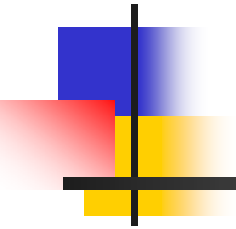
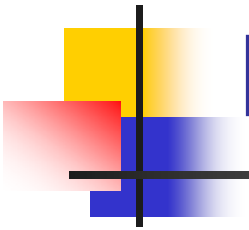


Taming BGP

An incremental approach to improving the dynamic properties of BGP

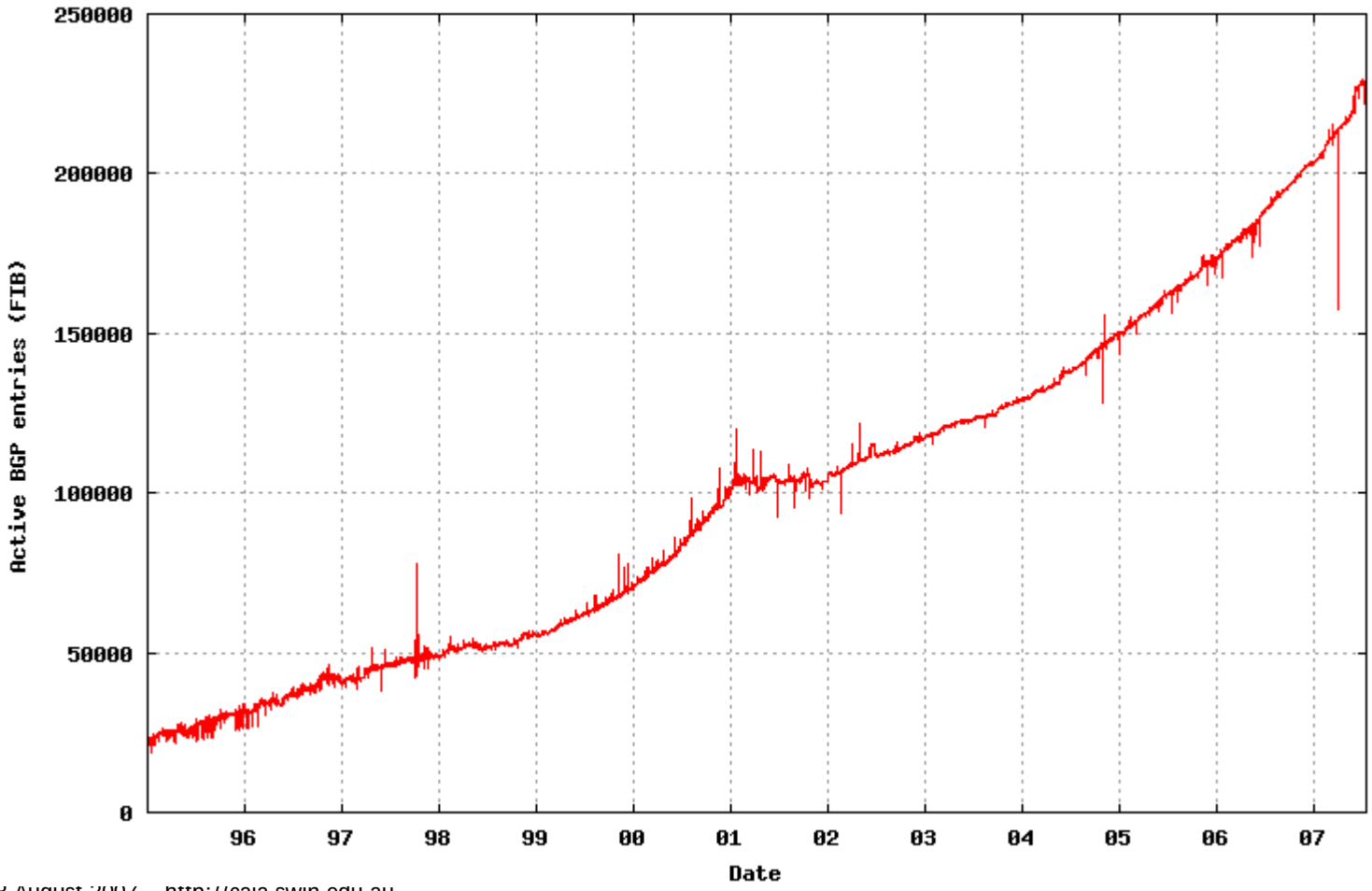
Geoff Huston

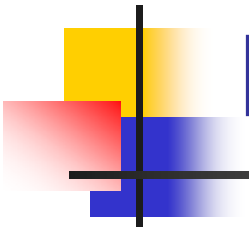


BGP is ...

- The inter-domain routing protocol for the Internet
- An instance of a Distance Vector Protocol with explicit Path Vector attributes

BGP Growth: Number of Routed Objects





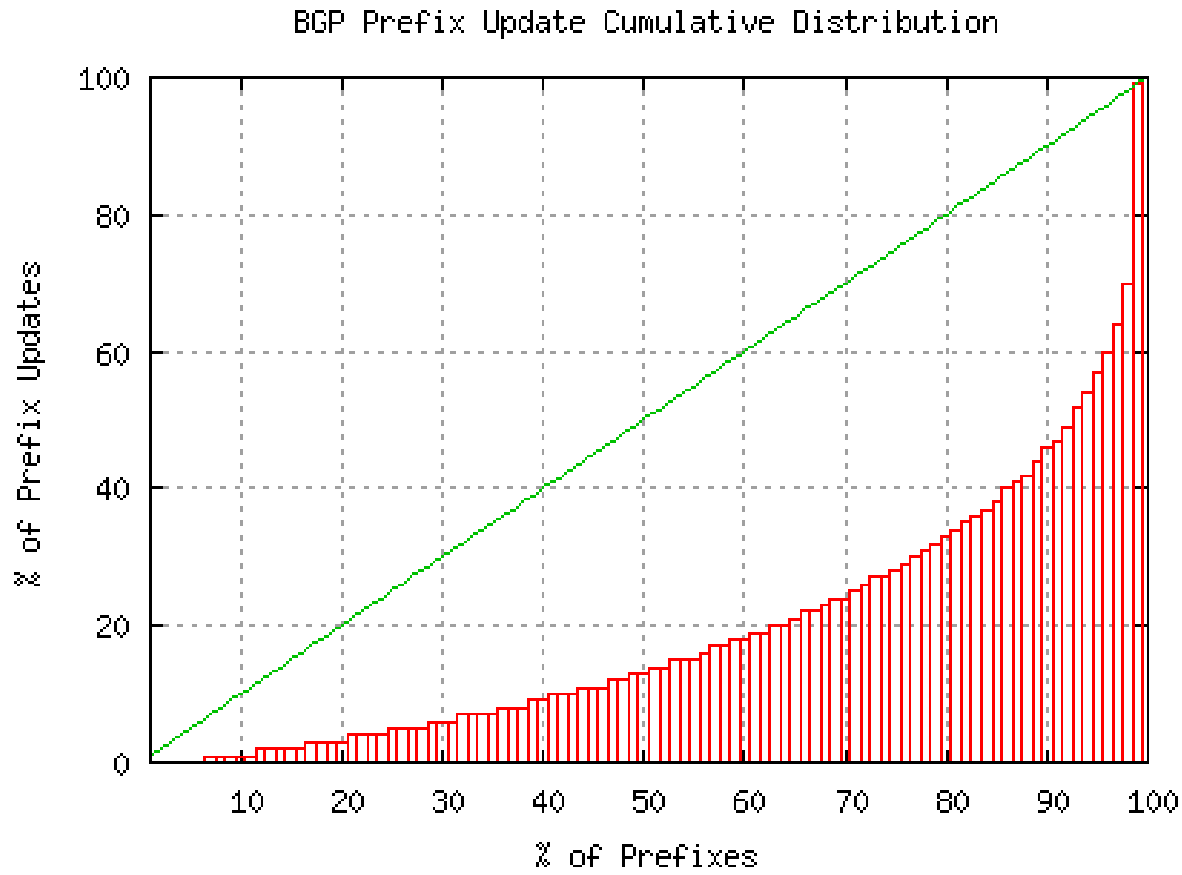
BGP Questions

- Are there practical limits to the size of the routed network ?
  - routing database size ?
  - routing update processing load ?
  - Time to reach “converged” routing states ?

Current Understandings

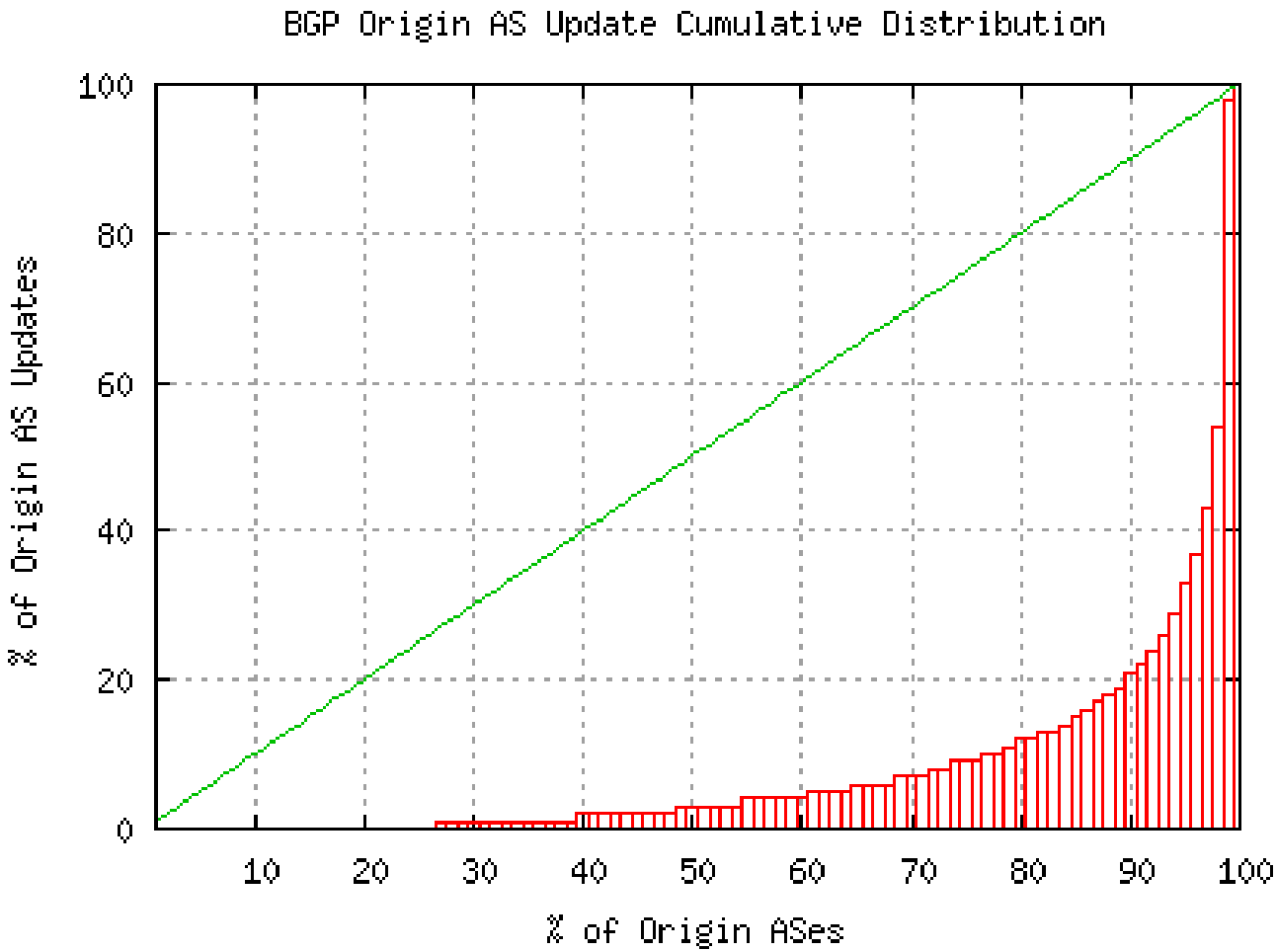
- The protocol message peak rate is increasing faster than the number of routed entries
  - BGP is a “chatty” protocol
  - Dense interconnection implies higher levels of path exploration to stabilize on best available paths
- Some concern that BGP in its current form has some practical limits in terms of size and practical convergence times

Update Distribution by Prefix

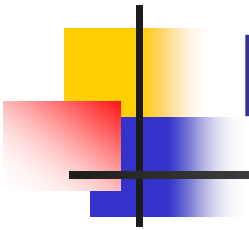


BGP Updates recorded at AS2.0, June 28 – July 12

Update Distribution by Origin AS



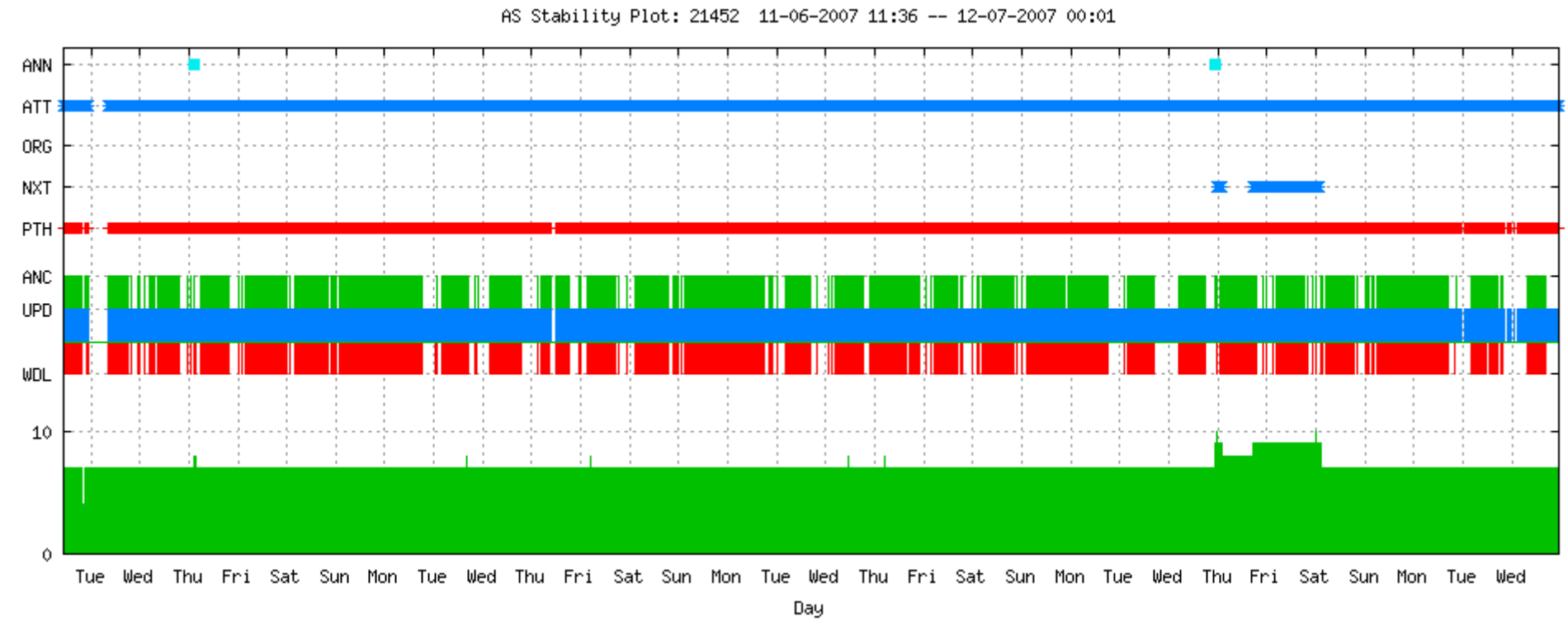
BGP Updates recorded at AS2.0, June 28 – July 12



Previous Work

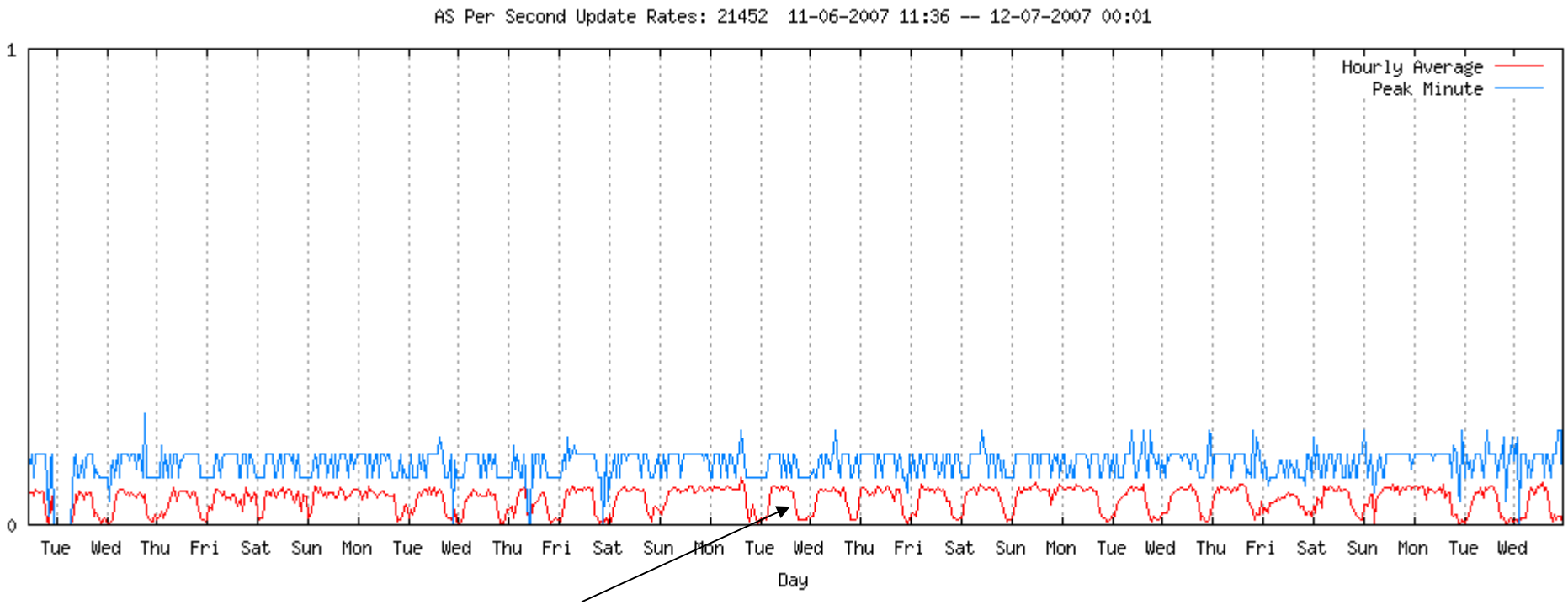
- The BGP load profile is heavily skewed, with a small number of route objects contributing a disproportionate amount of routing update load
- If we could identify this skewed load component within the BGP protocol engine then there is the potential for remote BGP speakers to significantly reduce the total BGP processing load profile

What's the cause here?



BGP Updates recorded at AS2.0, June 28 – July 12

What's the cause here?

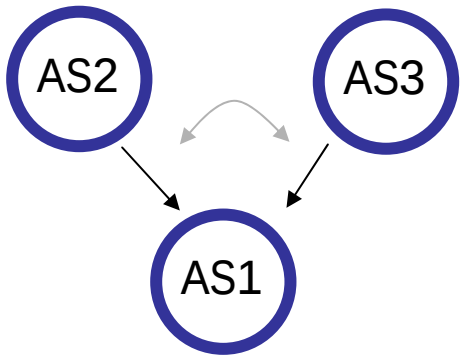


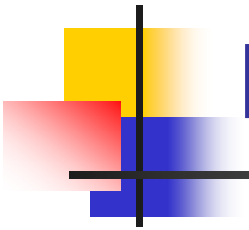
This daily cycle of updates with a weekend profile is a characteristic signature of a residential ISP performing some form of load-based routing

BGP Updates recorded at AS2.0, June 28 – July 12

Poor Traffic Engineering?

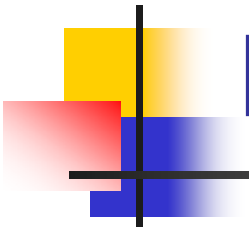
- An increasing trend to “multi-home” an AS with multiple transit providers
- Spread traffic across the multiple transit paths by selectively altering advertisements
- The use of load monitors and BGP control systems to automate the process
- Poor tuning of the automated traffic engineering process produces extremely unstable BGP outcomes!





BGP Update Load Profile

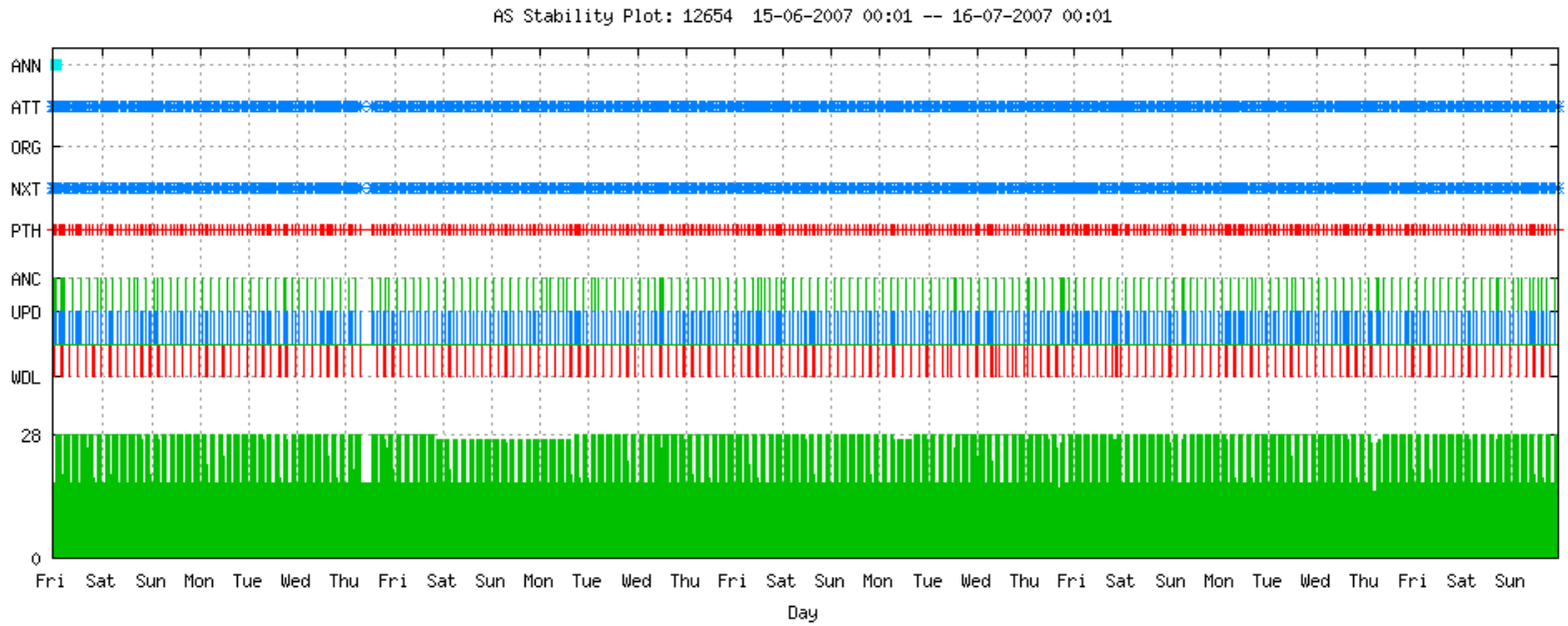
- It appears that the majority of the BGP load is caused by a very small number of unstable origination configurations, possibly driven by automated systems with limited or no feedback control
- This problem is getting larger over time
- The related protocol update load consumes routing resources, but does not change the base information state – its generally oscillations across a smaller set of states

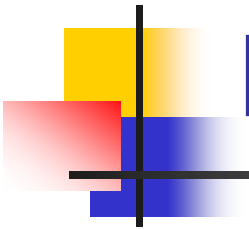


BGP “Beacons”

- Act as control points in the BGP environment, as they operate according to a known periodic schedule of announcements
  - Typical profile: 2 hours “up” then 2 hours “down” at origin
- Analyse update behaviour at a BGP observation point

BGP Beacon “signature”

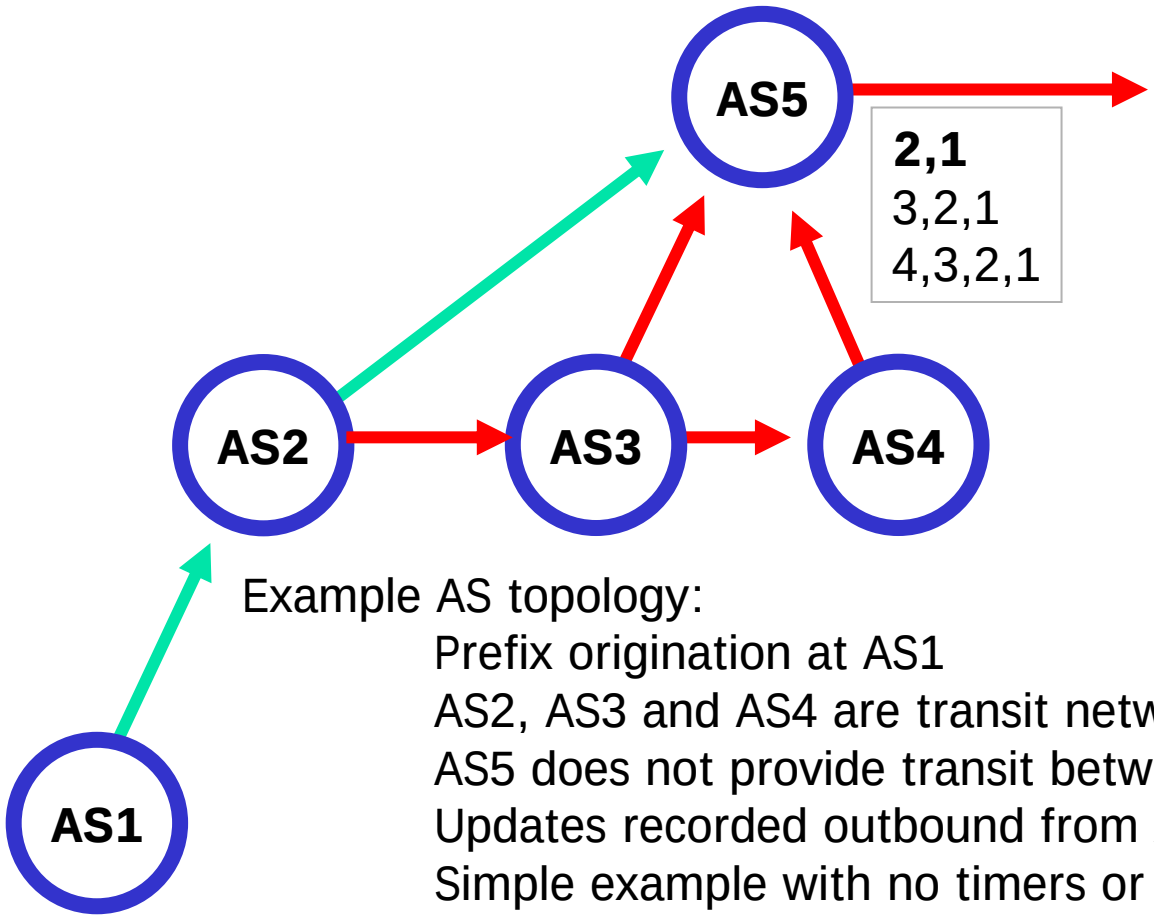




BGP “Beacons”

- Each withdrawal at the beacon source can generate up to 10 updates at a remote observation point!
- Hypothesis: BGP Path exploration on withdrawal appears to be a major factor in overall BGP update load

BGP Withdrawals Examined..



Example AS topology:

Prefix origination at AS1

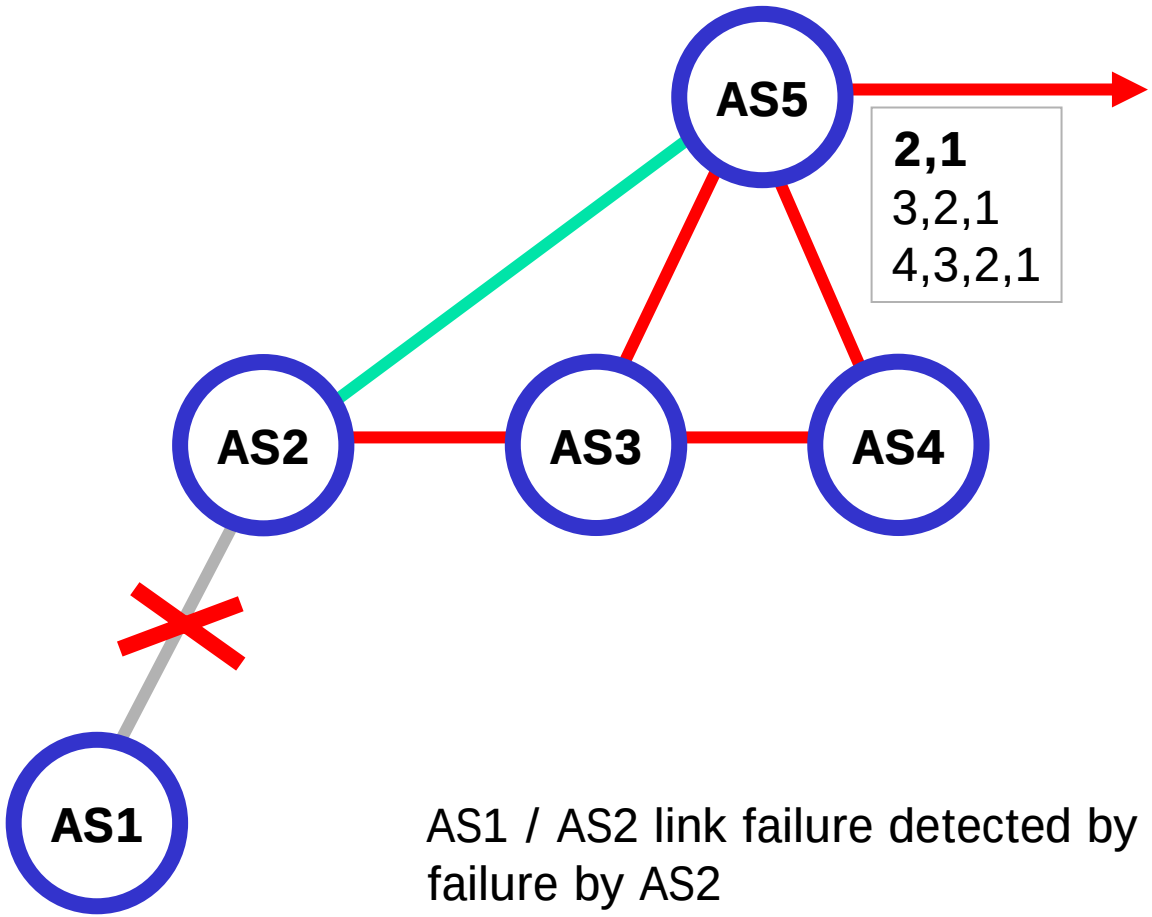
AS2, AS3 and AS4 are transit networks for AS1 and AS5

AS5 does not provide transit between AS2, AS3 or AS4

Updates recorded outbound from AS5

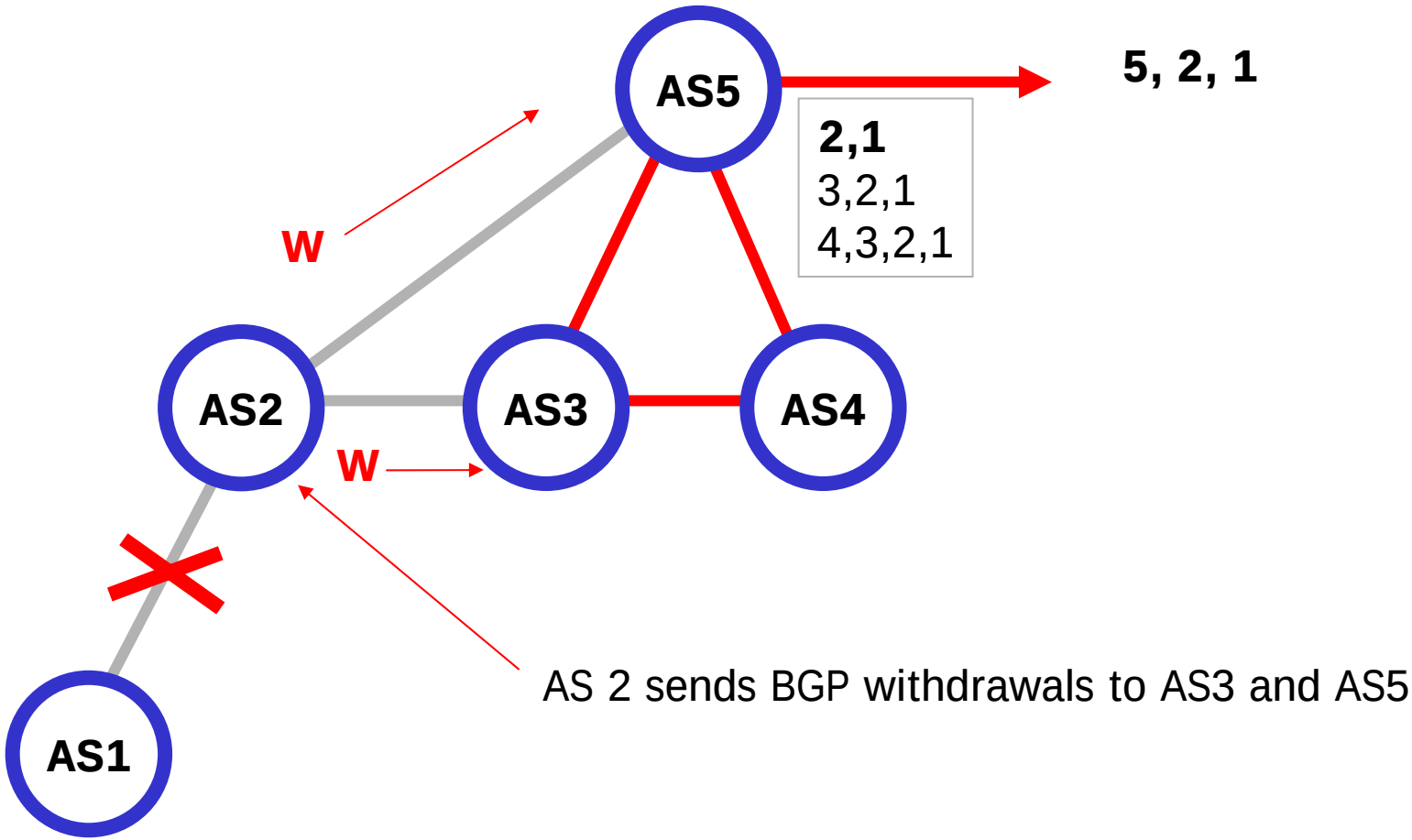
Simple example with no timers or damping controls

BGP Withdrawals

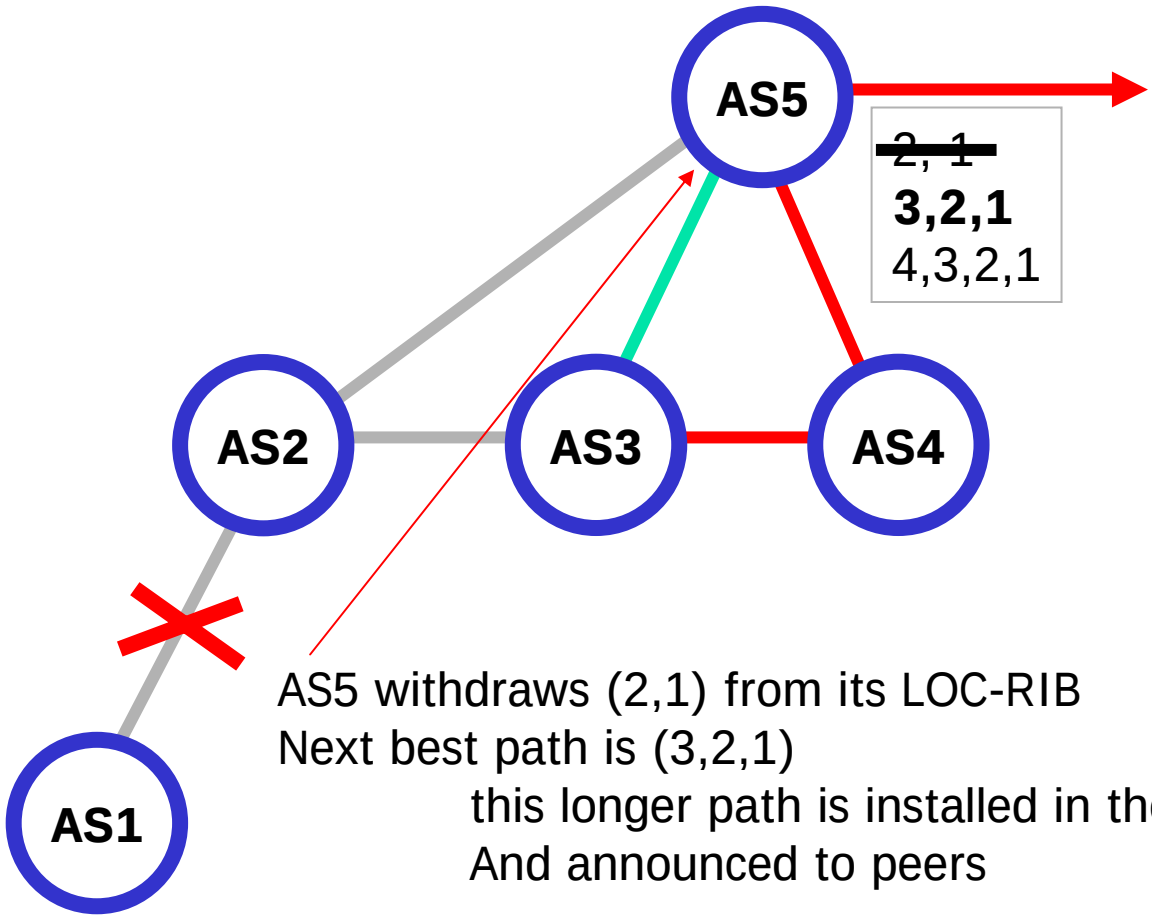


AS1 / AS2 link failure detected by BGP keepalive failure by AS2

BGP Withdrawals

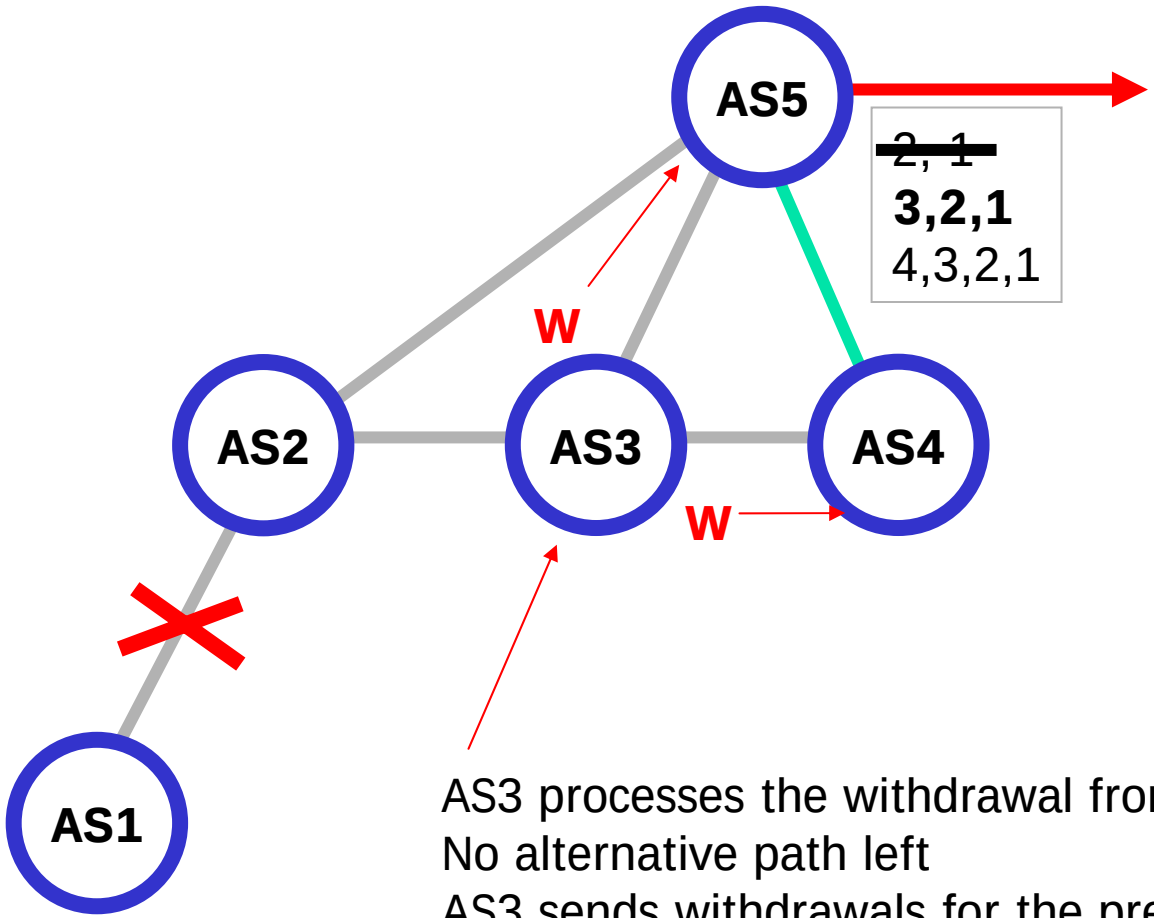


BGP Withdrawals



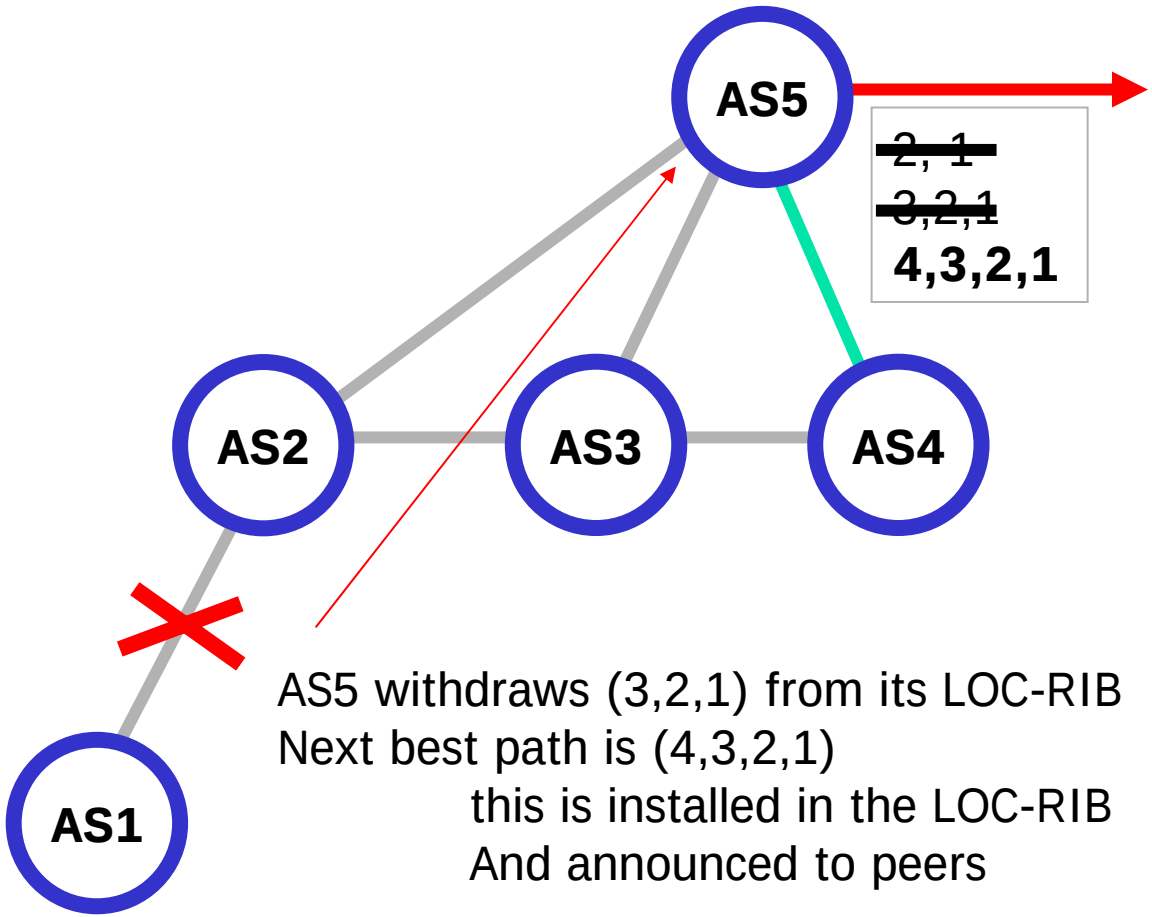
AS5 withdraws (2,1) from its LOC-RIB
Next best path is (3,2,1)
this longer path is installed in the LOC-RIB for AS 5
And announced to peers

BGP Withdrawals



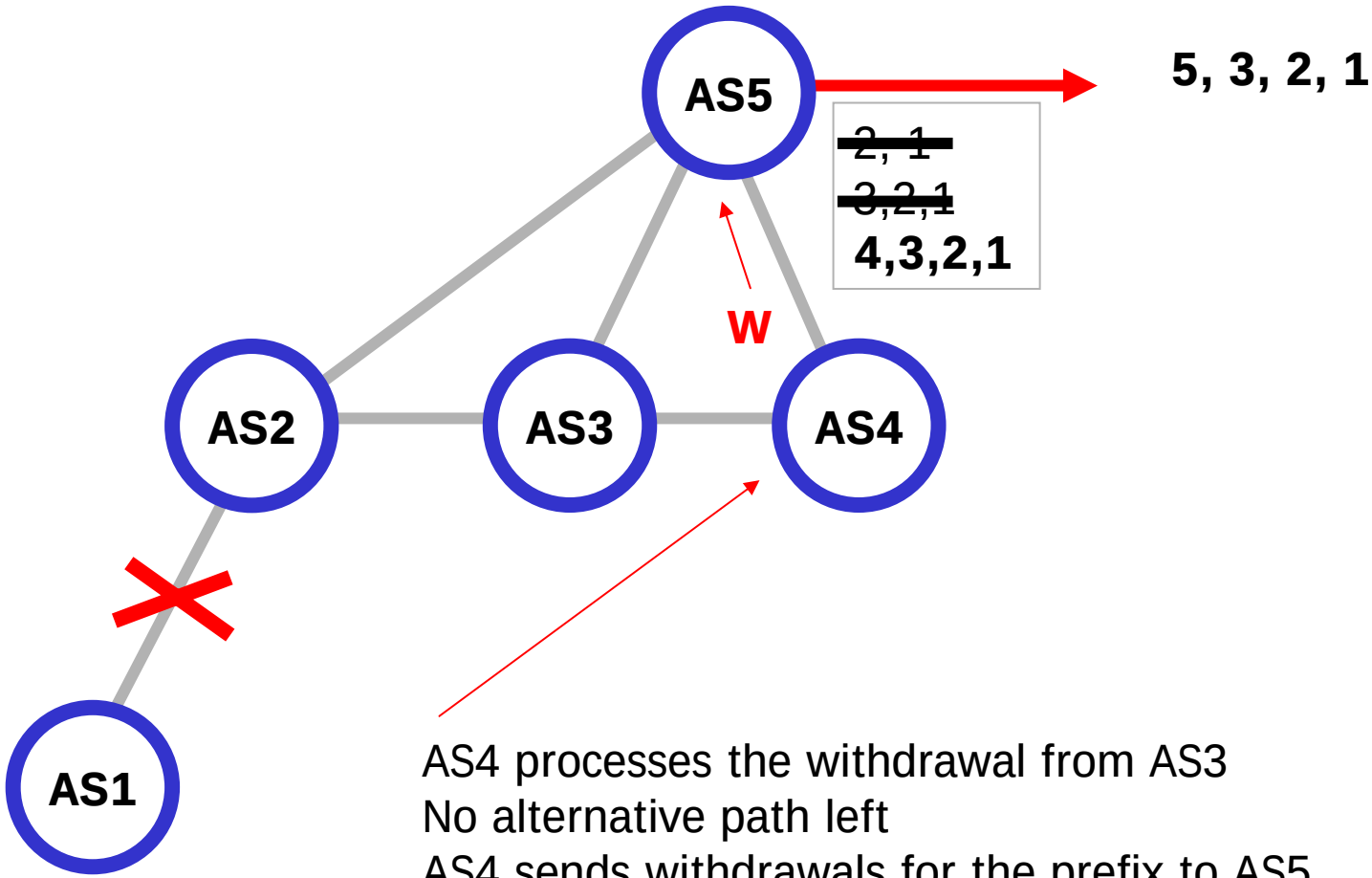
AS3 processes the withdrawal from AS2
No alternative path left
AS3 sends withdrawals for the prefix to AS4 and AS5

BGP Withdrawals



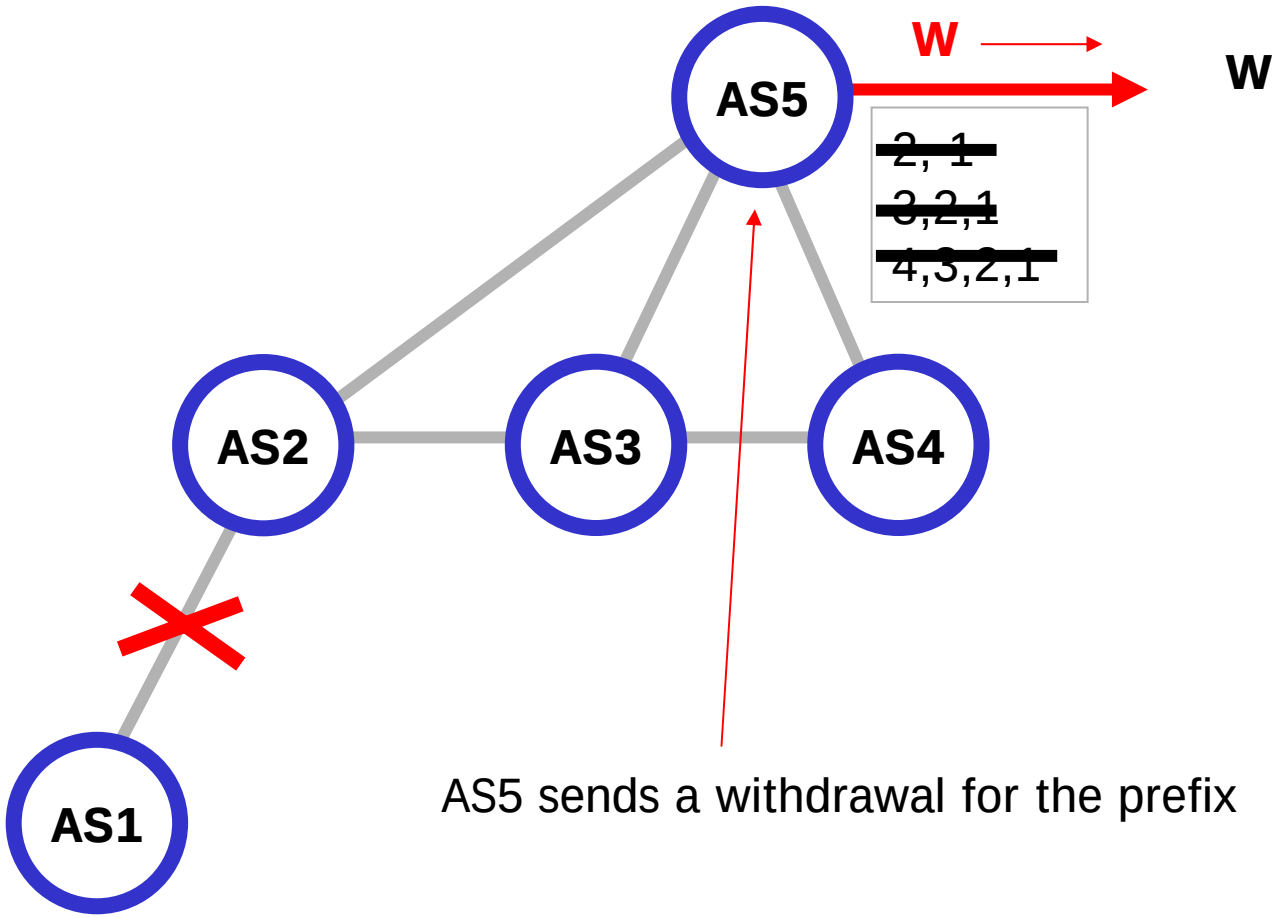
AS5 withdraws (3,2,1) from its LOC-RIB
Next best path is (4,3,2,1)
this is installed in the LOC-RIB
And announced to peers

BGP Withdrawals



AS4 processes the withdrawal from AS3
No alternative path left
AS4 sends withdrawals for the prefix to AS5

BGP Withdrawals



BGP Path Exploration

- Announcement sequence from AS 5:

Steady state:

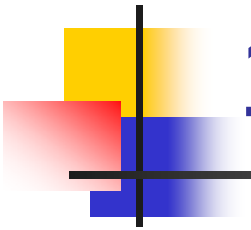
5,2,1

Withdrawal sequence:

1. Update with Path: 5,3,2,1
2. Update with Path: 5,4,3,2,1
3. Withdrawal

Mitigating BGP Update Loads

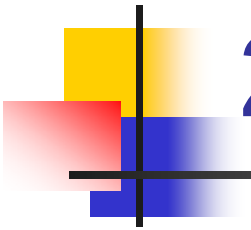
- Current set of “tools” to mitigate BGP update overheads:
  1. Minimum Route Advertisement Interval Timer (MRAI)
  2. Withdrawal MRAI Timer
  3. Sender Side Loop Detection
  4. Route Flap Damping
  5. Output Queue Compression



1. MRAI Timer

- Optional timer in BGP
  - ON in ciscos (30 seconds)
  - OFF in Junipers (0 secconds)
- Suppress the advertisement of successive updates to a peer for a given prefix until the timer expires
- Commonly implemented as suppress ALL updates to a peer until the per-peer MRAI timer expires

- *Output Queue (adj-rib-out) process*



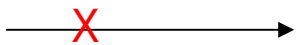
2. Withdrawal MRAI TIMER

- Variant on MRAI where withdrawals are also time limited in the same way as updates
- *Output Queue (adj-rib-out) process*

3. Sender Side Loop Detection

- Suppress passing an update to an EBGP neighbour if the neighbor's AS is in the AS Path
- *Output Queue (adj-rib-out) process*

192.9.200.0/24
Path 4,3,1

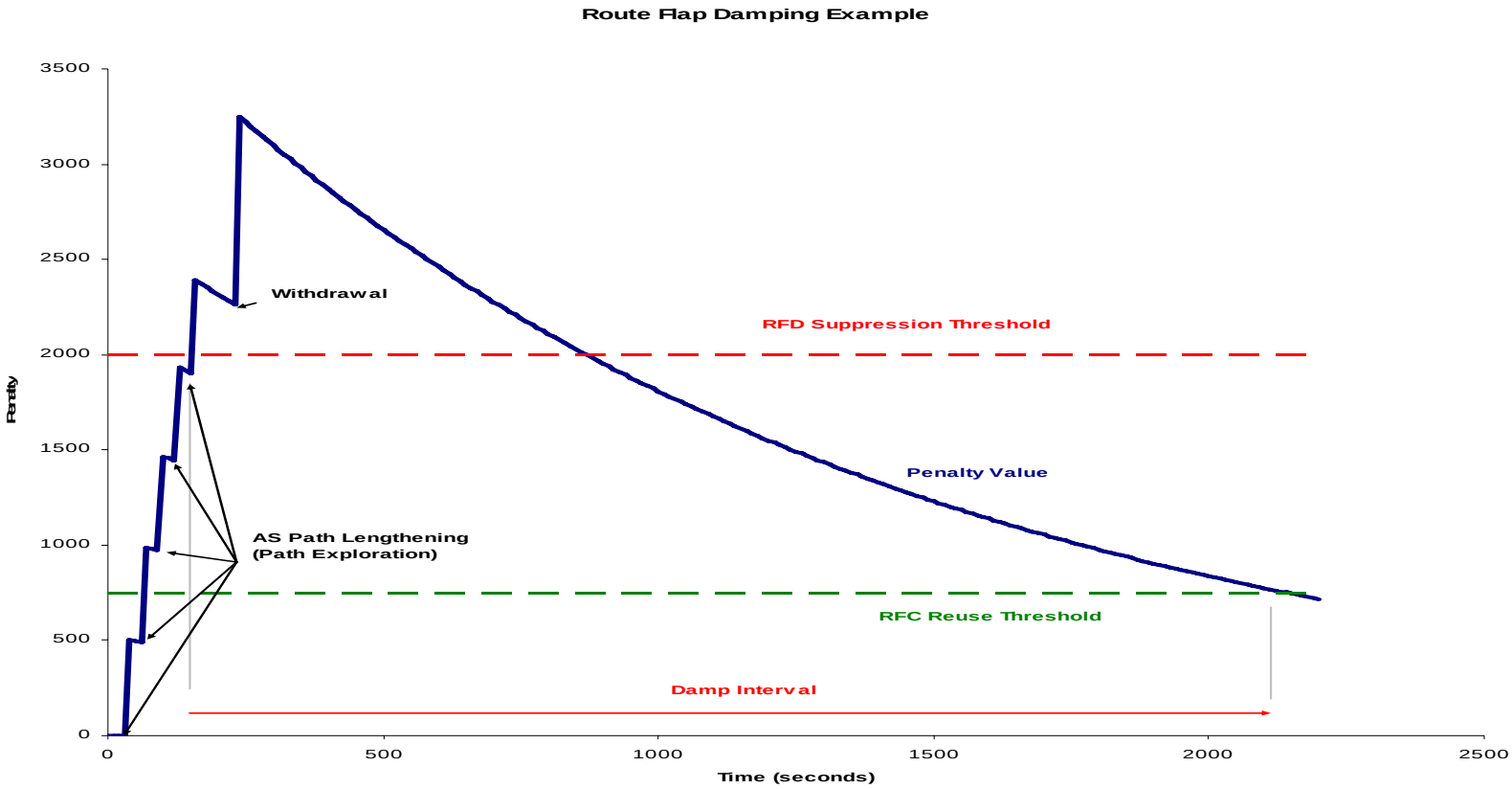


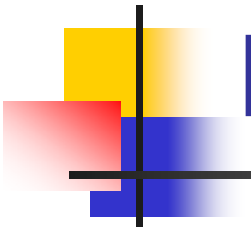
Update to AS3 suppressed by SSLD

4. Route Flap Damping

- RFD attempts to apply a heuristic to identify noisy prefixes and apply a longer term suppression to update propagation
- Uses the concept of a “penalty” score applied to a prefix learned from a peer
  - Each update and withdrawal adds to the score
  - The score decays exponentially over time
  - If the score exceeds a suppress threshold the route is damped
  - Damping remains in place until the score drops below the release threshold
  - Damping is applied to the adj-rib-in
- *Input Queue (adj-rib-in) process*

RFD Example


Route Flap Damping Example
Penalty
3500
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
Time (seconds)
Withdrawal
RFD Suppression Threshold
Penalty Value
AS Path Lengthening
(Path Exploration)
RFC Reuse Threshold
Damp Interval

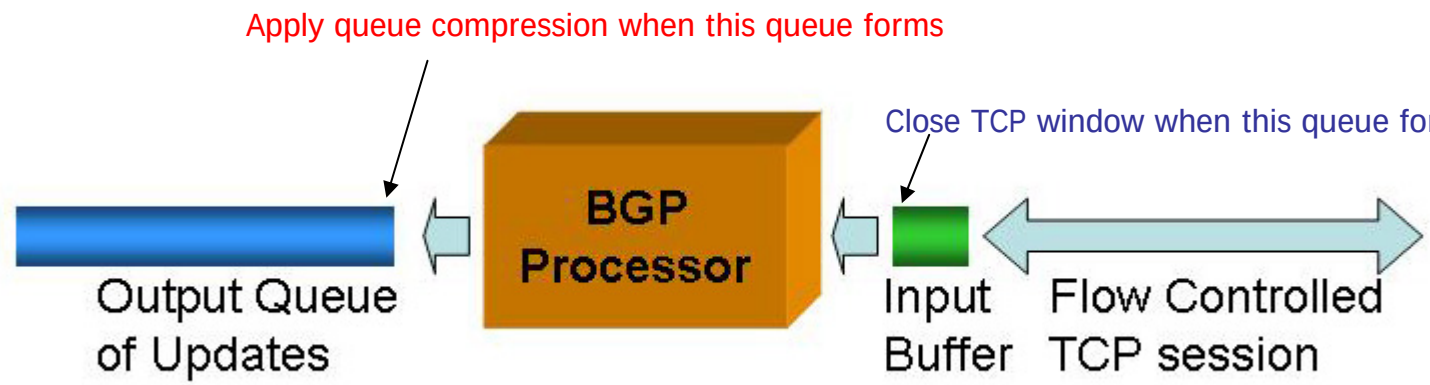


RFD and Network Operators

- RFD does not appear to be effective
- It causes the routing system to take extended intervals of hours rather than minutes to reach convergence
- It has done little to reduce the total routing update load
- It causes operational outages
- Edge link flapping is not prevalent in the routing system today, and Route Flap Damping exacerbates poor performance characteristics of BGP

5. Output Queue Compression

- BGP is a rate-throttled protocol (due to TCP transport)
  - A process-loaded BGP peer applies back pressure to the 'other' side of the BGP session by shutting down the advertised TCP recv window
  - The local BGP process may then perform queue compression on the output queue for that peer, removing queued updates that refer to the same prefix
- *Output Queue (adj-rib-out) process*



BGP Update Types

Announced-to-Announced Updates

Withdrawn-to-Announced Updates

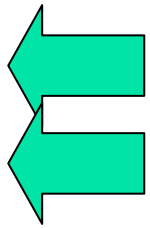
Announced-to-Withdrawn
Withdrawn-to-Withdrawn

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)
WA+	Announcement of a withdrawn prefix, with longer AS Path
WA-	Announcement of a withdrawn prefix, with shorter AS Path
WA0	Announcement of a withdrawn prefix, with different AS Path of the same length
WA*	Announcement of a withdrawn prefix with the same AS Path, but different attributes
WA	Announcement of a withdrawn prefix with the same AS Path and same attributes
AW	Withdrawal of an announced prefix
W W	Withdrawal of a withdrawn prefix (possible BGP error or a data collection error)

April 2007 BGP Update Profile

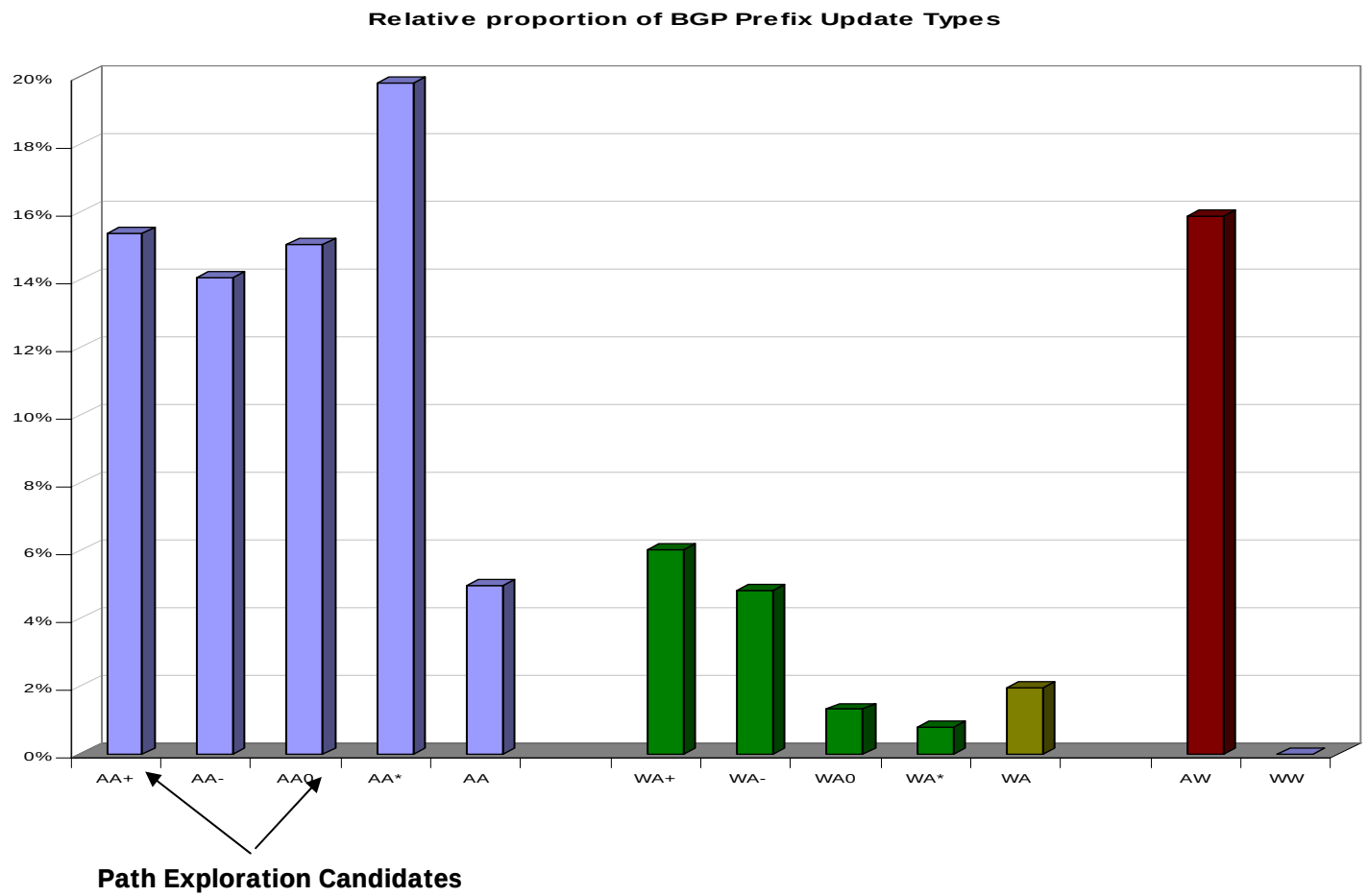
Totals of each type of prefix updates, using a recording of all BGP updates as heard by AS2.0 for the month of April 2007

Code	Count
AA+	607,093
AA-	555,609
AA0	594,029
AA*	782,404
AA	195,707
WA+	238,141
WA-	190,328
WA0	51,780
WA*	30,797
WA	77,440
AW	627,538
W W	0

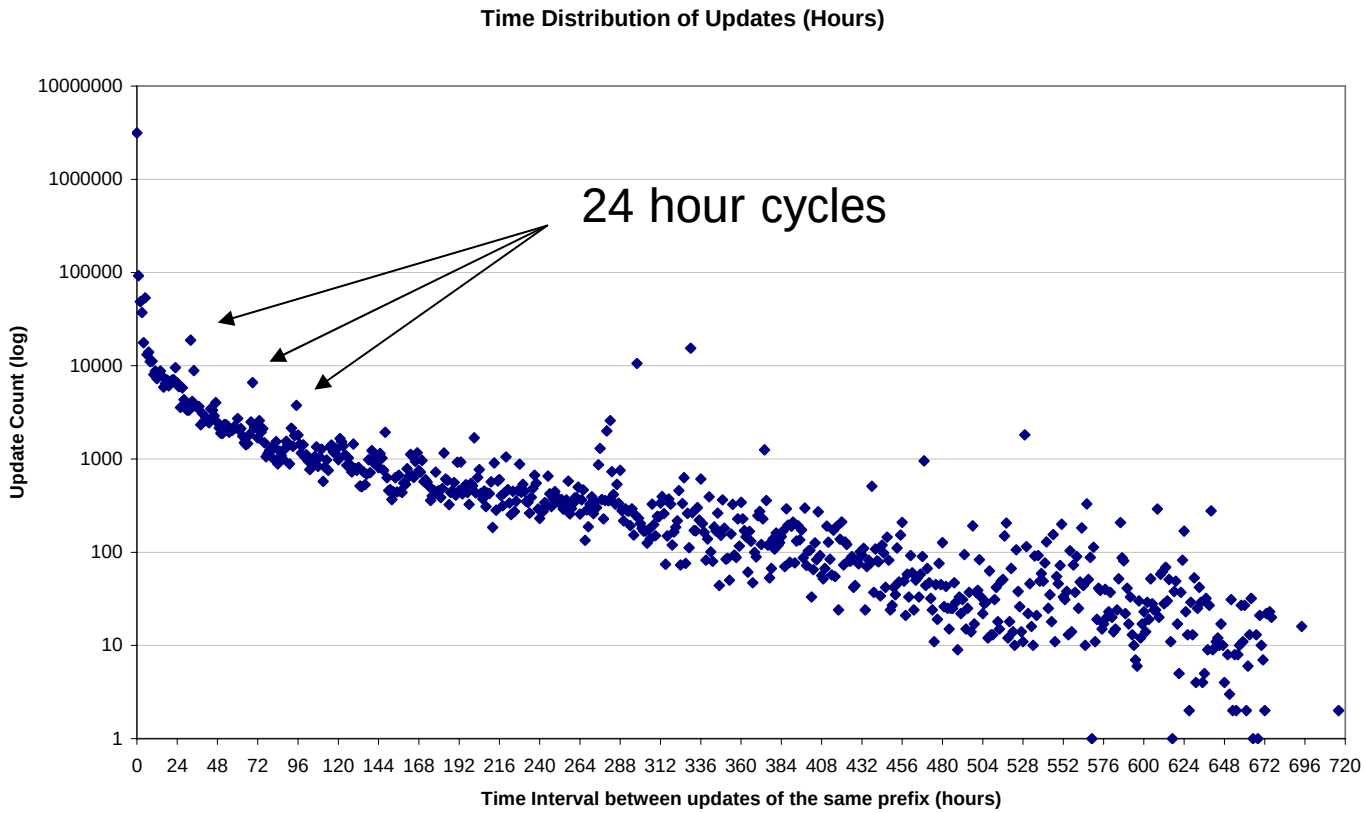


BGP Path Exploration?

BGP Update Profile

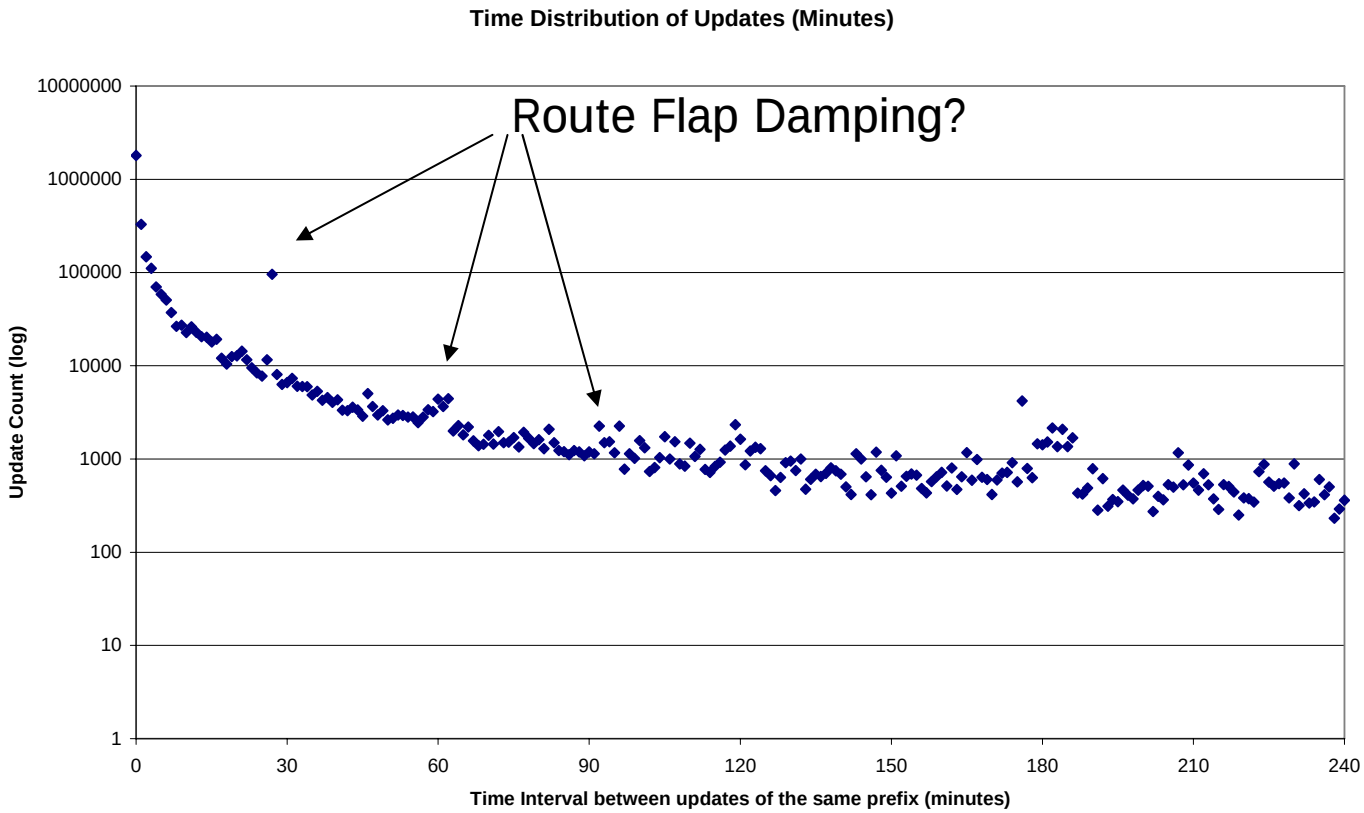


Time Distribution of Updates



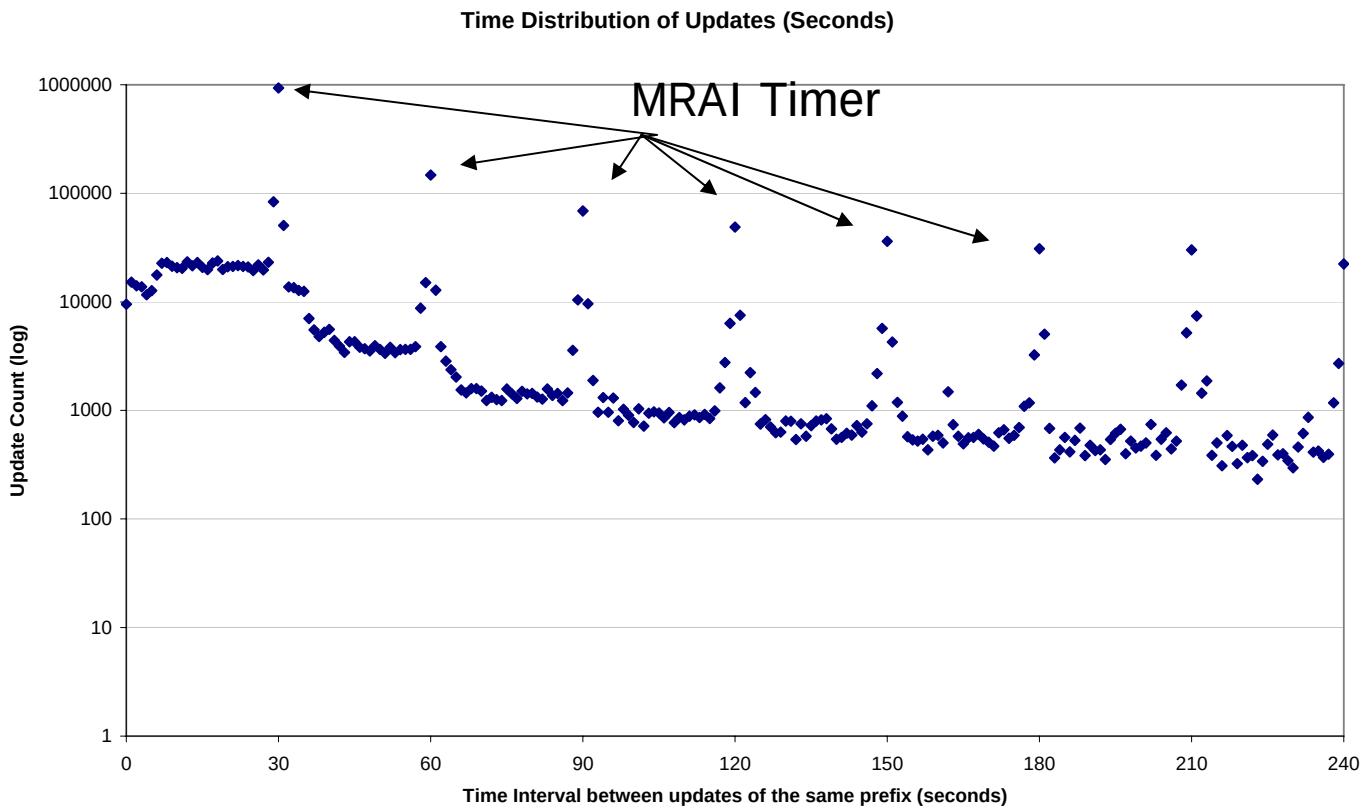
Elapsed time between received updates for the same prefix - days

Time Distribution of Updates



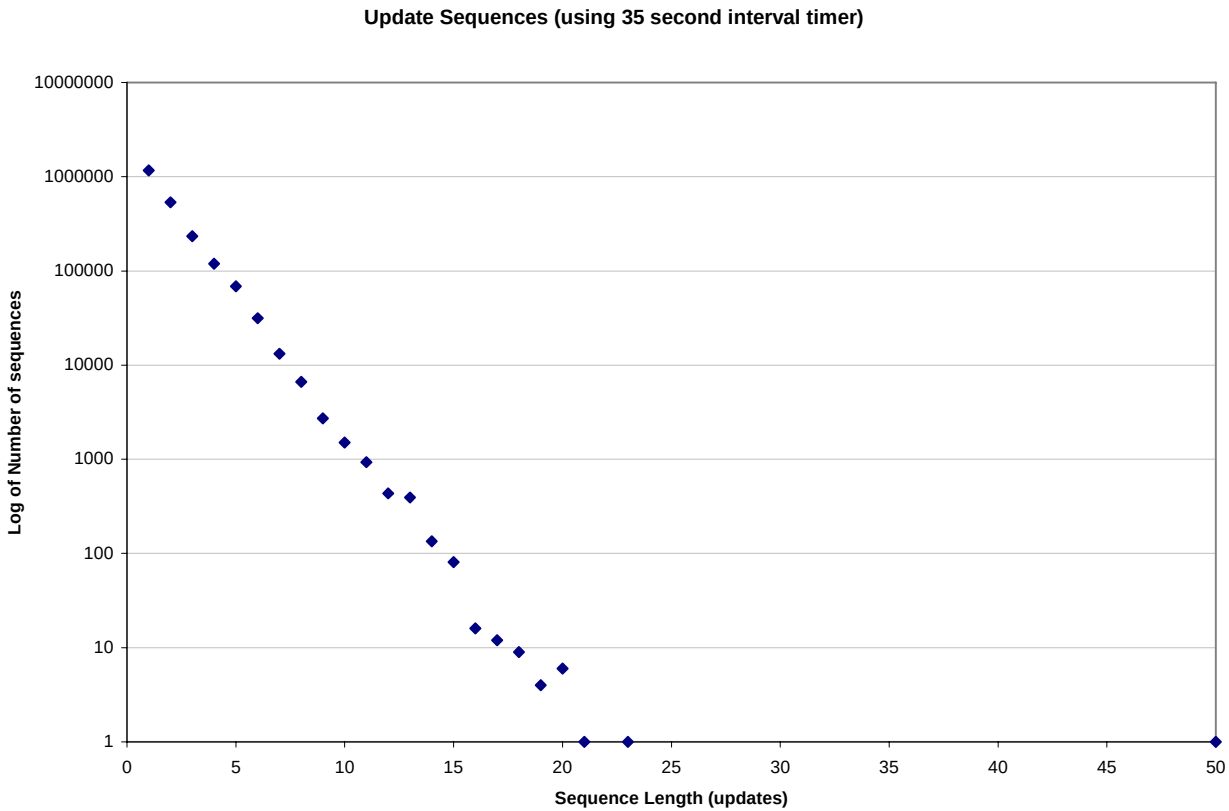
Elapsed time between received updates for the same prefix - hours

Time Distribution of Updates

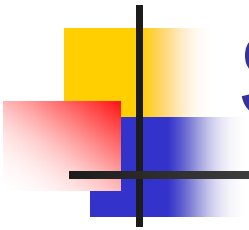


Elapsed time between received updates for the same prefix - seconds

Update Sequence Length Distribution



A “sequence” is a set of updates for the same prefix that are separated by an interval <= the sequence timer (35 seconds)



Some Observations

- RFD – long term suppression
  - Route Flap damping extends convergence times by hours with no real benefit offset
- MRAI – short term suppression
  - MRAI variations in the network make path exploration noisier
  - Even with piecemeal MRAI deployment we still have a significant routing load attributable to Path Exploration
- Output Queue Compression
  - Rarely triggered in today's network!

An alternate approach: Path Exploration Damping (PED)

- A prevalent form of path hunting is the update sequence of increasing AS path followed by a withdrawal, closely coupled in time

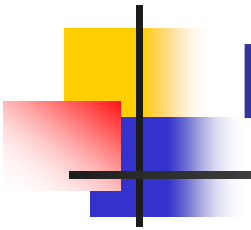
  {AA+ } *, AW

  The AA+ updates are intermediate noise updates in this case that are not valid routing states.

  Could a variation of Output Queue Compression be applicable here?

  i.e. Can these updates be locally suppressed for a short interval to see if they are path of a BGP Path Exploration activity? .

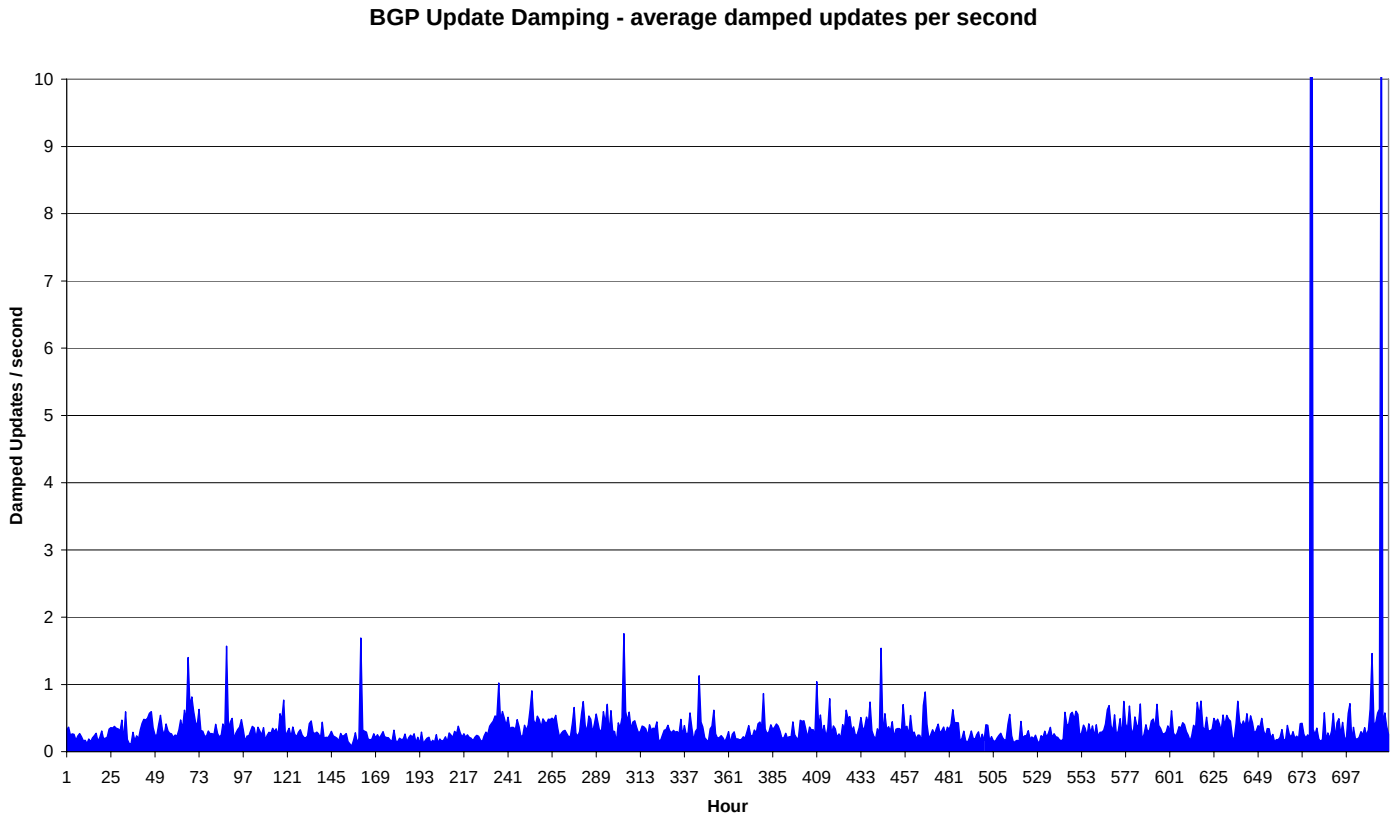
  The suppression would hold the update in the local output queue for a fixed time interval (in which case the update is released) or the update is further updated by queuing a subsequent update (or withdrawal) for the same prefix



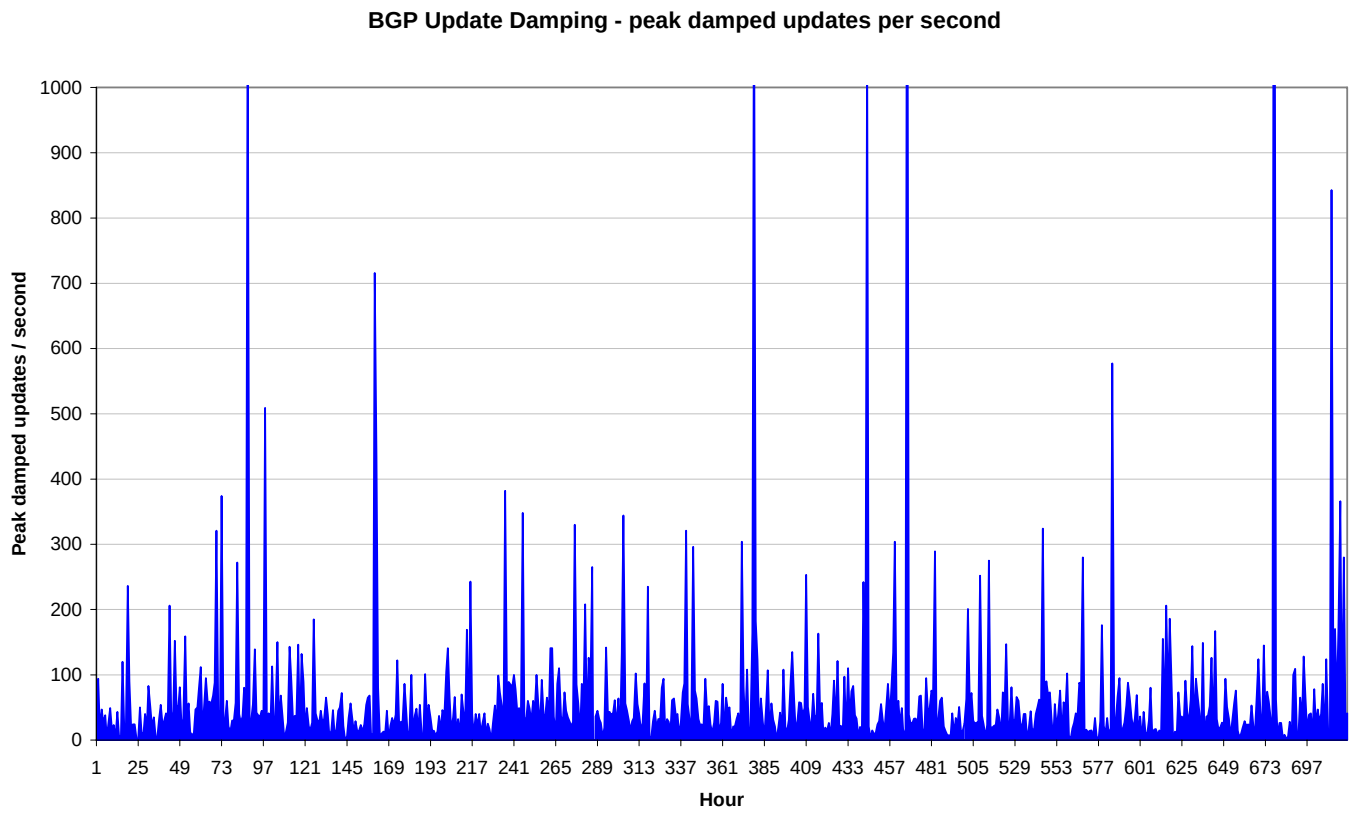
PED Algorithm

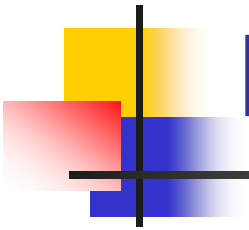
- Apply a 35 second MRAI timer to AA+, AA0 and AA updates queued to eBGP peers
- No MRAI timer applied to all other updates and all withdrawals
- 35 seconds is used to compensate for MRAI-filtered update sequences that use 30 second interval
- Algorithm:
  - If an update extends the AS path length then suppress its re-advertisement for 35 seconds, or until a further update for this prefix is queued for re-advertisement
  - Immediately re-advertise withdrawals and updates that reduce the AS Path length

PED Results on BGP data



PED Results on BGP data



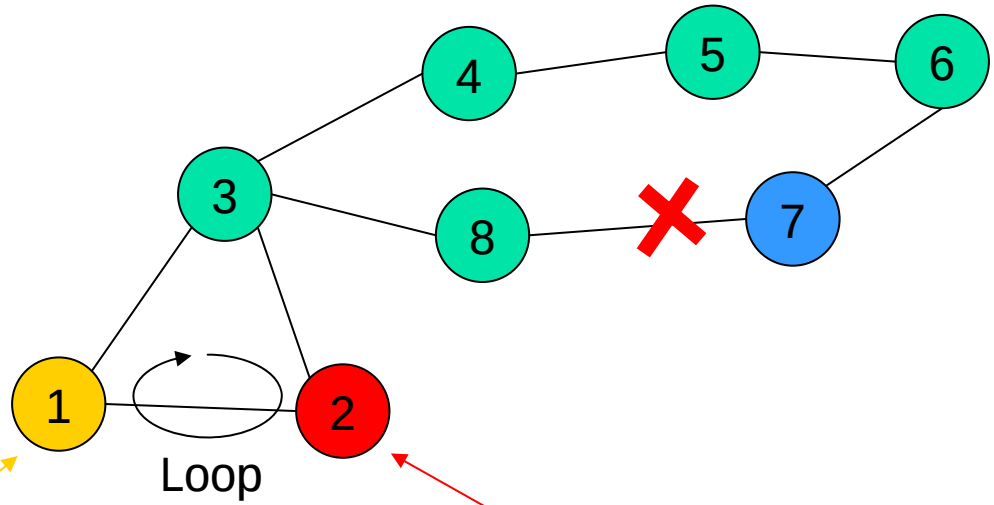


PED Results

- 21% of all updates collected in the sample data would've been eliminated by PED
- Average update rate for the month would fall from 1.60 prefix updates per second to 1.22 prefix updates per second
- Average peak update rates fall from 355 to 290 updates per second

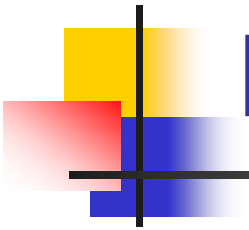
Could this PED suppression lead to transient Loops?

- Yes! (this is the case with MRAI and Output Queue Compression as well)



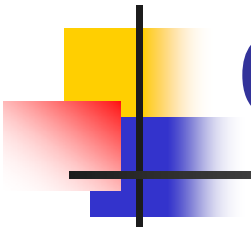
Update to 2 of 1,3,4,5,6,7 suppressed
Local best path is 2,3,8,7

Update to 1 of 2,3,4,5,6,7 suppressed
Local best path is 1,3,8,7



PED Tweaking

- Do **not** suppress the longer path advertisement to the best path eBGP peer
- This should prevent the formation of transient loops during the suppression interval

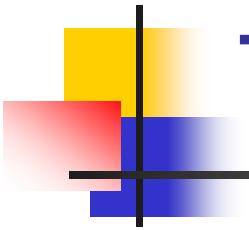


Conclusions

- Much of the background load in BGP is in processing non-informative intermediate states caused by BGP Path Exploration
- Existing approaches to suppress this processing load are too coarse to be completely effective
- Some significant leverage in further reducing BGP peak load rates can be obtained by applying a more selective algorithm to the MRAI approach in BGP, attempting to isolate Path Exploration updates by use of local heuristics

Potential Next Steps

- More data gathering
- Simulation of PED
- Code Development
- Field Testing and Measurements



Thank You

- Questions?