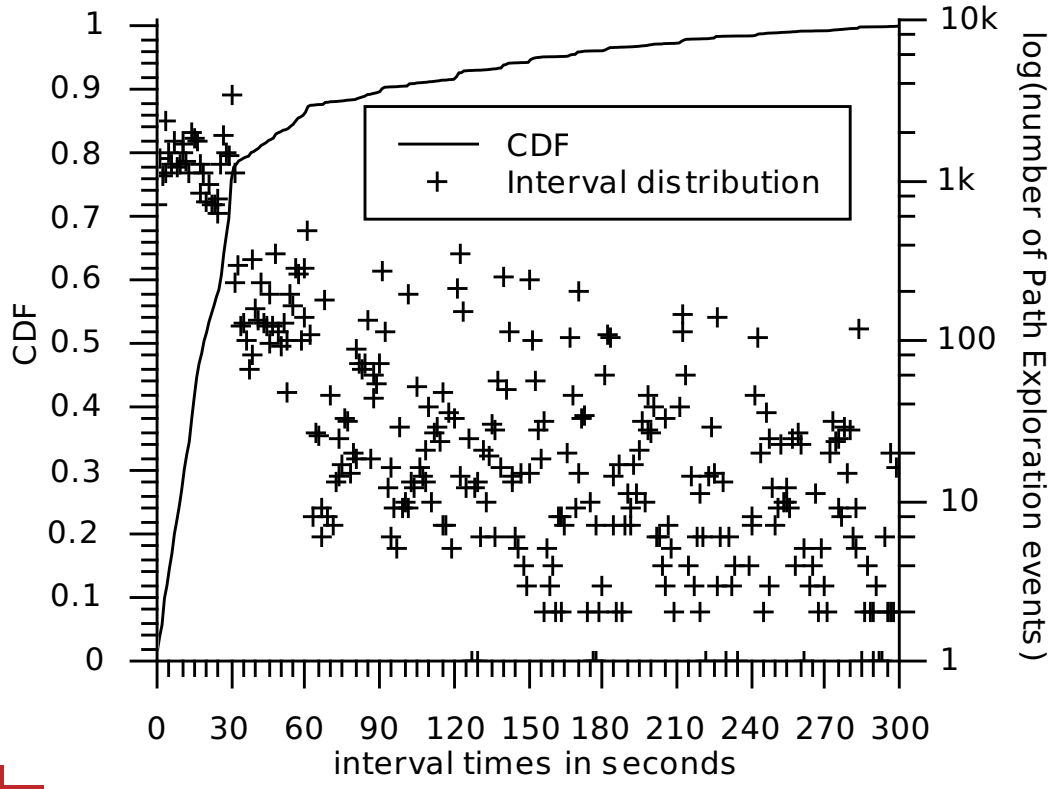





# PED – Reduction in update load

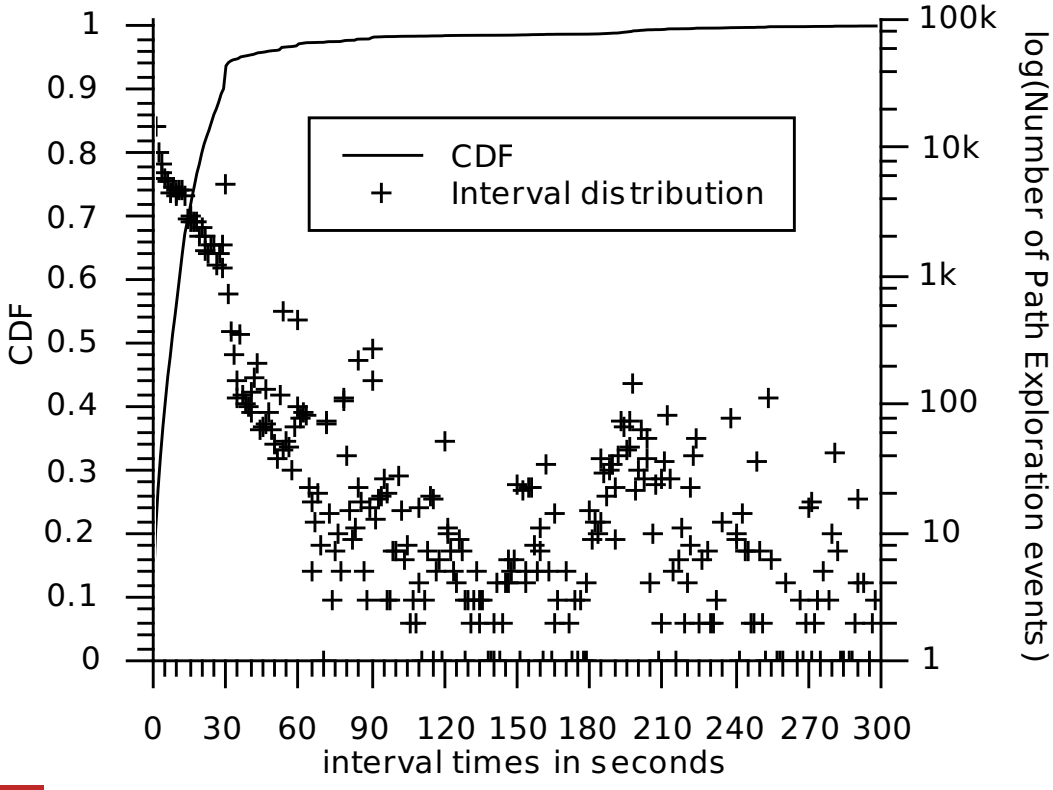
- Used a range of PEDI settings based on intervals of incoming Path Exploration sequences:
  1. APNIC: 30s – 70s, 5s steps
  2. Routeviews: 5s – 75s, 5s steps
- Compared to 0s MRAI (no delay – Juniper default) and 30s MRAI (Cisco default)

# Incoming Path Exploration intervals – APNIC

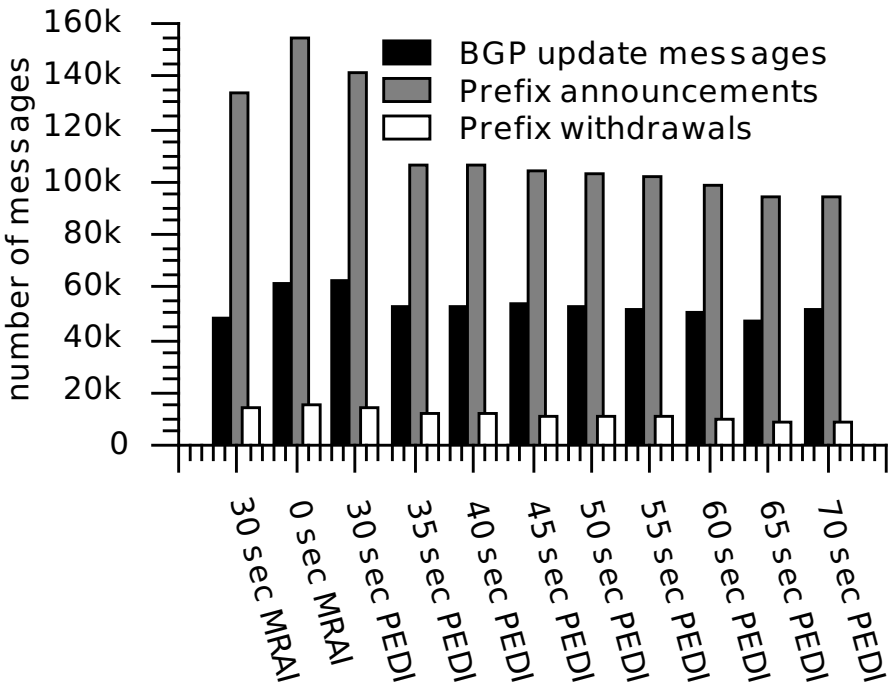





# Incoming Path Exploration intervals – Routeviews

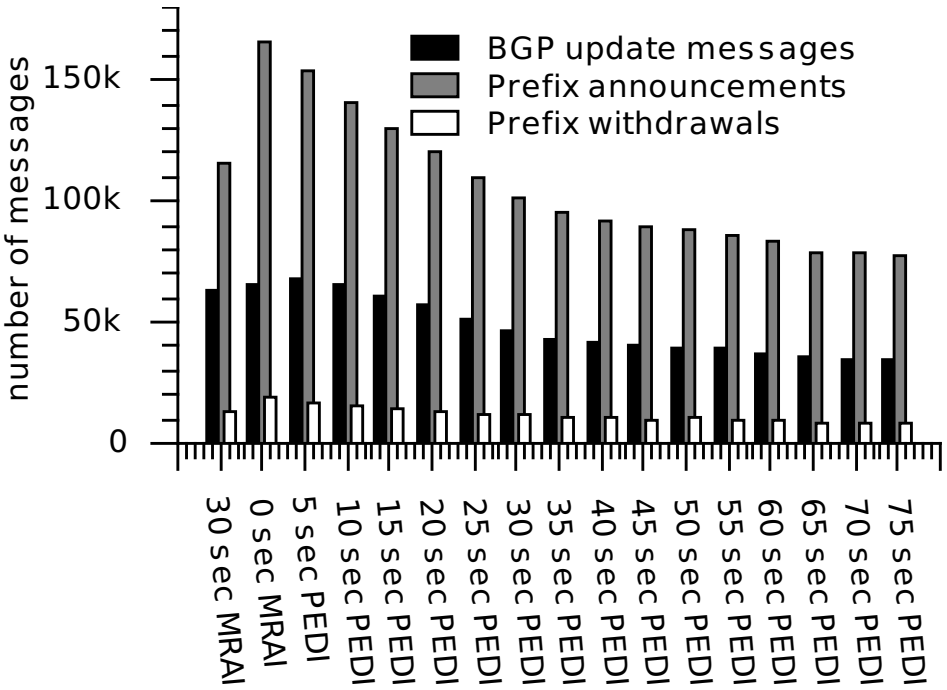


# Results – APNIC






# Results – Routeviews



# Results in Numbers

- Reduction of announcements compared to 30s MRAI:
- APNIC dataset:
  - **20%** for 35s PEDI
  - **29%** for 65s PEDI
- Routeviews dataset:
  - **18%** for 35s PEDI
  - **32%** for 65s PEDI



# Good results but...

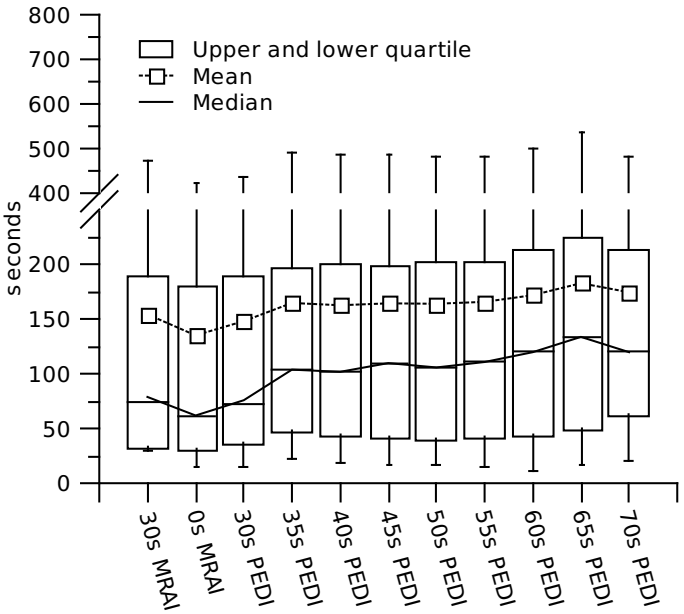
What about convergence time!?

# What is convergence in BGP?

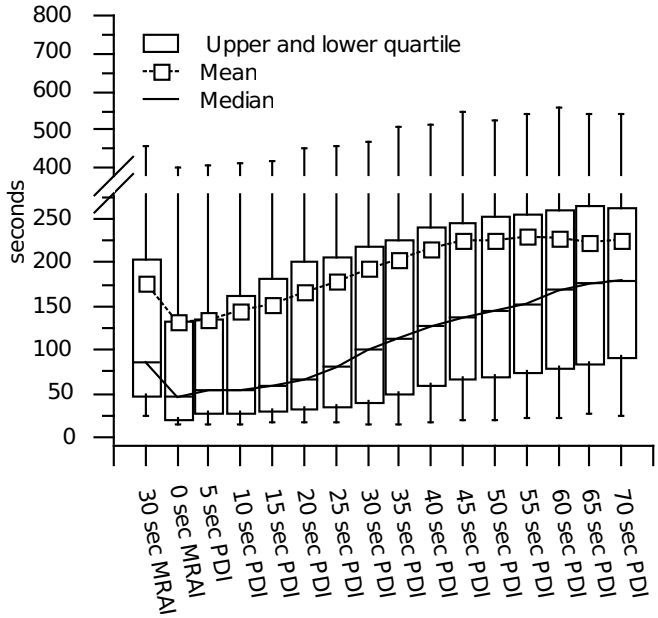
- Convergence is commonly understood as:
  - All ASes participating in a BGP system have received the latest, up-to-date routing information for a certain prefix
  - A BGP system is stable
- Approximated by: How long does it take for the upstream peer to have a stable route to a prefix?



# Convergence time approximation – APNIC data



# Convergence time approximation – Routeviews data






# Convergence refined – optimality vs. reachability

- Convergence in BGP is actually twofold:
- Optimality:
  - Every AS in the BGP system has **the best path** to the originator/owner of the prefix
  - Control plane convergence
  - The BGP system is stable
- Reachability:
  - Every AS in the BGP system has **a** path to the originator/owner of the prefix
  - Path doesn't need to be valid, data delivery needs to be ensured
  - Data plane convergence
  - BGP system doesn't need to be stable to achieve this state
  - Control plane convergence only needed up to an *altBGP* speaker
- Reachability ensures data delivery, optimality the best path

# Convergence refined – optimality vs. reachability

- We have 4 possible events that trigger instability:
  - $T_{long}$ – A link failure triggers an announcement of a longer (or equal-length) AS-path
  - $T_{short}$ – A link recovery triggers an announcement of a shorer AS-path
  - $T_{down}$ – A link failure triggers a withdrawal
  - $T_{up}$ – A link recovery triggers a new announcement
- Reachability and optimality are achieved at different times, depending on the event:
  - $T_{long}$ – Reachability is achieved before optimality
  - $T_{short}$ – Reachability is already ensured, only optimality needs to be achieved
  - $T_{down}$ – Reachability can not be achieved, optimality is achieved when every peer has withdrawn the route
  - $T_{up}$ – Reachability and optimality are mostly achieved at the same time, optimality can be delayed by timers though

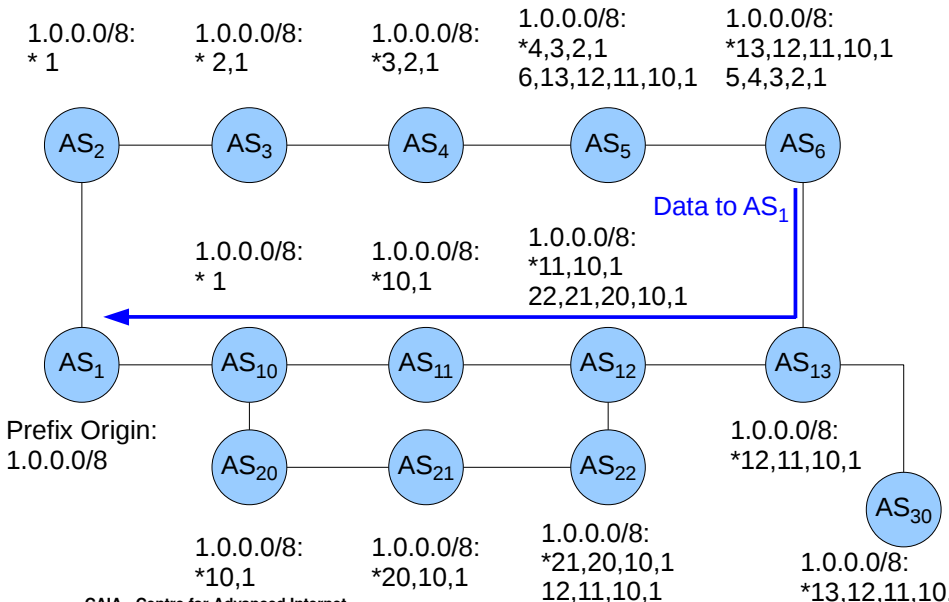


# Convergence with MRAI at 30s and PEDI at 35s

Announcement of initial route at $AS_6$:

- PED: 5 seconds
- MRAI: 60-120 seconds

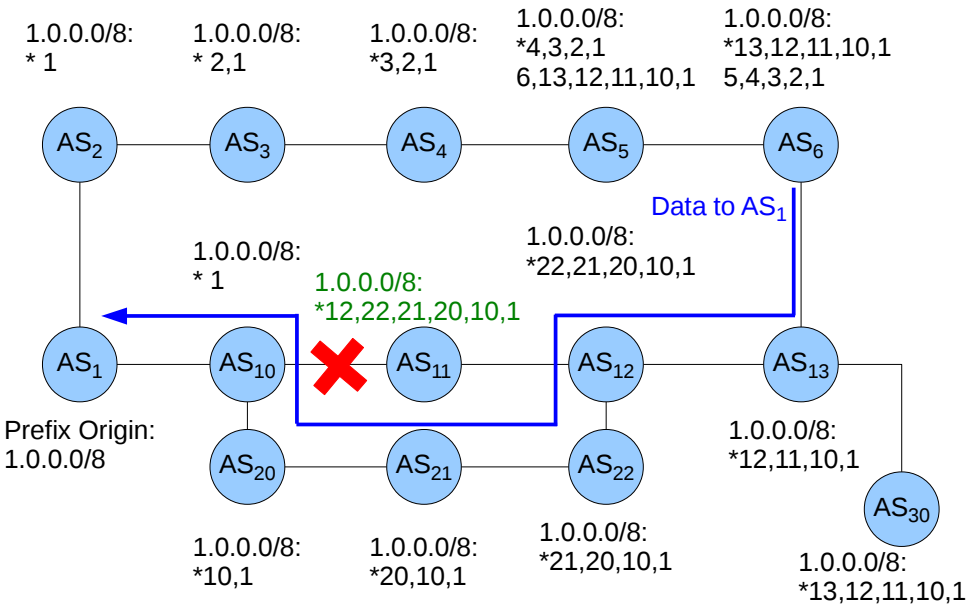
Stable System:



# Convergence – MRAI at 30s and PEDI at 35s

$T_{long}$ between $AS_{10}$ and $AS_{11}$: Reachability achieved ($AS_{11}$)

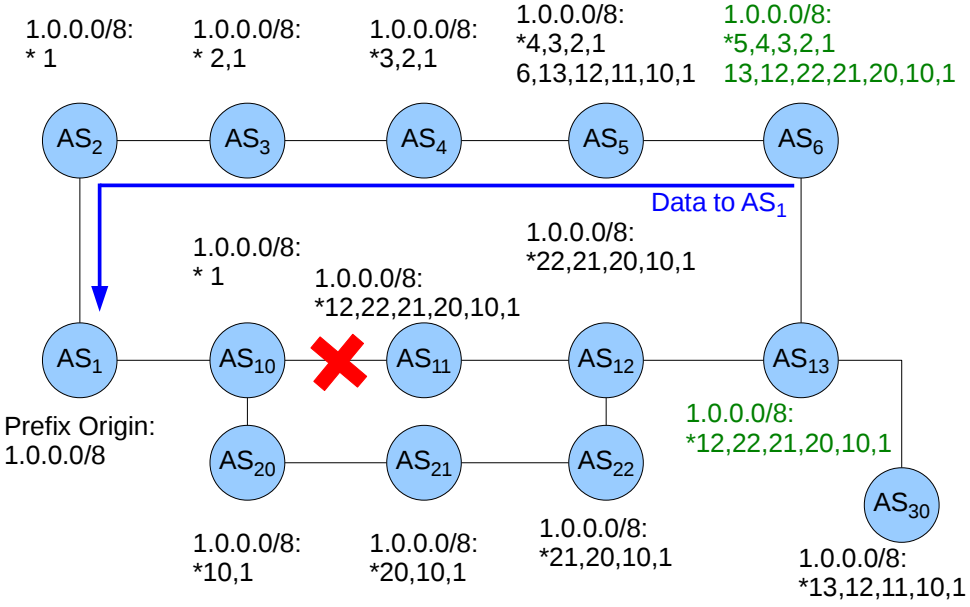
- PED: 0 seconds
- MRAI: 0-4 or 29-30 seconds






# Convergence – MRAI at 30s and PEDI at 35s

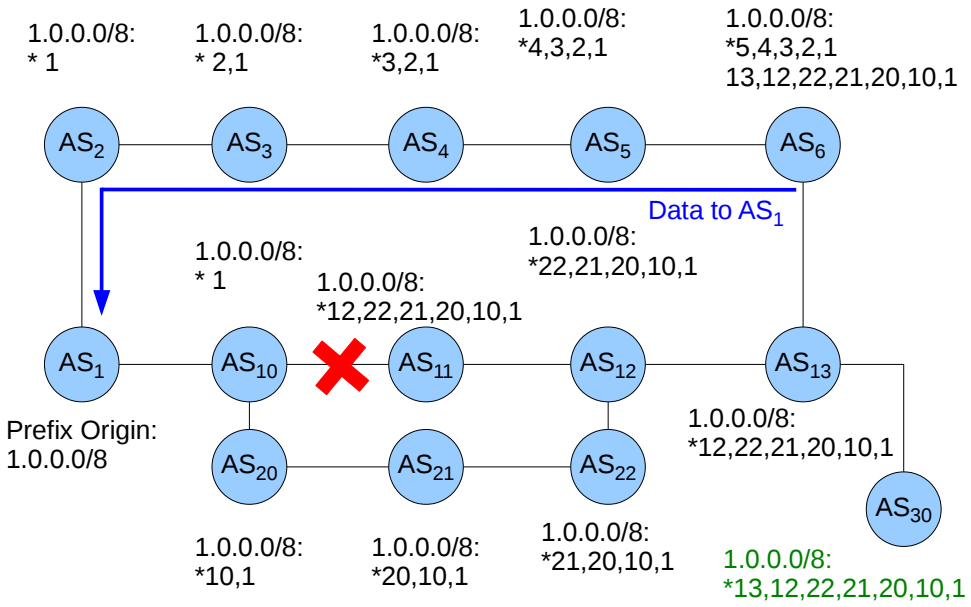
$T_{long}$ between $AS_{10}$ and $AS_{11}$: MRAI goes further within 1-30 seconds



# Convergence – MRAI at 30s and PEDI at 35s

$T_{long}$ between $AS_{10}$ and $AS_{11}$: Optimality achieved:

- PED: 66 seconds (+-jitter)
- MRAI: 2-58 seconds

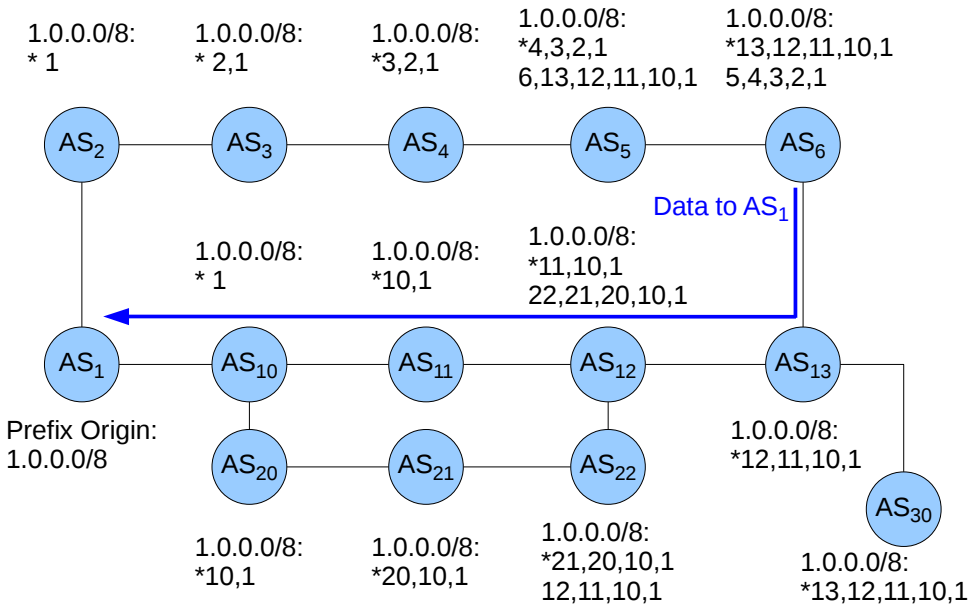





# Convergence – MRAI at 30s and PEDI at 35s

$T_{short}$ between $AS_{10}$ and $AS_{11}$: Optimality achieved:

- PED: 2 seconds
- MRAI: 31-33 and 55-60 seconds



# Convergence – MRAI at 30s and PEDI at 35s

Announcement of initial route at $AS_6$:

- PED: 5 seconds
- MRAI: 60-120 seconds

$T_{down}$ at $AS_1$ Optimality achieved (all routes withdrawn):

- PED: 0 seconds
- MRAI: 0 seconds

$T_{up}$ at $AS_1$ Optimality achieved (same as initial anouncement):

- PED: 5 seconds
- MRAI: 32-34, 58-60, 76-90 seconds



# Conclusions

- PED diminishes update load
- PED delays optimality in some cases
- PED is faster than MRAI for $T_{up}$ and $T_{short}$, slower for $T_{long}$ and same for $T_{down}$
- PED is more consistent than MRAI
- 35 second PEDI is a safe default value
- Can be deployed incrementally
- Interacts well with MRAI

# Future work

- Dynamic PEDI calculation
- Improvement of heuristics / addition of new heuristics
- Improvement of implementation
- Improvement of evaluation tools



Thank you!

Questions?