The ISP Column
An occasional column on things Internet

BGP Wedgies

September 2004

It has commonly been assumed that the Border Gateway Protocol (BGP) is a tool for distributing reachability information in a manner that creates forwarding paths in a deterministic manner. What goes in, in the form of configuration of BGP-speaking router, comes out, in the form of inter-domain traffic flows, and determinism in this context implies that there is a predictable correlation between a configuration and a traffic outcome.

Tim Griffin, a researcher with Intel Cambridge has been looking at a class of BGP configurations for which there is more than one potential outcome that BGP may select, where forwarding states other than the intended state are equally stable, and that the stable state selected by BGP may be selected in a non-deterministic manner. It is also possible to reach one of these states where any single Network Administrator does not have enough information at hand from the local network to determine how to get BGP to re-converge back to a stable state that was the intended state. To the local administrator it would appear that BGP has wedged in a manner that cannot be repaired. These stable, but unintended, BGP states are termed here "BGP Wedgies". In this article I'd like to explore Tim's work and look at its implications for inter-domain routing.

Describing BGP Routing Policy

BGP routing policies generally reflect each network administrator's objective to optimize their position with respect to their network's ***cost***, ***performance*** and ***reliability***.

With respect to ***cost*** optimization, the local network's default routing policy often reflects a local preference for routes learned from a customer over routes learned from some form of peering exchange. In the same vein the local network is often configured to prefer routes learned from a peer or a customer over those learned from a directly connected upstream transit provider. In other words, a provider would normally prefer to pass traffic to a paying customer over passing the same traffic to a paid transit provider. These preferences may be expressed via a local preference configuration setting, where the local preference overrides the AS path length metric of the base BGP operation.

In terms of engineering ***reliability*** in the inter-domain routing environment it's often the case that a service provider may enter into arrangements with two or more upstream transit providers, passing routes to both providers , and receiving traffic from both sources. If the path to one upstream fails the traffic will switch to other links, and once the path is recovered, the traffic should switch back.

Achieving outcomes that offer consistent ***performance*** in such situations of multiple upstream providers it is also common to place a relative preference on the providers, so that the connection that offers an optimal price and performance outcome is regarded as a preferred, or "primary" connection, and other connections are regarded as less preferred, or "backup" connections. The intent is typically that the backup connections will be used for traffic only for the duration of a failure in the primary connection.

It is possible to attempt to express this primary / backup policy using local AS path prepending, where the AS path is artificially lengthened towards the backup providers, using additional instances of the local AS. However, this is not a very reliable form of expressing policy preferences. For example, the selected primary provider may in turn be using AS path prepending to its backup upstream provider, and in certain cases the path through the local backup-only provider may still be selected as the shortest AS path length, and therefore be preferred over the primary link.

An alternative approach to routing policy specification uses **BGP Communities**. In this case the provider publishes a set of community values that allows the client to select the provider's **local preference**. The client can use a community to mark a route as "backup only" towards the backup provider, and "primary preferred' to the primary provider. In this case the local preference overrides the AS path length metric, so that if the route is marked "backup only", the route will be selected only when there is no other source of the route.

BGP Wedgies

The richness of local policy expression through the use of communities, when coupled with the behaviour of a distance vector protocol like BGP, leads to the observation that certain BGP configurations have more than one "solution", or more than one stable BGP state. An example of such a situation is indicated in Figure 1.

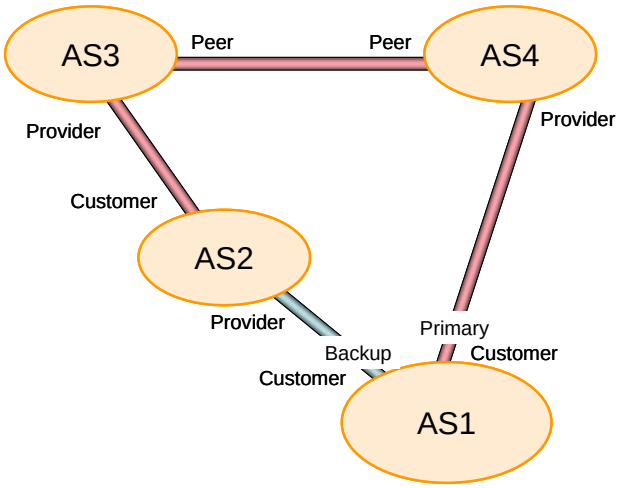

Figure 1

In this case AS1 has marked its advertisement of prefixes to AS2 as "backup only", and its advertisement of prefixes to AS4 as "primary". AS3 will hear AS4's advertisement across the peering link, and pick of AS1's prefixes with the path "AS4, AS1". AS3 will advertise this to AS2. AS2 will hear two paths to AS1, the first is by the direct connection to AS1, and the second is via the path "AS3, AS4, AS1". AS2 will prefer the longer path as the directly connected routes are marked "backup only", and AS2's local preference decision will prefer the AS3 advertisement over the AS1 advertisement. This is the intended outcome of AS1's policy settings, where no traffic passes from AS2 to AS1, and AS2, reaches AS1 via a path that transits AS3 and AS4. (Figure 2)

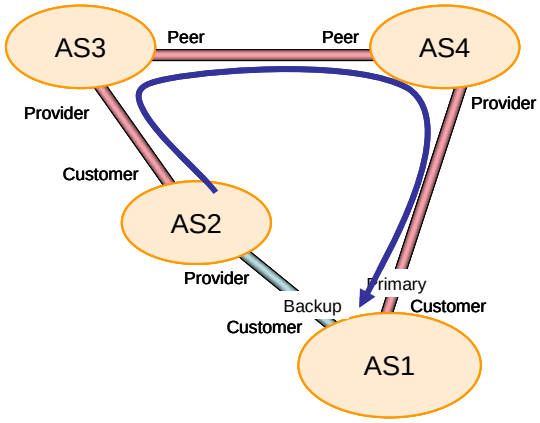

Figure 2

If the AS1 AS4 path is broken, AS4 will withdraw its advertisement of AS1's routes toAS3, who, in turn will send a withdrawal to AS2. As2, will then select the backup path to AS1. AS2 will advertise this path to AS3, and AS3 will advertise this path to AS4. Again, this is part of the intended operation of the primary / backup policy setting. (Figure 3)

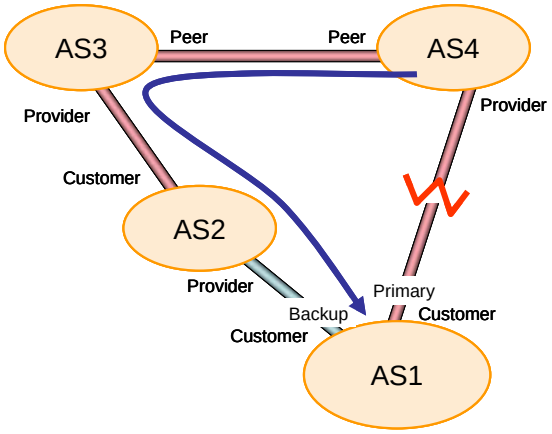

Figure 3

When connectivity between AS4 and AS1 is restored the BGP state will not revert to the original state. AS4 will learn the primary path to AS1, and readvertise this to AS3 using the path "AS4, AS1". AS3, using a default preference of preferring customer-advertised routes over peer routes will continue to prefer the "AS2, AS1" path. AS3 will not pass any updates to AS2. After the restoration of the circuit traffic from AS3 to AS1 and from AS2 to AS1 will be presented to AS1 via the backup path, even through the primary path via AS4 is in service. (Figure 4)

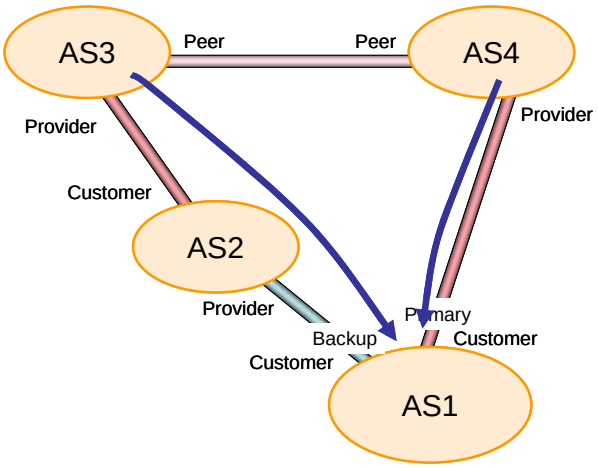

Figure 4

The intended forwarding state can only be restored by AS1 deliberately bringing down its eBGP session with AS2, even though it is carrying traffic. This will cause the BGP state to revert to the intended configuration and then the AS1 – AS2 link can be restored.

It is often the case that an AS will attempt to balance incoming traffic across multiple providers, again using the primary / backup mechanism. For some prefixes one link is configured as the primary link, and the others as the backup link, while for other prefixes another link is selected as the primary link. An example is shown in Figure 5.

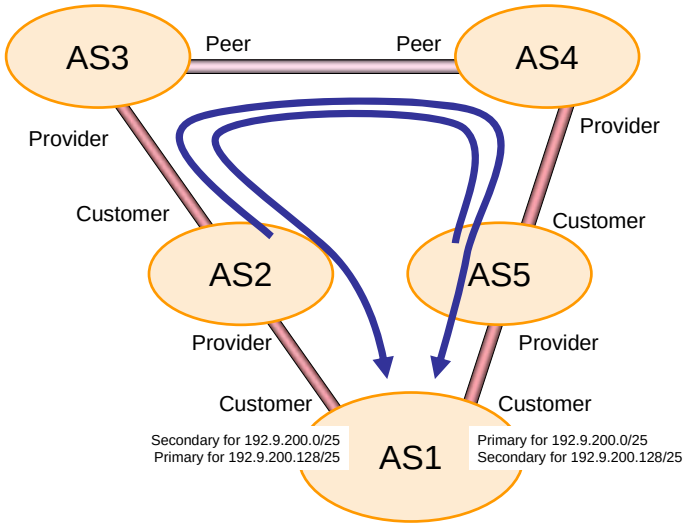

Figure 5

The intended configuration has all incoming traffic for addresses in the range 192.9.200.0/25 via the link from AS5, and all incoming traffic for addresses in the range 192.9.200.128/25 from AS2.

In this case if the link between AS3 and AS4 is reset, AS3 will learn both routes from AS2, and AS4 will learn both routes from AS5. As these customer routes are preferred over peer routes, when the link between AS3 and AS4 is restored, neither AS will alter its routing behaviour with respect to AS1's routes. This situation is now wedged, in that there is no eBGP peering that can be reset that will flip BGP back to the intended state. It is also the case that there has been no visible disruptions to the eBGP sessions directly connected to AS1, so from AS1's perspective nothing local has changed to cause this BGP state change. This is an instance of a BGP Wedgie.

The necessary corrective action is that AS1 has to withdraw the backup advertisements on both paths and then operate for an interval without backup, and then readvertise the backup prefixes. The length of the interval cannot be readily determined in advance, as it has to be sufficiently long so as to allow AS2 and AS5 to learn of an alternate path to AS1. At this stage the backup routes can be readvertised.

Multi-Party BGP Wedgies

This situation can be more complex when three or more parties provide upstream transit services to an AS. An example is indicated in Figure 6

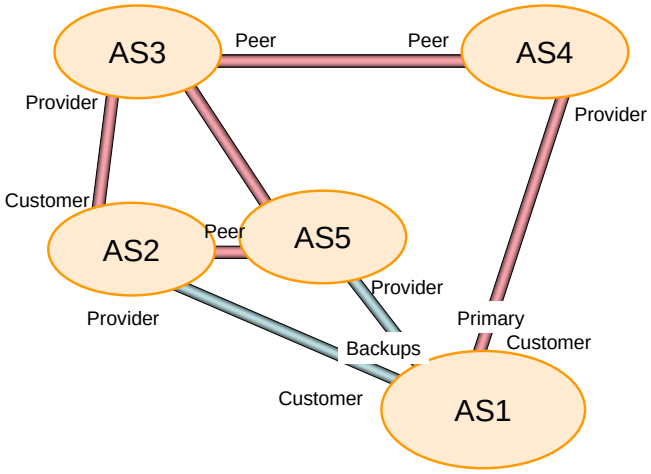

Figure 6.

In this example the intended state is that AS2 and AS5 are both backup providers, and AS4 is the primary provider. When the link between AS1 and AS4 breaks and is subsequently restored, AS3 will continue to direct traffic to AS1 via AS2 or AS5. In this case a single reset of the link between AS2 and AS1 will not restore the original intended BGP state, as the BGP-selected best route to AS1 will switch to AS5, and AS2 and AS3 will learn a path to AS1 via AS5. (Figure 7)

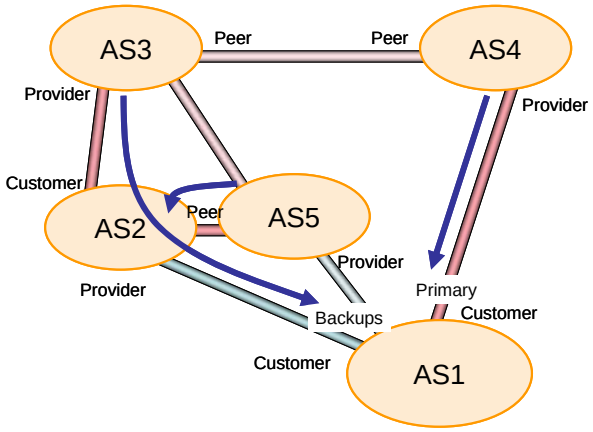

Figure 7

What AS1 is observing is incoming traffic on the backup link from AS2. Resetting this connection will not restore traffic back to the primary path, but instead will switch incoming traffic over to AS5. The action required to correct the situation is to simultaneously reset both the link to AS2, and also the link to AS5. This is not necessarily an intuitive solution, as at any point on time only one of these links will be carrying backup traffic, yet both BGP sessions need to be brought down at the same time in order to commence restoration of the intended primary and backup state.

BGP and Determinism

BGP does not behave deterministically in all cases, and, as a consequence, there is intended and unintended element of non-determinism in BGP. For example, the default final tie break in some implementations of BGP is to prefer the longest-lived route. To achieve determinism in this last step it would be necessary to use a comparison operator that has a predictable outcome, such as a comparison of router identifiers. This class of non-deterministic behaviour is termed here "intended" non-determinism, in that the policy interactions are, to some extent, predictable by network administrators.

BGP is also able to generate outcomes that can be described as "unintended non-determinism" that can result from unexpected policy interactions. These outcomes do not represent mis-configuration in the standard sense, since all policies may look completely rational locally, but their interaction across multiple routing entities can cause unintended outcomes, and BGP may reach a state that includes such unintended outcomes in a non-deterministic manner.

Unintended non-determinism in BGP would not be so bad if all stable routings were guaranteed to be consistent with the policy writer's intent. However, this is not always the case. The above examples indicate that the operation of BGP allows multiple stable states to exist from a single configuration state, where some of these states are not consistent with the policy writer's intent. These particular examples can be described as a form of "route pinning", where the route is pinned to a non-preferred path.

The challenge for the network administrator is to ensure that an intended state is maintained. Under certain circumstances this can only be achieved by deliberate service disruption, involving the withdrawal of routes being used to forward traffic, and re-advertising routes in a certain sequence in order to induce an intended BGP state. However, the knowledge that is required by any single network operator administrator in order to understand the reason why BGP has stabilized to an unintended state requires BGP policy configuration knowledge of remote networks. In effect there is insufficient local information for any single network administrator to correctly identify the root cause of the unintended BGP state, nor is there sufficient information to allow any single network administrator to undertake a sequence of steps to rectify the situation back to the intended routing state.

We should expect that the density of interconnection will increase, and also that the capability for policy-based preference setting of learned and re-advertised routes will become more expressive. We should therefore expect that the incidence of unintended BGP states will increase, and the ability to understand the necessary sequence of route withdrawals and re-advertisements will become more challenging to determine in advance.

Whether this could lead to eBGP routing system reaching a point where each network consistently cannot direct traffic in a deterministic manner is at this stage a matter of speculation. BGP Wedgies are an illustration that a sufficiently complex interconnection topology, coupled with a sufficiently expressive set of policy constructs, can lead to a number of stable BGP states, rather than a single intended state. As the topology complexity increases it is not possible to deterministically predict which state the BGP routing system may converge to. Paradoxically, the demands of inter-domain traffic engineering appear to require both greater levels of expressive capability in policy-based routing directives, operating across denser interconnectivity topologies in a deterministic manner. This may not be a sustainable outcome in eBGP-based routing systems.

Tim Griffin *(http://www.cambridge.intel-research.net/~tgriffin/)*

Geoff Huston *(http://www.potaroo.net)*

September2004

Disclaimer

The above views do not represent the views of the Internet Society. They were possibly the opinions of the authors at the time of writing this article, but things always change, including the authors' opinions!