1 Interdomain Routing

1.1 ISP structure and economics

Structurally, the Internet is a network of interconnected networks. Each network is called an autonomous system (AS). An AS is a network of routers managed by a single administrative entity. We will use the terms AS and Internet service provider (ISP) interchangeably although some large ISPs may be divided into multiple ASes for ease of internal administration. The ASes may either peer privately or through a Internet Exchange Point (IXP), also sometimes referred to as a network access point or NAP, where several ASes interconnect.

ASes are organized as a tiered hierarchy. Tier-1 ASes are at the top level of this hierarchy and provide transit service typically to tier-2 ASes that in turn provide transit typically to tier-3 ASes and so on. As the bottom of this hierarchy are stub ASes that provide connectivity to end users. Tier-1 ASes are large and it is estimated that there are only roughly ten or so of them in the Internet today. The AS structure is not a strict hierarchy, i.e., there are interconnections between ASes at the same tier. A tier-3 AS may also obtain transit service directly from a tier-1 AS. In fact, the tier to which an AS belongs is not a piece of globally agreed upon public information, so thinking in terms of tiers only loosely helps understand the nature of the Internet’s AS hierarchy. A better characterization of the AS hierarchy is in terms of the economic relationship between interconnected ASes.

An AS can be a provider, peer, or customer of another AS as determined by the economic relationship between them. A customer pays a provider for transit service. Typically, peer ASes typically do not charge each other for transiting traffic between their respective customers. The exact nature of the economic relationship between ASes is typically proprietary information and may evolve over time. Note that a customer always pays a provider irrespective of which direction (from/to the customer) the traffic flows. This aspect of Internet service pricing is fundamentally different from traditional long-distance telephone services where typically the initiator of a call pays for the call while the recipient does not.

The difference in the pricing schemes between Internet service and telephone service can be attributed to the nature of their traffic and resource allocation schemes. A telephone call has a call setup phase that clearly identifies the initiator and traffic flow in both directions is more or less balanced, so it suffices to charge just the initiator. Before caller identification became widespread, charging the initiator also avoided the scenario where a receiver of the call had to worry about the cost of the call without knowing its origin. In contrast, Internet traffic does not lend itself to an initiator-based pricing model. Consider a typical Web transaction where an end-user downloads content from a web site. One would be inclined to identify the end-user as the initiator of the connection, but most of the traffic flows from the web server to the end-user. Who
should pay for this traffic and how does the network identify this entity? Who is responsible for additional
TCP connections created by programs at either end? In the Internet, both parties—the end-user and the
web server—pay their respective ISPs. The network layer is stateless and does not attempt to account for
resource usage on a per-connection basis. Instead, economic settlement happens on a bilateral basis between
neighboring ASes based on the total volume of traffic exchanged each month.

How does the total volume of traffic exchanged determine the settlement cost charged by a provider to
a customer? There are several different pricing schemes commonly used in practice. The most familiar one
is the fixed fee model that is popular for residential broadband customers or more generally low volume
traffic exchange. In this model, the cost of connectivity is a fixed fee per billing period, e.g., $50 per month,
irrespective of actual bandwidth usage. ISPs typically set the price point based on the average bandwidth
utilization across all customers in that plan ignoring differences between high volume and low volume
users. Such plans also typically place a cap on the the instantaneous available bandwidth capacity as well as
sometimes the total amount of traffic exchanged during the billing period. Mobile data as well as some
residential broadband service providers may also resort to more creative tactics such as slowing down
“power users” if their total utilization during a billing period exceeds a threshold.

An alternative pricing scheme is one that charges based on the average utilization during the billing period.
One way to measure average utilization, especially for smaller customers as well as mobile data plans where
bandwidth is more expensive relative to wired plans, is to measure the total volume (in bytes) or traffic
exchanged during the billing period, a method that requires more accounting infrastructure in place on part
of the provider. An alternative is to employ a combination of sampling and windowing wherein the provider
divides the billing period into fixed size, e.g., 5-minute intervals, and measures the average bandwidth
utilized over all or a sampled subset of intervals over the course of the billing period.

Max- or 95th-percentile utilization pricing charges the customer based on peak instead of the average util-
ization. The peak utilization may be computed either as the maximum or a high percentile, e.g., the 95th
percentile, of the utilization as computed over all or a sampling of short fixed length intervals over the billing
period.

Committed information rate (CIR) pricing combines aspects of the fixed fee and utilization-dependent
schemes in that the customer is charged a baseline fixed cost irrespective of usage or for usage unto a
threshold plus an additional cost that increases with the actual utilization above the threshold. The latter
cost in turn may be computed based on average or peak utilization as above.

1.2 BGP overview

The border gateway protocol (BGP) is the Internet’s de facto interdomain routing protocol. BGP has two
components: eBGP or external BGP and iBGP or internal BGP. Routers at the border of an AS run eBGP
to announce and receive routes to/from border routers in neighboring ASes. All routers run iBGP to re-
announce routes learned via eBGP to internal routers and to rank and select routes based on the criteria
below.

Route announcements in BGP include the NLRI or network layer reachability information that identifies
the address prefix of the destination and a number of attributes such as (1) ORIGIN_CODE that identifies how
the prefix was learned by the router advertising the route, e.g., an origin code of IGP or EGP respectively
indicates that the router learned the route via an intradomain routing protocol or via BGP; (2) NEXT_HOP that
identifies the interface address of the next-hop router (furthest away from the destination prefix) along the
path; (3) AS_PATH, an ordered list ASes traversed along the path starting from the next_hop interface all the
way to the destination prefix; (4) local_p pref that indicates how much the AS prefers the path and is only used internally by routers in an AS (and not passed between ASes); (5) MED, or multiple exit discriminator that is used by an AS to tell a neighboring AS which of the multiple peering points should be used by the latter to send traffic to the former; and (6) communities, an optional transitive attribute that can be used to carry almost any information about a route between ASes and are typically tags that group routes sharing common characteristics that can not be described by other route-specific attributes. Communities are generally not directly used to determine policy or the best path to a destination but may be used to influence the values of other attributes that determine routing policy, e.g., no_export specifies that the route should not be advertised outside the local AS; no_advertise specifies that the route should not be re-advertised to peers of any router to which that route is advertised, etc.

Routes get transformed as they propagate through the network in three steps. First, when a route propagates from a node \( w \) to a neighboring node \( u \), node \( w \) applies its export policy to the route. Second, \( w \) applies a path vector transformation by adding itself to the route's AS_PATH, setting next_hop accordingly, and filtering the route if its AS_PATH contains \( u \). Finally, node \( u \) applies its import policy to the route. In particular, the import policy sets the local_p pref attribute, a value that is used locally to select the best route but may be stripped off (by setting it to a default value) before readvertising the route. These three steps in combination are referred to as a peering transformation.

Routers compute routes by ranking routes received from neighboring routers and choosing the best route. Interdomain concerns take precedence over intradomain concerns in this ranking procedure. The most important criterion is local preference indicated by the local_p pref attribute and the rules used to set local_p pref values are policy decisions implemented by operators. In particular, local_p pref values may have little to do with the path length, delay, available bandwidth, or other performance metrics, which is why interdomain routing is described as a policy routing protocol in contrast to shortest-path or other path-performance based routing protocols. For example, an AS commonly prefers to route via a customer over a peer or a provider, and a peer over a provider, in keeping with its economic objectives.

The second criterion is the length of the AS path. Although the length of the AS path is somewhat correlated with its performance, i.e., longer AS paths are more likely to have higher delays, the ordering of the first two criteria make it clear that policy takes precedence over any user-perceived path performance in BGP. The MED attribute is relevant only when there are multiple peering points between two ASes, which is typically the case between tier-1 or large ASes. The fourth criterion is to check if the route was learned from a router in a neighboring AS via eBGP or from an internal router via iBGP and to prefer the latter over the former. The fifth criterion is commonly referred to as hot potato or early-exit routing, which seeks to prefer routes that get packets out of the local AS quicker. Just like you would not like to hold a hot potato in your hands for long, ASes try to get packets out of their network quickly by picking a route with the lowest IGP cost to the egress router. The last criterion is a tie-breaker so as to ensure a strict ordering of route preferences.

The stable paths problem: A formal model for policy routing

This section presents a simple model for eBGP treating each AS as a single node developed by Griffin, Shepherd, and Wilfong.¹ Let \( G = (V, E) \) be an undirected graph representing the physical AS topology where \( V = \{0, 1, 2, \ldots , n\} \) is the the set of nodes and \( E \) is the set of edges. Since routes to different destinations are computed inde-
<table>
<thead>
<tr>
<th>Priority</th>
<th>Attribute</th>
<th>How used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Local_Pref</td>
<td>Highest Local_Pref, e.g., prefer customer routes over peer or provider routes.</td>
</tr>
<tr>
<td>2</td>
<td>AS_Path</td>
<td>Shortest AS path</td>
</tr>
<tr>
<td>3</td>
<td>MED</td>
<td>Lowest MED preferred. Typically ignored unless explicitly negotiated.</td>
</tr>
<tr>
<td>4</td>
<td>eBGP &gt; iBGP</td>
<td>Routes learned via eBGP are preferred to routes learned via iBGP.</td>
</tr>
<tr>
<td>5</td>
<td>IGP path</td>
<td>Pick route with lowest IGP cost to egress router.</td>
</tr>
<tr>
<td>6</td>
<td>Router ID</td>
<td>A tie-breaker based on the smallest router IP address.</td>
</tr>
</tbody>
</table>

Table 1.1: BGP route selection criteria.

Independently, we explain the protocol assuming a single origin node that serves as the destination prefix to which all other nodes attempt to establish routes. A path is said to be permitted at a node \( u \) if a sequence of peering transformations results in a route being received at \( u \).

Let \( \mathcal{P}(u) \) denote the set of permitted paths from a node \( v \) to the origin. Let \( \lambda^u(P) \) denote a ranking function that assigns a nonnegative integer rank to a permitted path \( P \). If \( P_1, P_2 \in \mathcal{P}(u) \) and \( \lambda^u(P_1) < \lambda^u(P_2) \), then \( P_2 \) is said to be preferred over \( P_1 \). We assume that for all nodes \( v \) other than the origin 1) the empty path \( e \) is permitted and is the lowest-ranked path, 2) the ranking among paths is strict, i.e., paths with different next-hop nodes have different ranks, 3) the paths do not have cycles or repeated nodes. Note that the tie breaking mechanism based on unique router identifiers and the path vector nature of the BGP protocol, i.e., including the entire AS_PATH in the advertised route, help ensure the strict ranking and loop-freeness properties.

A path assignment is a function \( \pi \) that maps each node \( u \in V \) to a path \( \pi(u) \in \mathcal{P}(u) \). Note that \( \pi(0) = \{0\} \).

The set of paths \( \text{choices}(\pi, u) \) for \( u \neq 0 \) is defined to be

\[
\text{choices}(\pi, u) = \{(u, v)\pi(v) | \{v \in \text{neighbors}(u)\} \cap \mathcal{P}(u) \}
\]

i.e., the set of permitted paths formed by extending the paths assigned to neighbors of \( u \). The path assignment \( \pi \) is stable at node \( u \) if

\[
\pi(u) = P \in \text{choices}(\pi, u, u) \text{ with the highest } \lambda^u(P)
\]

The path assignment \( \pi \) is stable if it is stable at all nodes. Note that the strictness of rankings implies that the above condition is well defined and that if \( \pi(u) = (u, w)P \), then \( \pi(w) = P \). The path \( \pi(u) \) is empty iff \( \text{choices}(\pi, u) \) is empty. Thus, any stable path assignment implicitly defines a tree rooted at the origin. However, this may not be a spanning tree as some nodes may not have a nonempty path.

**Examples of gadgets**

TBD.
Simple Path Vector Protocol: A model for BGP

The shortest path vector protocol (SPVP) is a simple model for how BGP addresses SPP. In SPVP, adjacent nodes exchange messages that are simply paths. When a node adopts a path \( P \), it sends the path \( P \) to each of its neighbors. Each node maintains a route information base (RIB) and a forwarding information base (FIB). The path \( \text{fib}(u) \) is \( u \)'s current path to the destination. The path \( \text{rib}(u \leftarrow w) \) stores the most recent path processed at \( u \) and received from \( w \). The set of path choices available at \( u \) is defined to be

\[
\text{choices}(u) = \{(u,w) \in P^{u} | P = \text{rib}(u \leftarrow w)\}
\]  

(1.3)

All messages are assumed to be processed using a reliable FIFO message queue between neighbors. A node updates its forwarding information base \( \text{fib}(u) \) upon receiving a path \( P \) from some neighbor \( w \) by executing the following two steps atomically:

1. Set \( \text{rib}(u \leftarrow w) = P \);
2. Let \( P' \) be the path with the highest rank in \( \text{choices}(u) \). If \( \text{fib}(u) \neq P' \), set \( \text{fib}(u) = P' \) and send \( P' \) to all neighbors.

The network state of the system is the collection of values \( \text{fib}(u) \) and \( \text{rib}(u \leftarrow w) \) across all nodes and all messages queued at nodes or in transit on communication links. The network state implicitly defines a path assignment \( \pi(u) = \text{fib}(u) \). The network state is stable if there are no queued or in-transit messages. An instance of the SPP problem is said to be solvable if a stable network state exists.

**Theorem 1.2.1** The problem of determining whether an instance of Stable Paths Problem is solvable is NP-complete.

**Theorem 1.2.2** The path assignment associated with any stable state in SPVP is a stable path assignment, and thus a solution to the Stable Paths Problem. Therefore, if the Stable Paths Problem is unsolvable, then SPVP never converges to a stable state.

Note that the converse of the above theorem is not true, i.e., solvability of the Stable Paths Problem does not imply that SPVP will converge to a stable state.

**Dispute Wheels**

Given the NP-completeness of the solvability problem, we look for a simple sufficient condition for SPVP to converge to a stable state (implying solvability of the Stable Paths Problem). This condition is the absence of a preference structure referred to as a dispute wheel.

Formally, a dispute wheel is a 3-tuple consisting of a sequence of nodes \( u_{0}, u_{1}, \ldots, u_{k-1} \), and sequences of nonempty paths \( Q_{0}, Q_{1}, \ldots, Q_{k-1} \) and \( R_{0}, R_{1}, \ldots, R_{k-1} \), such that for each \( 0 \leq i \leq k-1 \) we have 1) \( R_{i} \) is a path from \( u_{i} \) to \( u_{i+1} \), 2) \( Q_{i} \in P^{u_{i}} \), 3) \( R_{i}Q_{i+1} \in P^{u_{i}} \), and 4) \( \lambda^{u_{i}}(Q_{i}) \leq \lambda^{u_{i}}(R_{i}Q_{i+1}) \), where all subscripts are interpreted modulo \( k \). Figure ?? shows an illustration.

Both naughty_gadget and bad_gadget have a dispute wheel. Can you identify them?

**Theorem 1.2.3** If a Stable Paths Problem has no dispute wheel, then it has a unique solution and SPVP converges to the corresponding stable state.
Furthermore, any Stable Paths Problem obtained by deleting a subset of edges and permitted paths containing those edges in the original problem, also does not have a dispute wheel.

Note that examples such as bad_gadget and naughty_gadget and others where SPVP diverges involve some nodes preferring longer paths (in terms of hop count) over shorter ones. Indeed, such preferences are necessary for SPVP to diverge. Suppose edges were annotated with positive weights and the length of a path defined as the sum of the weights of its edges. Then,

**Theorem 1.2.4** If each node always prefers a shorter path over a longer path to a destination, then the Stable Paths Problem does not have a dispute wheel.

### 1.3 BGP in practice

**Typical routing policies**

Routing policies in the real-world may not always be consistent with some intuitive notion of shortest paths. However, persistent instability is rarely observed suggesting that dispute wheels are rare in practice. Why is this? Gao and Rexford investigated this question and formalized a set of policy guidelines to ensure that BGP converges to a stable state. Nicely, these guidelines and their implicit assumptions are also believed to be largely obeyed in practice because of the nature of business relationships between ISPs in practice.

AS relationships are of two types: customer-provider, or peer-to-peer. The customer-provider and peer-to-peer agreements translate into the following rules governing BGP export policies:

- **Exporting to a provider**: In exchanging routing information with a provider, an AS can export its routes and the routes of its customers, but cannot export routes learned from other providers or peers.
- **Exporting to a customer**: In exchanging routing information with a customer, an AS can export its routes as well as routes learned from its providers and peers.
- **Exporting to a peer**: In exchanging routing information with a peer, an AS can export its routes and the routes of its customers, but cannot export the routes learned from other providers or peers.

Customer-provider relationships between ASes form a hierarchy. The hierarchical structure arises in practice because an AS typically chooses a much larger AS as its provider. It is unlikely that a nationwide AS is a customer of a metropolitan AS. That is, if an AS $u$ is a customer of $v$ and $v$ is a customer of $w$, then $w$ can not be a customer of $u$. AS $w$ is said to be a direct provider of $u$ whereas ASes like $w$ that can be reached by following more than one provider links in sequence are called indirect providers. Any direct or indirect provider of $u$ can not be a customer of $u$. Formally, the subgraph consisting of only provider-to-customer links (or only customer-to-provider links) is a DAG.

**Theorem 1.3.1** For a BGP system that has only customer-provider and peer-to-peer relationships, if all ASes strictly prefer customer paths over peer or provider paths, then the BGP system converges to a stable state.

The above theorem provides a sufficient condition for BGP convergence. Note that if the AS topology and relationships were completely known, then the conditions in the theorems above can be checked efficiently. For example, cycles in a directed graph can be detected in time linear in the size of the graph. It can be verified that the assumptions above about — 1) export policies, 2) the hierarchical structure, and 3) ASes strictly preferring customer paths over non-customer paths — in conjunction imply the absence of a dispute
wheel that is a sufficient condition for SPVP convergence. Can you explain which of the above assumptions _naughty_gadget or _bad_gadget fail to satisfy?

The restriction that ASes strictly prefer customer paths over peer paths can be relaxed in the theorem above allowing a customer path to have the same preference as a peer path provided the AS hierarchy satisfies the following condition: an AS can not have a peer-to-peer relationship with a direct or indirect provider. Note that customer paths must still be strictly preferred over provider paths.

Customer-provider and peer-to-peer are the two most common relationships between neighboring ASes. However, an AS may also have a backup relationship with a neighboring AS. Having a backup relationship with a neighbor is important to maintain connectivity in face of link failures.

**Theorem 1.3.2** If all ASes 1) strictly prefer a route containing no backup links to a route containing a backup link, 2) strictly prefer a customer route over a provider route, and 3) prefer (possibly not strictly) a customer route over a peer route, then the BGP system converges to a stable state.

**A note on relationship Inference**

Today, we do not have a complete map of even the Internet’s physical AS topology, not to mention the exact nature of AS relationships that are not publicly disclosed. However, we have the ability to observe a sizable portion of the AS routing topology by conducting measurements from distributed vantage points.

These observed routes can be used to infer relationships between AS using a heuristic inference algorithm. If ASes obey the export rules listed above, then the resulting AS paths are _valley-free_, i.e., a route follows zero or more customer-to-provider links, zero or one peer-to-peer link, and zero or more provider-to-customer links. The heuristic inference algorithm attempts to discover inter-AS relationships such that the number of instances where valley-free routing is violated is minimized.

**Message complexity and timers: The devil in the details**

The discussion so far implicitly assumed that the message processing overhead in BGP is small, i.e., each node has sufficient resources to process all messages sent (reliably) by its neighbors. However, the number of messages generated by BGP during convergence can be prohibitively high. Labovitz et al. investigated the message complexity of BGP and showed it to be superexponential.

- spurious path exploration

**Minimum route advertisement interval**

BGP limits the exploration of spurious paths and the associated message complexity using a hold-down timer called the _MinRouteAdvertisementInterval_ (MRAI). The MRAI timer has a recommended default value of about 30 seconds. After a route to a prefix has been adver-
tised to a peer, subsequent updates are held back from that peer until the MRAI timer expires. The MRAI timer tries to ensure that a node processes an update from all of its peers before processing a second update from the same peer. This reduces the exploration of spurious paths. For example, in the complete graph example above, the MRAI timer ensures that there are no pending announcements for paths of length \( k \) when a \((k + 1)\)-length path has already been announced by some node. Thus, only paths increasing monotonically in length are explored by each node reducing the message complexity from \( O(n!) \) to \( O(n) \). 

**Route flap damping**

Although the MRAI timer reduces the number of messages during convergence, it can not suppress route instabilities caused by extraneous factors such as unstable links or policy decisions. Route flap damping (RFD) was designed to reduce route flux at coarse-grained time scales. A simple way to think about the two mechanisms is that MRAI reduces the number of messages for each routing event identified by some root cause (e.g., a link failure, recovery, or policy change) while RFD limits the rate at which these routing events are allowed to happen.

RFD works by suppressing routes to unstable prefixes. For each prefix \( D \) and each peer \( N \), a router maintains a penalty \( p[D,N] \) that is incremented every time the router receives an update from \( N \) for \( D \). When the penalty crosses a threshold called the suppression threshold, the route is suppressed. At all times, the penalty value exponentially decays with a configured half life (typically 15 minutes). After hitting the suppression threshold, when the penalty value decays to a lower threshold called the reuse threshold, the routes to \( D \) from \( N \) are allowed to be reused.

Mao et al. show that route flap damping can significantly delay Internet convergence times.

**Inconsistency: Why BGP suffers from unavailability**

Unavailability in BGP fundamentally stems from an inconsistent view between routers. Internet routing, especially interdomain routing, has traditionally favored responsiveness, i.e., how quickly the network reacts to changes, over consistency, i.e., ensuring that packets traverse adopted routes. A router applies a received update immediately to its forwarding table before propagating the update to other routers, including those that potentially depend upon the outcome of the update. Responsiveness comes at the cost of availability: a router A thinks its route to a destination is via B but B disagrees, either because 1) B's

---


old route to the destination is via A, causing loops, or 2) B does not have a current route to the destination, causing blackholes.

Below are several examples where inconsistent forwarding tables cause transient unavailability in interdomain routing today. In each case, the unavailability could last several tens of seconds (and sometimes minutes) due to BGP message processing and propagation delays.

**BGP link failures** Figure 1.1 shows how link failures cause transient loops in BGP. Bold lines show selected paths. If link 4-5 goes down, 4 would immediately send a withdrawal to 2 and 3. However, because both 2 and 3 know of alternate paths 3-4-5 and 2-4-5 respectively, they start to forward traffic to each other causing loops. The MRAI timer prevents 2 and 3 from advertising the new paths even though they have adopted them to forward traffic. The timer is believed necessary to prevent a super-exponential blowup in message overhead, and its recommended value is 30 seconds. Eventually, when the timer expires, both 2 and 3 discover the alternate path to 5 through 1 that existed all along.

**BGP policy changes** Figure 1.2 shows an example of how policy changes cause routing loops. AS4 may wish to engineer its traffic by withdrawing a prefix from 2 and 3 while continuing to advertise it to 6 for load balancing purposes. (For instance, by diverting traffic to arrive from 6 instead of 2, internal congestion within AS4 might be decreased.) If 2 and 3 each prefer the other over 6, routing loops would result like in (a). A similar situation also occurs if 5 wishes to switch its primary (backup) provider from 4 (1) to 1 (4); in this case, 5 is forced to either withdraw the route advertised (and potentially being used) to 4, or wait for a reliable indicator of when all traffic has completely moved over to the new primary provider. Other gadgets involving longer unavailability due to policy changes have been found.

**BGP policy cycles** Figure 1.3 is similar to the classic “bad gadget” involving cyclic preference dependencies. Each of ASes 1, 2, and 3 prefer to route via its clockwise neighbor over the direct path to AS 0, and

---


11 URL http://dl.acm.org/citation.cfm?id=1387589.1387614

---

does not prefer a path of length 3. The routes will never stabilize because there is no configuration where no AS wants to change its route to AS 0. Furthermore, the system goes through many repeated states involving routing loops causing chronic unavailability.

**iBGP link recovery** Figure 1.4 shows a transient blackhole caused by iBGP inconsistency. Routers A, B, and C belong to AS1 while D belongs to the adjacent AS2. iBGP is a BGP protocol that runs between routers inside an AS (in this case, A, B, and C). All routers route via D to the destination P in AS3. Suppose the previously failed link A-P recovers and is preferred by all AS1 routers over the route via AS2. If the AS1 routers all peer with each other, C will withdraw C-D-P from both A and B when it hears from A that A-P is available, but will leave it to A to announce AP to B directly because of the full-mesh design. If A is waiting upon its iBGP timer, B experiences a transient blackhole. The current BGP spec recommends an iBGP timer shorter than interdomain timers, and typical values range from 5-10 seconds.

Previous work showed that such blackholes can cause packet loss for tens of seconds. If AS1 routers use route reflection as opposed to a full-mesh configuration, similar consistency problems can cause unavailability.

---


Bibliography


RIPE. https://atlas.ripe.net/.