The ISP Column
An occasional column on things Internet

June 2007

Geoff Huston

Damping BGP

The topic of the scalability of the Internet's routing system has been one that has achieved elevated attention in recent months, and that's not just counting the five articles in this column in the past 12 months! One of its fascinating aspects is that the routing system appears to be one that has no natural constraints. Adding another prefix in the global routing table, or generating another protocol update, are actions that happen without any particular cost, penalty or constraint. So it's little wonder that from time to time we ponder whether the growth trends in routing exceed our ability to throw ever more capable technology at routing.

But of course this is not just a passing fad. The underlying issues with routing scalability are persistent themes that have been around for many years. These themes relate to the overall stability of the routing protocol and the performance of the packet forwarding subsystem when dealing with ever larger routing domains, finer granularity of information and ever denser levels of element interconnection.

The study of scalability of routing breaks into the classic subtopics of memory, processing and bandwidth. The larger the routing system grows the larger the required search space in order to perform the correct lookup operation for packet forwarding, inferring a requirement for larger memory for forwarding lookups. The combination of a larger routing space and a denser set of interconnections also increases the memory requirements for the operation of the routing protocol itself. The combination of the absolute size of the routing space and the level of dynamic update, or stability, have processing implications. The higher the level of dynamic changes in the routing state, then the greater the processing requirement that is imposed on each routing element. If the processing element can fall behind in processing the update load then there are implications in terms of the accuracy of the routing state, and potential implications in increased memory demands to hold the backlogged queues of unprocessed updates. And because routing is a distributed protocol, the greater the size of the routing space, the more densely interconnected the network and, more importantly, the greater the level of dynamic change, then the larger the amount of generated routing traffic and, by implication the higher the bandwidth demands for routing itself. So as the routing system grows so does the demands for additional memory processing capacity and bandwidth.

Of the three, bandwidth is perhaps the lesser of these scalability concerns, leaving memory and processing loads as consistent themes in studies of the routing system over many years.

One of the early observations relating to processing load was that it appears that a relatively small number of route objects were the subject of a large number of routing updates. The commonly observed behaviour at the time was that a route object would be announced, then withdrawn, then announced again, and so on for extended periods of time, and often at quite high frequency. A response to this type of load in the internet's inter-domain routing protocol, the Border Gateway Protocol (BGP) was the introduction of BGP Route Flap Damping. In this article I'd like to take a closer look at Route Flap Damping and see if there are other approaches to damping that may alleviate some aspects of BGP scaling pressure.

BGP Route Flap Damping

BGP Route Flap Damping (RFD) was first presented to the IETF BGP Working Group in October 1993, and documented as a Proposed Standard in November 1998 in RFC2439. BGP Route Flap Damping was seen as an effective approach to mitigating the effectives of link instability at the periphery of the network.

The motivation for this approach was the observation that many of the circuits used in the periphery of the network were analogue circuits over copper conductors, and such circuits often had a marginal failure mode where the circuit would be established, then lose synchronization, then re-establish itself, and so on. If such a circuit was carrying a BGP routing session then each time the circuit failed both ends of the circuit would effectively withdraw those routes that were being carried across the circuit. When the circuit was re-established, then the routes would be re-announced once more. An unstable circuit was capable of generating an endless sequence of withdrawals and announces. In itself this is perhaps a source of irritation rather than a major routing issue. However there is a second factor, namely that BGP supports a flat routing space, and routing announcements and withdrawals are propagated as far as possible across the entire routing space. If an address prefix was being announced from behind an unstable circuit, and generating a routing update once a second,

then the total update load imposed on the network is in effect the number BGP speakers in the entire network processing an update every second. As the network grows in size, then the total load imposed by such unstable circuits also grows in size, and, presumably, the number of unstable circuits would also grow in size, and the processing-based scaling properties of the routing system becomes a source of significant concern.

The objective of the RFD technique is to reduce the BGP processing load through the selective suppression of BGP updates that are considered to be reflective of persistent unstable network states. The concern that was being addressed in this approach was that a "BGP implementation may tend to fail due to high routing update volume." [RFC2439] This was seen as a cause of peer session failure, thereby further adding to the processing load imposed on other routers which "may cause failures in other instances of the same implementation or other implementations with a similar weakness." [RFC2439]

The general concept was that each time an address prefix was announced or withdrawn by a BGP peer a penalty score was applied to the prefix. If the penalty score then exceeded the damping threshold then this BGP peers' prefix advertisement was "damped" and not considered for selection as a local best path. While damped, further announcements or withdrawals of this prefix by the BGP peer continue to incur additional penalty scores. This penalty score decays over time using an exponential decay model, and when the penalty score decays below a re-announcement threshold, the prefix and associated path is once more considered as a candidate prefix for selection as a local best path.

The original intent of route flap damping was to damp down the effect of rapid routing oscillations by damping the noisy prefix for an extended period of time (between 30 minutes and 1 hour in a typical flap damping configuration), and the major anticipated benefit was to reduce the BGP update load on routers and mitigate the issue of router crashes.

It's a fine theory, but in practice over the past decade of scaling BGP, RFD has been largely ineffectual!

What Causes Route Instability?

Subsequent investigation of the dynamic characteristics of BGP has revealed that many of the route flap behaviours that could trigger a flap damping response were not generally caused by sustained link instability, or by any other persistent operational abnormality in the network's circuitry.

In the majority of cases it appears that the update sequences that trigger flap damping are the outcome of entirely normal operation of BGP. A route withdrawal causes a distance vector protocol like BGP to embark on a "path hunt" where the withdrawal will trigger a number of successive updates until either a stable path is found or the prefix is withdrawn (Fig 1).

This path hunt behaviour is exacerbated by variance in the Minimum Route Advertisement Interval (MRAI) Timer (the amount of time a BGP speaker will wait before passing on successive route updates for the same prefix) across various BGP implementations. This variance in timers causes withdrawals and updates to pass through different network paths at different speeds, generating a number of transient intermediate routing states until the routing state ultimately converges.

In RFD the successive updates associated with these intermediate state transitions each incur a flap damping penalty, so that a single initial event can generate an update sequence that can cause a route to the damped for an hour or more.

In the operational community RFD was seen as a measure that did little to remove potential harmful routing pathologies from the network, while at the same time RFD was a seen as a source of disruption to overall reachability in the network. Current operational notes on RFD discourage its use in large scale production networks that support the public Internet [RIPE-378].

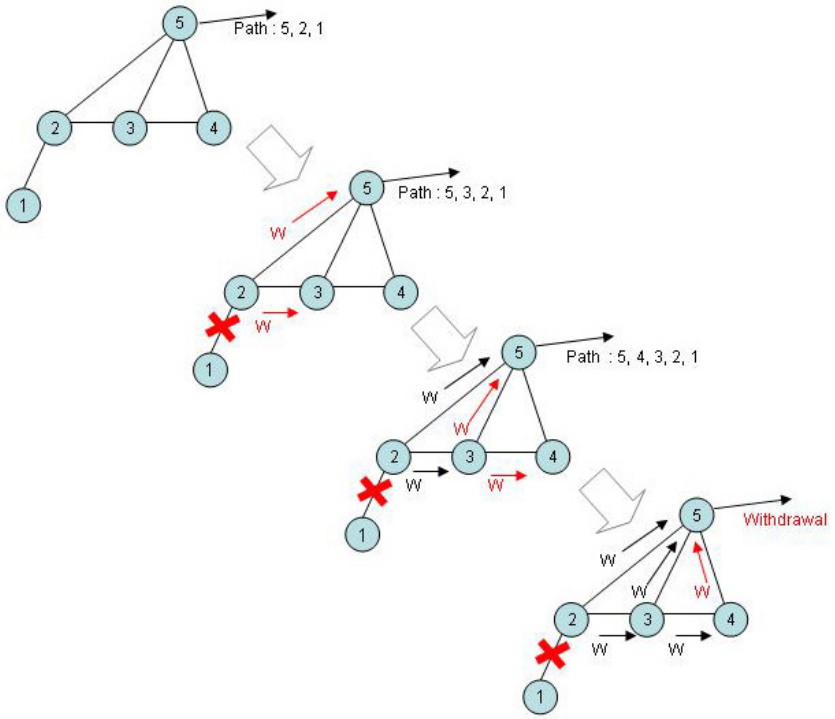

Fig 1 – BGP Path Hunting

BGP and Route Update Processing

Is turning off RFD harmful? What has changed in the past decade or more that alters the picture of BGP route updates and the impact of intense update load on routers?

A naive model of a BGP speaker is as an input queue of pending updates that have been received from a BGP peer that are yet to be processed, a BGP processor, and an output queue of updates that are to be sent to a BGP peer that reflect changes in the local best path state (Fig 2).

Fig 2 – BGP Processing Model

The concern that appeared to motivate the desire for RFD was that if a BGP speaker has many peers, then the input queues could grow at a faster rate than the local BGP processor could process them. For example, assuming that the BGP speakers all take the same amount of time to process a single update request, then a BGP speaker with three peers, all of whom are generating updates at their maximal processing rate, will receive an input stream of updates that represent a processing load three times the maximal processing rate for the local BGP instance, and queue buildup will be an inevitable outcome.

In such a model all updates on the input queue have already been acknowledged by the underlying BGP TCP transport protocol, so that if the local processor drops the update message prior to processing it, then the local routing state will have fallen out of synch with the BGP peer's routing state, and there is the distinct possibility of loop formation or black hole routing in the network's routing state. The implication is that with finite memory available for queues within the BGP engine, a high update rate causes the potential for loss of routing information.

This naïve model of BGP behaviour is not one that is in general use. A more commonly implemented approach is to effectively remove the input queue from the BGP model and couple the TCP transport state with the local processing rate, such that the BGP processor throttles the TCP session to prevent input queue buildup, thereby applying back pressure to the BGP peer holding the output queue. Normally this would just shift the problem of queue buildup and memory use pressure back to the BGP peer. However, the BGP peer may also apply update compression to its local output queue. This update compression can be seen as a queue management regime that removes preceding queued updates that refer to the same prefix as that being enqueued, so that at any point in time any given prefix occurs in the output queue at most once [Fig 3].

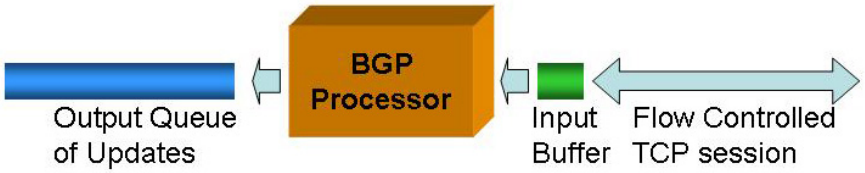

Fig 3 – TCP Rate-Controlled input to BGP

These measures can combine to significantly improve the dynamic behaviour of BGP under load. By coupling the transport flow control to the BGP processor throughput the local buildup of acknowledged but unprocessed updates, and the associated risk of queue drop through memory exhaustion, is avoided, and by performing output queue compression redundant BGP updates are removed before they are transmitted.

Within such an implementation model of BGP the precise nature of the underlying problem that RFD was intended to address becomes somewhat harder to define. As the BGP update rate increases local BGP speakers apply back pressure to their peers to slow the incoming update rate to match the local processing capability, avoiding local dropping of updates due to memory exhaustion. As the back pressure increases BGP speakers build up their output queues, at which point highly active prefixes trigger output queue compression.

This appears to further reinforce the perspective that BGP Route Flap Damping is of marginal value, and the extended outages triggered by a prefix damping event are more of a burden to the integrity of the network than the problem that flap damping was intended to alleviate.

Route Update Profile

So if the BGP problem space is shifting, then what is it shifting to? What do route updates look like these days? A recent study looked at the BGP updates over the month of April 2007 based on an observation point adjacent to AS 4637.

During that month a single eBGP peering session received 1,341,520 BGP update messages, reflecting 3,523,906 individual prefix updates and 627,538 individual prefix withdrawals. Considering that there were an average of some 215,000 individual prefixes in the BGP routing table across the month, that's an average of around 19 updates for every prefix in the month, assuming a uniform distribution of updates across the entire routing domain.

Of course, the distribution of updates is not uniform, and most of the network is highly stable. Half of these 210,000 prefixes had less than 10 routing updates across the month, and only 20,000 prefixes had more than 40 updates for the month. In other words, this is a very skewed distribution, with 10% of announced prefixes being responsible for 53% of all routing updates, and the busiest 1% of prefixes responsible for 24% of the routing updates for the month [Fig4]

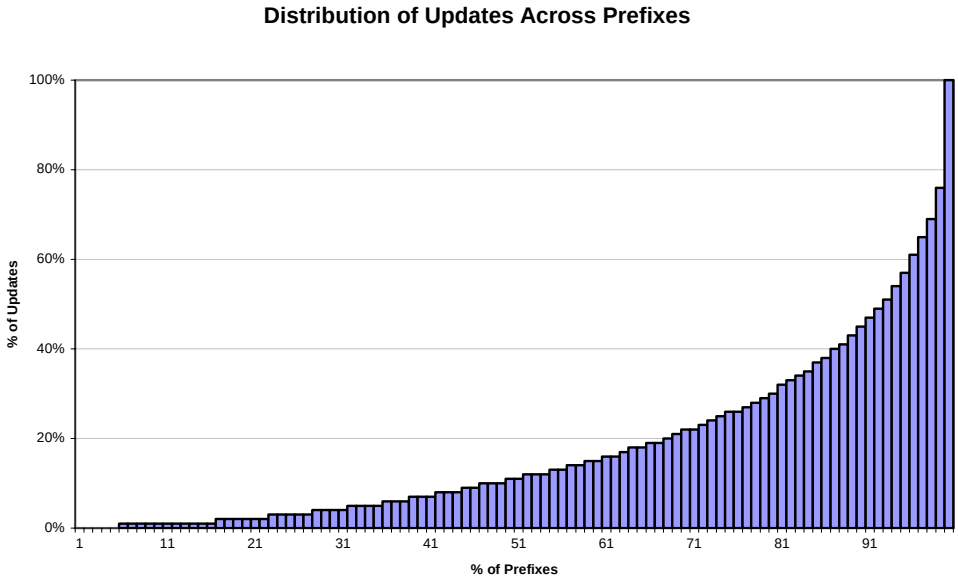

Figure 4 – Update Distribution

On the face of it, this represents a pretty compelling argument for some form of selective route damping. It can't be the case that more than a million routing updates actually reflect true underlying changes in topology of the network, given that these one million updates only refer to 2,000 prefixes. It would certainly be intuitive to guess that the overall majority of these routing updates represent intermediate routing states that are heading towards converged routing outcomes. Perhaps a deeper inspection of what is behind these busy prefixes is called for in order to understand what form of route flap damping would be effective here.

So what is the "typical" profile of BGP updates? Well "typical" is a tough question, but here I'd like to report on the profile of what happened in April 2007, BGP Update-wise, in my corner of the network.

The first step is to look at what kinds of updates one can expect from a single peer in BGP. The following table classifies the types of BGP updates of interest here.

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
WW	Withdrawal of a withdrawn prefix (possible BGP error or a data collection error)

So what did BGP look like for the April 2007 data? The following table classifies all the updates according to this taxonomy.

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
WW	0

The interesting numbers here are those associated with BGP path hunting following a withdrawal, which are likely to be associated with the 607,093 AA+ updates and the 627,538 AW updates. But the population of these update types alone is not enough on its own to justify a conclusion that over 1.2 million updates are associated with path hunting as a precursor to prefix withdrawal events. The other salient factor that needs to be examined is the time distribution of updates, as the path hunting condition is associated with a rapid burst of updates.

In looking at the time distribution of updates for the same prefix, ([Fig-5], [Fig-6] and [Fig-7]) there are some prominent peaks. The operation of the 30 second MRAI timer appears to be very prominent, and 934,391 updates occurred precisely 30 seconds after the previous update for the same prefix, and a total 1,636,093 updates for the same prefix occurred within 31 seconds of the previous update. That's the equivalent to 39% of the entire BGP update activity for the month. There are further local peaks at 30 second intervals at 60, 90 and all the way through to 240 seconds. Almost half of the BGP update activity occurs in a closely coupled manner [Fig-5].

There are also smaller local peaks at 30 minute and 1 hour intervals [Fig-6]. It is likely that these correspond to Route Flap Damping outcomes, where the damping period is typically one of these two values. Interestingly, there are also local peaks at 1, 2 and 3 day intervals [Fig-7]. This is unlikely to be an artefact of Route Flap Damping, and is more likely to be the outcome of some form of time-managed traffic system that performs routing changes on a regular 24 hour basis.

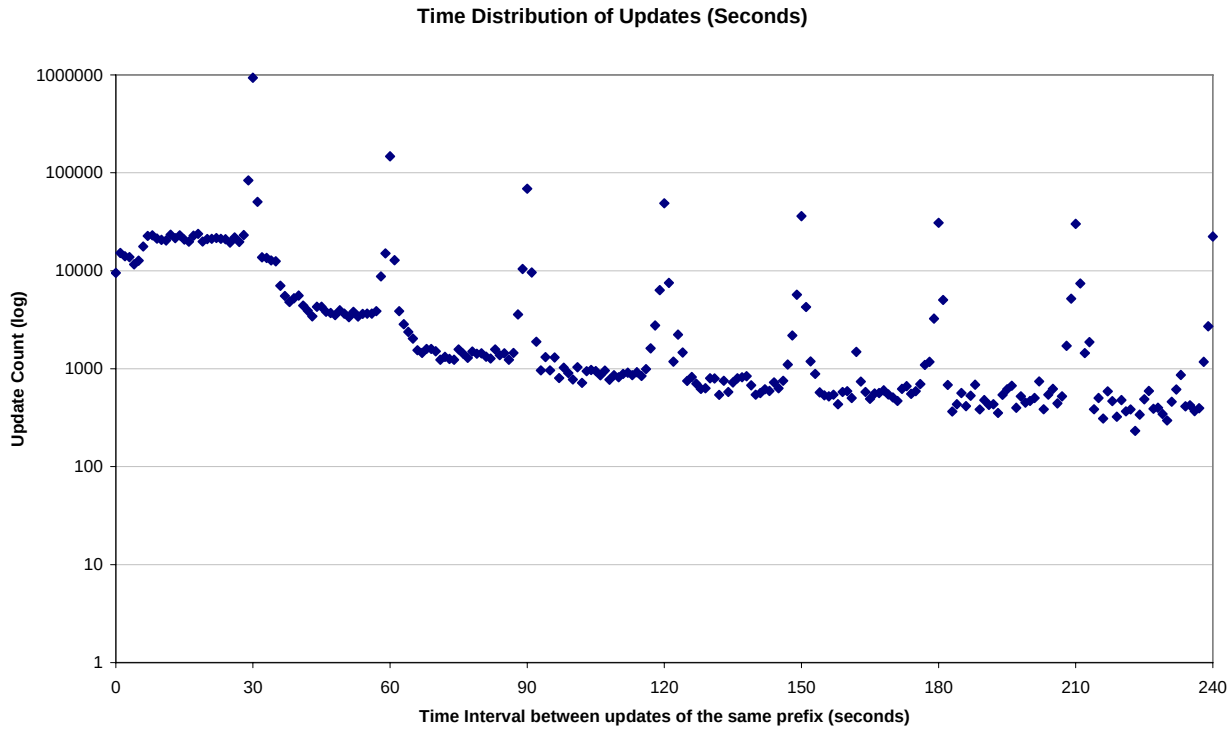

Fig 5 – Time Distribution of Updates (Seconds)

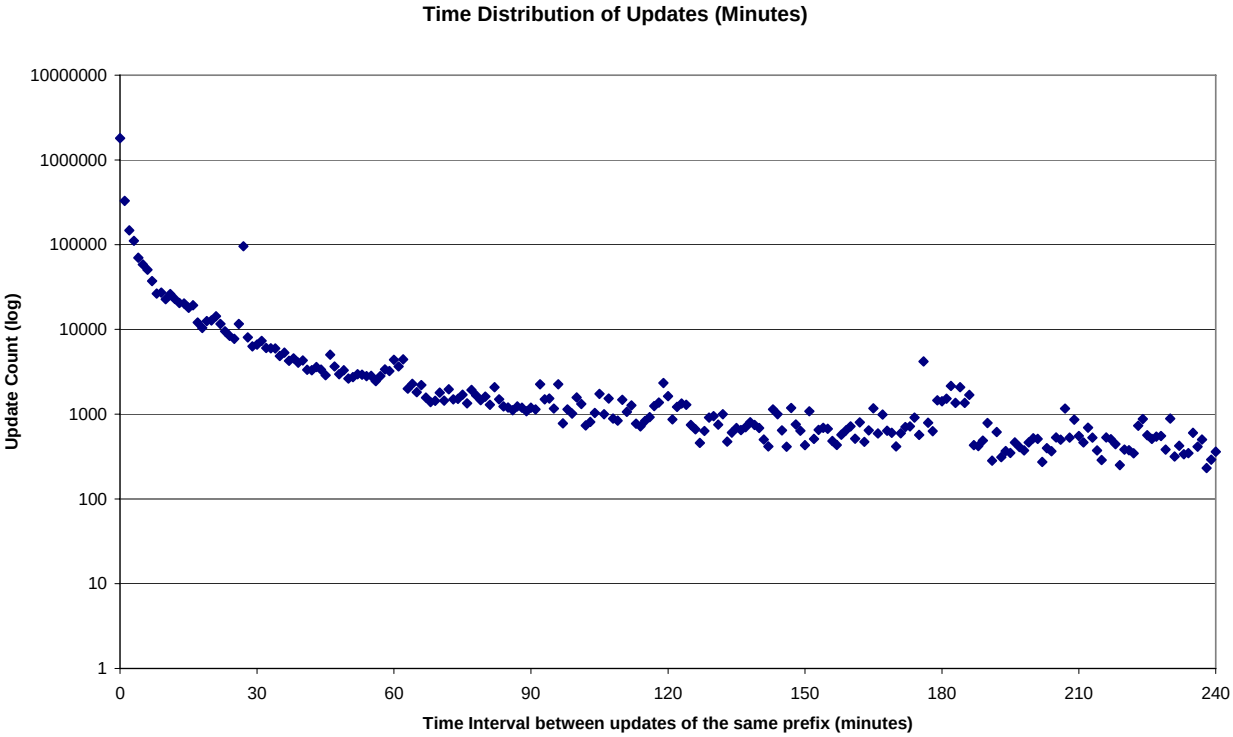

Fig 6 – Time Distribution of Updates (Minutes)

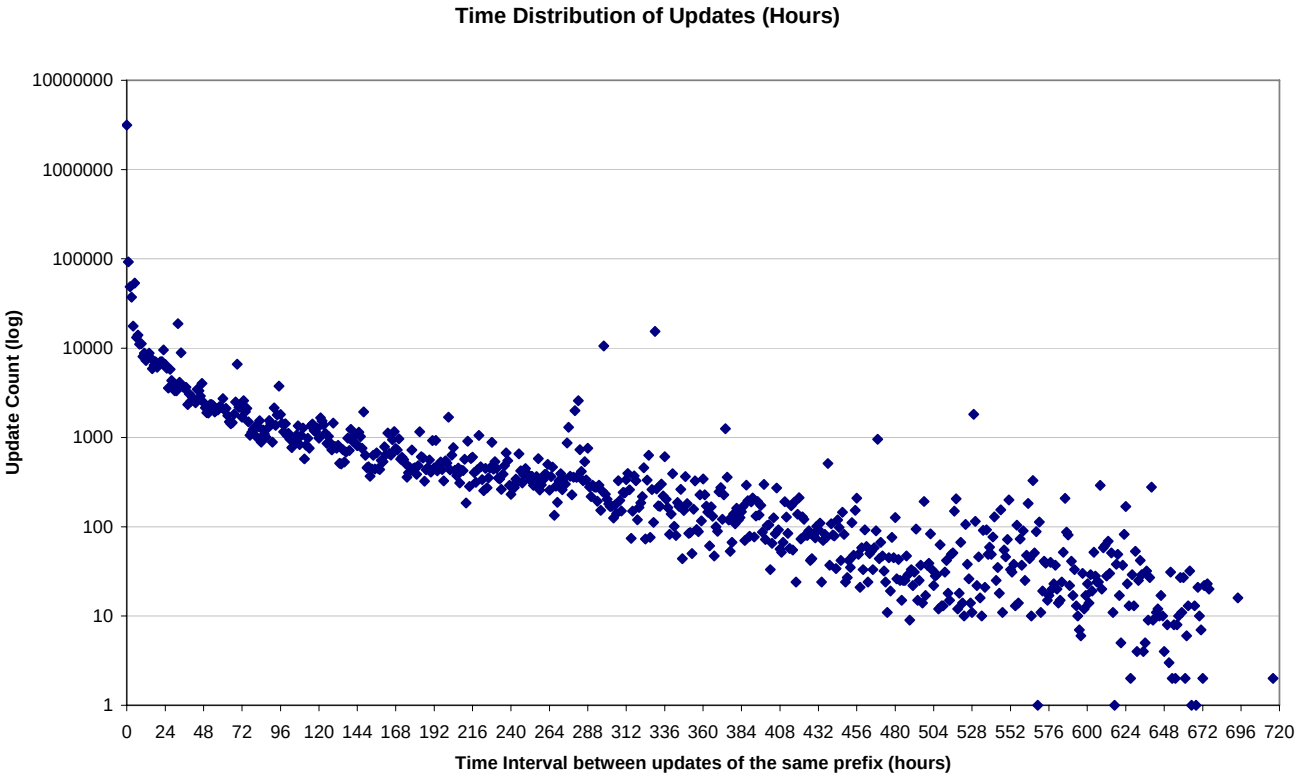

Fig 7 – Time Distribution of Updates (Hours)

Another way of looking at this time distribution of updates is to construct update "sequences" where a pair of updates is considered to be part of the same sequence if it refers to the same prefix and is received within 65 seconds (or slightly longer than two MRAI Timer intervals) of any previous update for the same prefix. Only 28% of the updates for the month are not part of any sequence, while 26% of updates are part of a coupled update pair, and 46% of updates are part of sequences of 3 or more updates. Interestingly enough, changing the timer as to what defines a sequence does not alter the profile greatly. Extending the sequence timer to 125 seconds (or four MRAI Timer intervals) produces the outcome that 54% of updates are part of sequences of 3 or more updates, while reducing the sequence timer to 35 seconds produces the outcome that 36% of all updates are part of sequences of 3 or more updates [Fig-8].

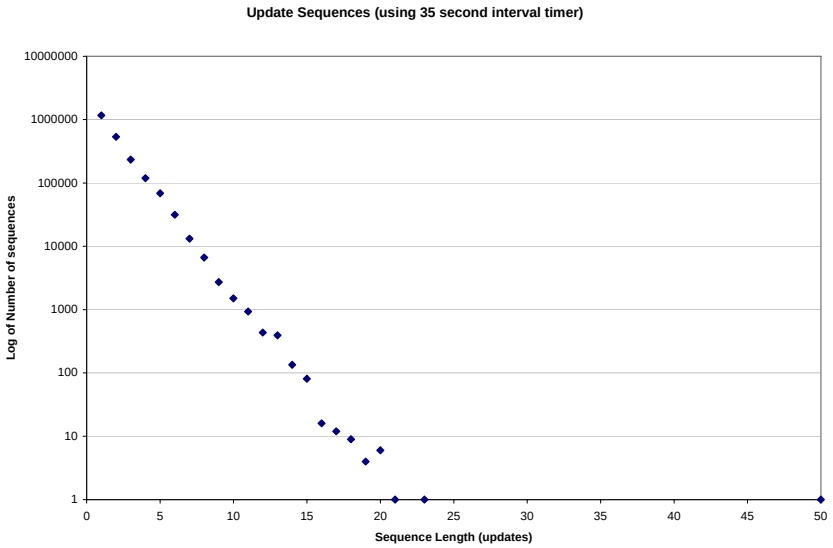

Fig 8 – Update Sequence Length Distribution (using 35 second interval timer)

Update Damping

The approaches to flap damping to date have tended to look at flap damping as a persistent condition that lasts for hours or longer, and are an outcome of a strongly persistent announcement and withdrawal characteristics that are assumed to be associated with some form of cyclical behaviour of an underlying circuit. This now appears to be well wide of the mark in terms of capturing the profile of what appears to be redundant BGP updates that reflect transitory routing states that are not in any converged state.

The question this prompts is whether there is any value in looking at BGP update patterns in the micro view rather than the macro? Can we identify, on the fly, update sequences that are highly likely to correlate to the BGP behaviour of path hunting to a withdrawal and damp the intermediate path hunting states and simply propagate the resultant converged state? This was part of the intent of the MRAI timer, but rather than simply apply a uniform damping interval to update propagation, can we devise a selective algorithm that attempt to pinpoint routing transitions that are strongly associated with BGP path hunting?

There are a number of observations here that appear to point to some value in considering this approach:

- A BGP update generator may perform "update compression" by removing an already queued update when a further update that refers to the same prefix is generated. Thus, when using the MRAI timer, only the most recent state for each prefix is passed to the BGP peers, and any intermediate state that occurred during the MRAI-imposed quiescent time is suppressed.

- Convergence in BGP appears to take longer than a single MRAI timer interval. As noted in the sequencing profile for updates, some 36% of all sequences take more than two MRAI timer intervals, or more than 60 seconds, to complete.

- Path hunting in BGP is commonly represented as an update sequence of the form {AA+}* AW, i.e. a sequence of lengthening AS path lengths followed by a withdrawal.

- Suppression of updates that lengthen the AS Path length of a prefix does not implicitly create any risks of routing loop formation during the suppression period. If the peer had already selected a different path as the best path, then the update to a longer path would have no impact on the previous selection. If the peer was using the path advertised by this BGP speaker as its best path, then the suppression may cause the peer to continue to use this out-of-date path, but would not cause a path loop, as if the peer was listed on the longer path then the peer would already have a shorter path, and this update would not alter the peer's forwarding state.

So the profile of update sequences that appear to be effective targets for some form of local suppression are those that lengthen the AS Path, and possibly also those updates that do not change the AS Path Length, and are also part of a sequence of time-clustered updates for the same prefix.

One approach to Update Damping is to delay the processing an update if the update represents a lengthening of the AS Path for an already announced prefix, selecting the AA+ updates. Furthermore, the updates that are of interest here are those that occur during BGP path hunting, so the length of time of the suppression should not be minutes or hours, but slightly over one MRAI time interval, or 35 seconds. If no further updates for this prefix are generated in this suppression interval, then the update is processed at the end of the suppression time, otherwise the suppressed update is replaced by its successor update.

How effective would this form of Update Damping be in the context of the BGP Update data set we've been examining here?

The algorithm used to implement this damping response is to suppress the processing all AA+ updates by up to 35 seconds. If a further update for this prefix occurs during this suppression interval, then the suppressed update is ignored and the successor update is processed in stead. If this update represents a further lengthening of the AS Path then it, too, is suppressed for 35 seconds. There are 607,093 AA+ updates in this set of suppressed updates, or some 15% of the total update load for the month. Update Damping would result in not processing 371,943 updates, or some 9.5% of the total update load. This result also indicates that 61% of all AA+ updates are followed by a subsequent update for the same prefix within one MRAI time interval.

Decreasing the sensitivity of the suppression parameters to a little over 2 MRAI intervals, or 65 seconds, increases the number of unprocessed updates to 418,805, or an additional 1%, so it appears that a damping

sensitivity of a single MRAI interval represents a suitable point of compromise between maximising the number of damped BGP events without making the BGP convergence process significantly slower.

Of these damped updates, how many are actually path hunt updates? Some 96,135 of these damped updates are immediately followed by a withdrawal within the 35 second suppression period, and a further 36,691 damped updates are followed by another suppressed AA+ update.

This approach could be extended in a number of directions. One approach is to regard any update that does not reduce the AS Path length or withdraw the prefix as being a candidate for update damping. In this case some 845,290 updates would be damped or 21% of all updates. Of these update just under one quarter, or 208,007 of these damped updates are followed by a withdrawal within one MRAI interval, and a further 474,234 of these damped updates are followed by an update with an AS path of the same or greater length. The implication being that update damping removes around one fifth of the total BGP update volume without reducing the time to convergence for route withdrawal, nor the time for propagation of more preferred routing paths.

Using this latter form of Update Damping, over the month the average prefix update rate per second falls from 1.60 prefix updates per second to 1.22 prefix updates per second [Fig-9], with 0.38 damped updates per second on average [Fig-10].

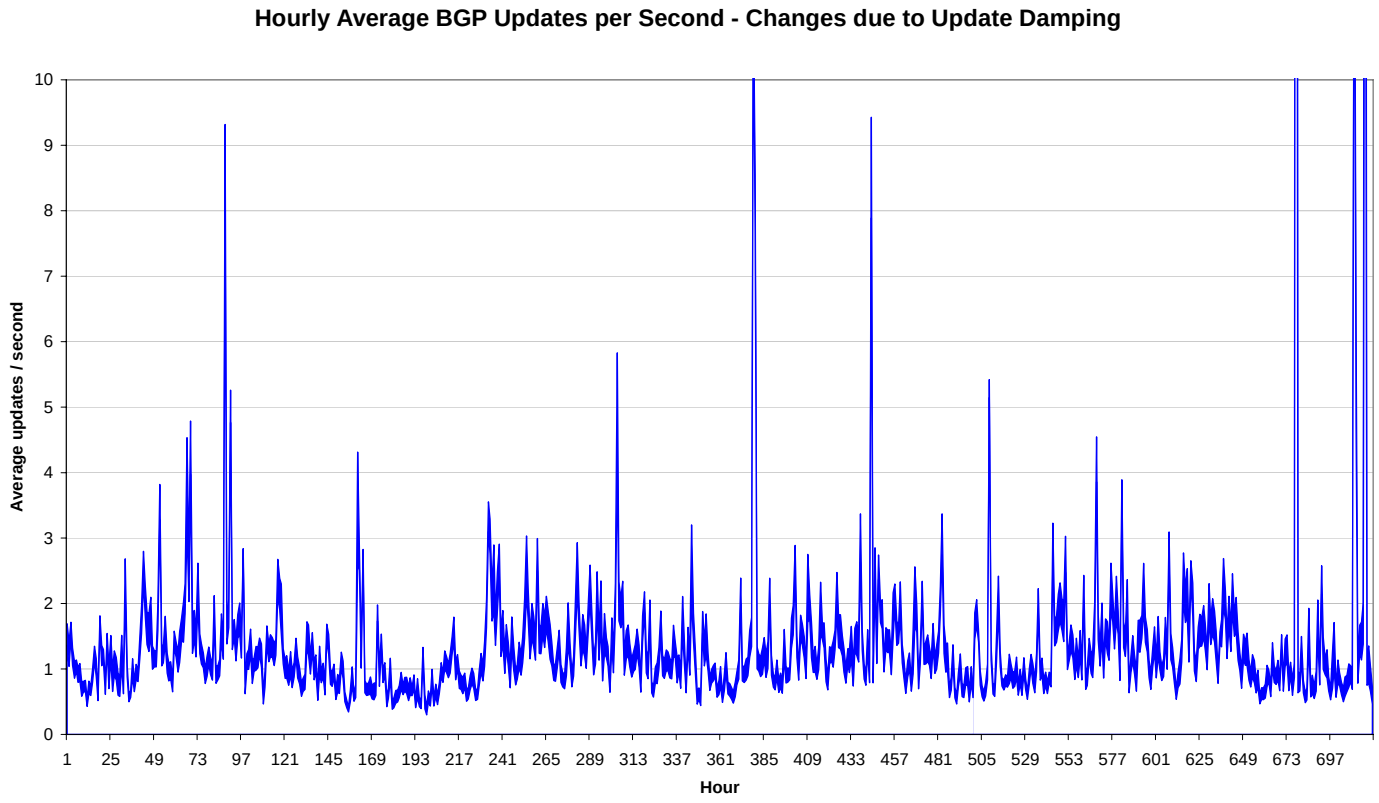

Fig 9 – Average Updates per second – Difference between undamped and damped

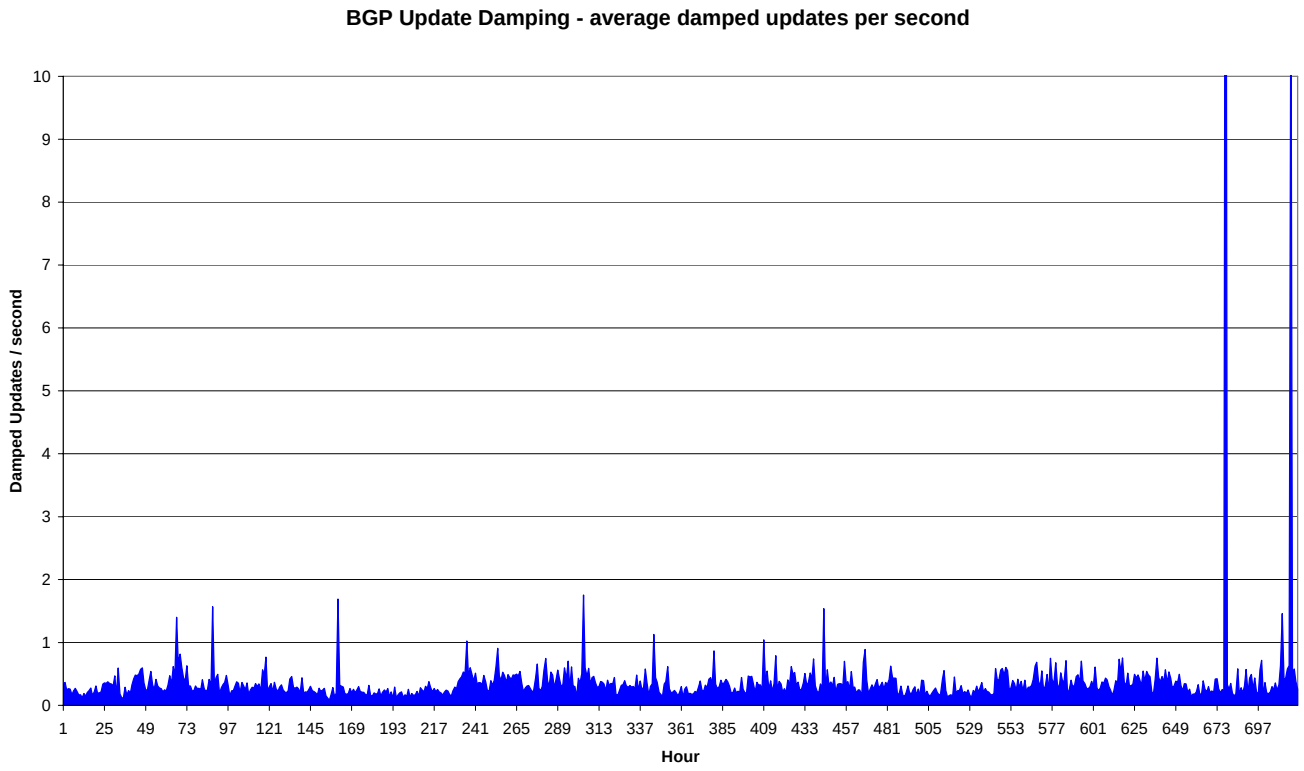

Fig 10 – Average Damped Updates per second

The average peak update rate per hour falls from 355 to 290 prefix updates per second using Update Damping [Fig-11], an average reduction of 65 updates per second on the hourly peaks [Fig-12].

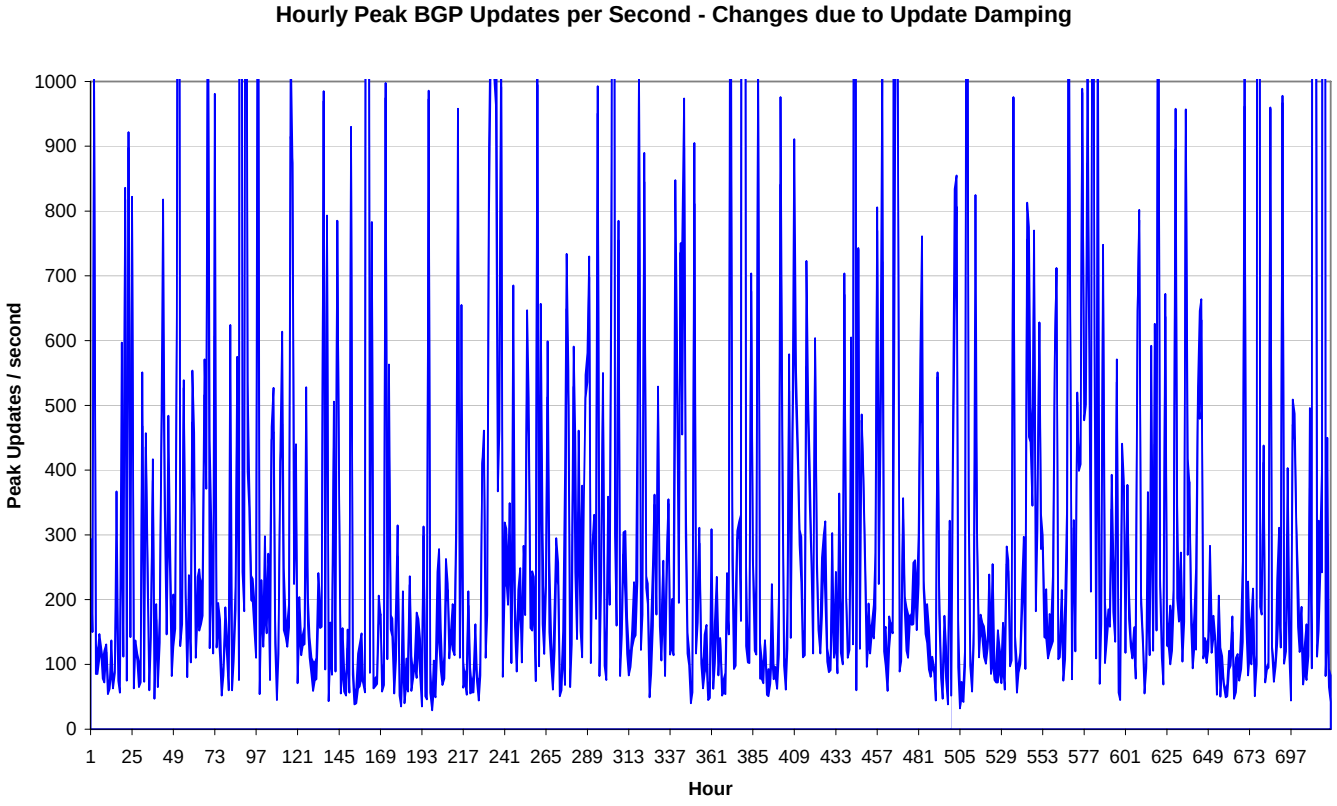

Fig11 – Hourly Peak Updates per second – Difference between undamped and damped

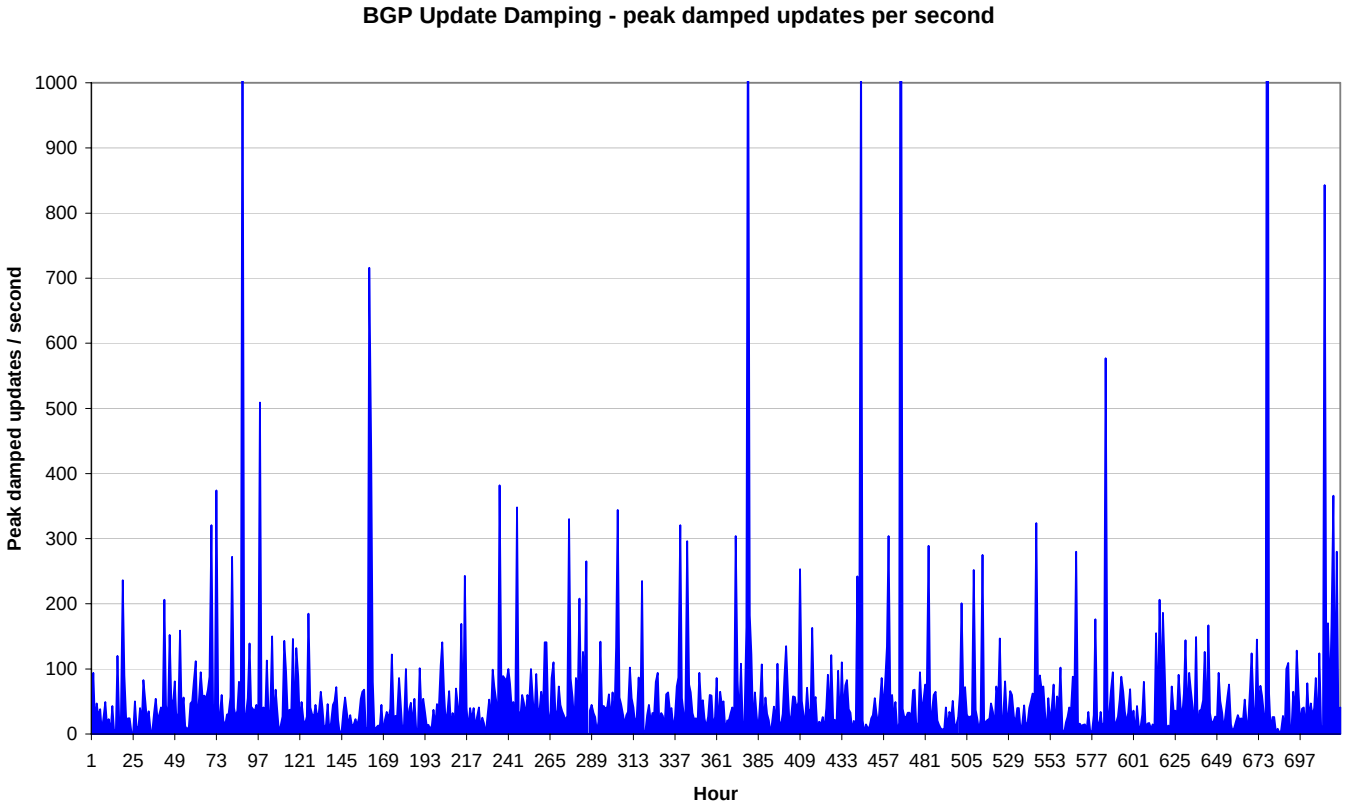

Fig 12 – Hourly Peak Damped Updates per second

Some Observations

Of course this is not the first, and will by no means be the last, suggestion on how to make marginal improvements in BGP. However, it's the process that lies behind making this suggestion is the major topic of interest here, where the steps are to gather relevant data, examine the data for properties, patterns and relationships, and then test various hypotheses about how these patterns may be exploited by testing the data. If it still looks plausible then the subsequent exercise is about gathering more data, performing more analysis, simulations and then possibly undertaking some experimentation.

On a more general note, this work highlights an important aspect of BGP behaviour. The original assumption with the deployment of BGP was that external events exacerbated potential problems in the routing protocol, and the most extreme was persistent instability in the network that caused continual load to be imposed on the routing system. Route Flap Damping was a response to alter the behaviour of BGP to cope with pathologically broken network conditions. Further experience with BGP now reveals a somewhat different picture, where it is the operation of the protocol itself that causes relatively simple single root cause events to be amplified into extended event sequences across the entire network. This is a by-product of the very nature of BGP as a distance vector protocol, and is further amplified by the divergence in propagation times of events across the network due to the differences in processing capability, bandwidth, relative load and differences in the implementation of MRAI timers across various BGP implementations.

What it tends to suggest is that one of the most productive efforts to reduce the processing loads associated with BGP are efforts directed at attempting to protect BGP from the worst aspects of its own behaviour!

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

About the Author

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001.

http://www.potaroo.net