An Interior Path Vector Routing Protocol

Conor Creagh
Department of Computer Science,
University College Cork, Ireland.
conor@creagh.org

Cormac Sreenan
Department of Computer Science,
University College Cork, Ireland.
cjs@cs.ucc.ie

Abstract

Today's intra-domain protocols are limited in their scalability. We examine these limitations and propose an alternative in the form of an IGP based on path vectors. Taking advantage of the recent research interest in BGP's performance, we are able to develop a protocol that converges quickly, produces a relatively low level of control-plane traffic and promises to scale to very large networks, while still producing shortest path trees based on minimising latency or maximising bandwidth. We show that such a protocol converges and present results of its simulation.

1. Introduction

Interior gateway, or intra-domain, protocols have different requirements to exterior, or inter-domain, protocols. IGPs must converge quickly and minimise routing loops and control traffic. Traditionally, they have not included a means to apply routing policies; increasingly, they must be scalable, to deal with networks consisting of thousands of nodes.

In today's networks, two classes of protocols for routing IP traffic are in common use: distance vector and link state. Within autonomous systems (AS), the link state protocols Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS), as well as the distributed distance vector technique, Enhanced Interior Gateway Routing Protocol (EIGRP), dominate.

Between ASes, the Border Gateway Protocol (BGP-4) is used to exchange routes and apply policies that reflect each domain's business agreements with its customers, peers and providers. BGP-4 is the only IP routing protocol using path vectors, a variation on standard distance vectors where routing updates include the full path to a destination with the next hop information, rather than simply the distance and next hop. The aim of including the path to a destination with routing updates is to reduce the counting to infinity problem inherent to distance vector protocols. However, as we shall see, BGP can suffer from a *bouncing effect* in densely-meshed environments that drastically increases its time to convergence.

To the best of our knowledge, there has been no attempt to apply the results gained from the recent, intensive research on the workings and performance of BGP to the use of a path vector protocol as an interior gateway protocol. In the rest of this paper, we will explore this recent research (Section 2); in Section 3 we will examine the weaknesses of current IGPs that motivate the search for an alternative; Section 4 describes how we have designed such an interior path vector protocol (IPVP) and in Section 5 we present the results of a simulation and analysis of this protocol.

2. Related Work

In this section, we concentrate on the recent analyses and developments of path vector protocols, which have been stimulated in no small way by the importance of BGP's role in the stability and resiliency of the Internet.

Following the initial reports that inter-domain routing was not displaying the stability or speed of reconvergence under BGP that was expected, a number of analyses began into its behaviour, both empirical and theoretical. Varadhan, Govindan and Estrin looked at BGP from the point of view of the possibility that policies applied locally to a router might prevent network convergence [12]. Specifically, they found that these protocols could display persistent path oscillations given certain domain policies, noting that such problems had not yet been found in the field but predicting that they would begin to occur as the Internet grew.

At the same time, Labovitz, Malan and Jahanian collected data on BGP update traffic from several central Internet exchange points and found that the volume of this traffic was orders of magnitude greater than might have been expected [8]. Their results showed that the majority of control traffic was pathological, high-frequency routing updates and that there was a high degree of routing instability. (For example, there were, on average, 125 updates per network destination per day at the time of their study.) They

1-4244-0419-3/06/$20.00 ©2006 IEEE

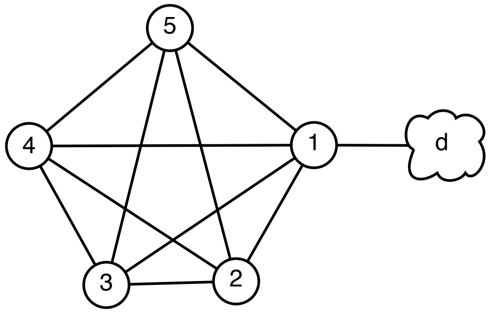

Figure 1. Bouncing Effect in Clique Topology.

apportioned blame for a small portion of this traffic to one router manufacturer's BGP implementation but at the time they were unable to explain the source of the majority of the pathological traffic.

Since then, research has continued along these two lines. Griffin, Wilfong, Shepherd and Premore have made a number of important contributions to the study of BGP not converging when the shortest path paradigm is not followed due to being overridden by routing policies [5, 6]. Labovitz *et al.* continued their measurements and analyses of routing traffic in the Internet: in [7] they showed that the theoretical upper bound for convergence of BGP is $O(n!)$, where n is the number of ASes, though that is unlikely to be reached in practice. They also showed that, for example, 20% of T_{long} and 40% of T_{down} events required more that three minutes to converge. (A T_{long} event is where a relatively short path to a destination is withdrawn but a longer path is already known; a T_{down} is where a destination is withdrawn with no replacement path available. Correspondingly, a T_{up} occurs when a network becomes initially reachable and a T_{short} is where a new best path to a destination appears.) They attributed the majority of the delayed convergence to the *bouncing effect* inherent to BGP, which can be visualised with the aid of Figure 1.

With all links up, the best path from router 4 to the destination network d is (4 1 d). In addition, many backup paths are available: (4 2 1 d), (4 3 1 d) and (4 5 1 d) traverse three links; (4 2 3 1 d), (4 2 5 1 d), (4 3 2 1 d), (4 3 5 1 d), (4 5 2 1 d) and (4 5 3 1 d) all traverse four links; and the paths (4 2 3 5 1 d), (4 2 5 3 1 d), (4 3 2 5 1 d), etc., all traverse five links to reach d. When the link between 1 and d fails, router 1 will withdraw its previous advertisement of d from its neighbours 2, 3, 4 and 5. Upon receipt of the withdrawal, 4 will look in its routing information bases (RIB) for its neighbours (Adj-RIB-in) for alternatives and will choose, say, (4 2 1 d), even though this path includes the broken link (1 d). Other routers will chose other alternative, and equally invalid, backup paths, and advertise them in turn to their neighbours. This continues as paths of increasing length (up to five hops in this example) are examined and dismissed as ineligible before the system finally converges.

A number of proposals have been made to limit this exploration of invalid, alternative paths. Bremler-Barr, Afek and Schwarz [2] recommend deleting (flushing) the advertisements to invalid (ghost) destinations that would otherwise be updated by routing announcements but are delayed due to the MRAI, thereby eliminating invalid paths.

Pei *et al.* proposed comparing path information from neighbours advertising the same destination network and checking them for consistency [10]. If two paths intersect at a node, then succeeding nodes should normally be the same in both paths. If not, then inconsistent information is being propagated by one or more nodes and all but one path is marked as infeasible.

BGP with Root Cause Notification (BGP-RCN) [9] is perhaps the most aggressive, and effective, way of tackling the bouncing effect of Path Vector Protocols. By including the reason and location of the causal node in withdrawal messages, every other node in the network is immediately able to avoid falling back on other paths that also include this node. The authors show that the upper bound on convergence is reduced to $O(d)$ where d is the diameter of the network. The use of a sequence number in BGP-RCN updates also contributes to the increased speed and reduced number of routing updates following a change in the network. Updates are only accepted as being valid by a node u if the update contains a sequence number for some node v in the ASPATH that is higher (sequentially) than the one already stored by u for v. We will examine RCN in greater detail in Section 4.2.

3. Interior Routing Today

Since 1992, Open Shortest Path First (OSPF) has been the routing protocol recommended by the Internet Architecture Board for use as an IGP. Intermediate System to Intermediate System (IS-IS) and Enhanced Interior Gateway Protocol (EIGRP) are also in common use, the former especially in the ISP community. In this section, we briefly examine these protocols, paying particular attention to their scalability. In addition, we look at how BGP is often used out of necessity as a glue to bind multiple, separate instances of IGPs together in a large AS.

3.1. OSPF

OSPF is the interior gateway protocol in most common use today. In medium-sized networks, it converges quickly and does not consume much bandwidth for flooding Link State Advertisements (LSA) to other routers. Unfortunately,

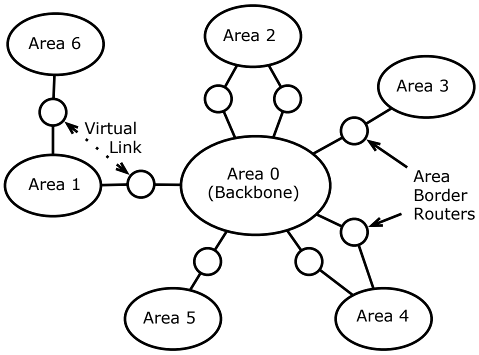

Figure 2. Areas in OSPF.

it does not scale linearly: the shortest path calculation (usually a variation on Dijkstra's algorithm) is O$(l * log(n))$, where l is the number of links in the network and n the number of nodes. In addition, the greater the number of routers in an OSPF network, the greater the level of flooding traffic, as LSAs must be reliably flooded to all other routers upon any change in the network and, in any case, at regular, thirty-minute intervals. Every time a router receives an LSA, it performs its shortest path calculation again: in relatively stable networks, this leads to a high proportion of shortest path calculations resulting in no change to the network's shortest path trees.

OSPF uses a two-level hierarchy to reduce the sizes of routers' link state databases, routing tables, shortest path calculations and routing control-plane traffic. The hierarchy uses a single backbone area (area 0) to interconnect all other areas. In each non-backbone area, OSPF runs in the normal, flat fashion, using network and link LSAs for descriptions of the network nodes and interconnecting links, respectively. Each area's link state database is maintained separately and each of these areas is connected to the backbone area by Area Border Routers (ABR), at which point address summarisation of the networks within areas can occur if addresses are appropriately distributed. Within the backbone area, a distance vector technique is used to disseminate information from non-backbone areas.

3.2. IS-IS

Deployment of IS-IS is generally limited to the backbone networks of larger Internet Service Providers (ISP), where it was often already running before OSPF's development was finalised. As with OSPF, IS-IS also enables hierarchical routing, though in a slightly different way. Routers are in exactly one area but can be of three types: Level 1, Level 2 or a combination of Level 1 and Level 2. Level 1 routers communicate with other Level 1 routers in the same area whereas Level 2 routers create an intradomain backbone between areas. (In contrast, ABRs in OSPF belong, by definition, to more than one area.) These roles are illustrated in Figure 3.

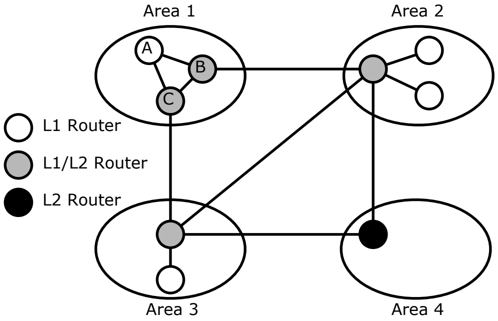

Figure 3. Areas and Levels in IS-IS Routing.

Routers that are both Levels 1 and 2 indicate this in their link state packets or PDUs (LSP) to other Level 1 routers who can, in turn, calculate their closest Level 2 neighbours. Level 1 routers then transmit packets destined for areas other than their own via their closest Level 2 neighbour, which can result in sub-optimal routing. For example, in Figure 3, if the IS-IS metric between routers A and C is less than that between routers A and B, then A will choose C as its nearest Level 2 router, even for traffic to Area 2. Level 2 routers usually have neighbours in more than one area and therefore must participate in the link state calculations of more than one area, with the increased overhead of LSP distribution and SPF calculation.

3.3. EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is the most recent DV protocol to be developed and widely deployed in IP networks. It was designed in 1994 and is based on the Diffusing Update Algorithm (DUAL), developed by Garcia-Lunes-Aceves in 1993 [4].

Unlike traditional distance vector protocols, EIGRP does not use periodic updates, relying instead on neighbouring relationships with other routers: once a router has sent a routing update for a remote destination to an adjacent neighbour, it will not send another update related to this destination while its path to the destination is unchanged. EIGRP uses a method of calculating costs of paths that depends on bandwidth, occupancy (load), delay and reliability. It scales relatively well and converges quickly but is a proprietary protocol, preventing it from being used in multi-vendor networks.

DUAL lies at the core of EIGRP and differentiates it from other DV protocols that use Bellman-Ford. The concept used is that when a router u chooses a next hop v for destination d, the path is guaranteed to be loop-free only when the cost from v to d is less than the least cost from u to d via any alternative next hop. EIGRP is unique in making use of this fact.

When a router does not have a backup path available (a feasible successor), it has to "go active" by specifically querying its neighbours. It can sometimes become stuck in this state, and after a time, it must reinitialise its peerings with all its neighbours. Steps to reduce the likelihood of this happening involve limiting the query scope by hiding networks through route filtering and summarisation, actions that are both manual and prone to error.

3.4. Interconnecting IGPs with BGP

Up to now, we have only looked at single instances of IGPs, such as IS-IS, OSPF and EIGRP, within an autonomous system. However, there are often good reasons to use BGP in the centre of large networks: most of these are related to the inability of current IGPs to scale well.

When OSPF areas or IS-IS levels reach a certain size it would be useful to create further levels of hierarchy: unfortunately, this is impossible. An alternative is to create a new backbone with BGP, allowing multiple instances of one or more types of IGP to coexist in one large, loosely-knit routing domain. If external BGP (eBGP) is used for this core, then the interactions between different IGPs, or different instances of IGPs, can be controlled by BGP policies, and each IGP "zone" is encapsulated with an AS, allowing further subdivision if required. If internal BGP (iBGP) is used, different possibilities ensue, such as load-sharing across the iBGP core. In either case, another instance of an IGP would usually be needed to provide the Layer 3 connectivity required for the establishment of BGP's TCP connexions.

4. Description of IPVP

As we have seen in Section 2, BGP is a path vector protocol for a network whose nodes are autonomous systems. It is slow to converge following T_{down} and T_{long} events and there is a large overhead of routing traffic for these events. Nonetheless, many of the ideas from BGP can be usefully reused in an interior path vector protocol (IPVP) and much of the research into path vector protocols inspired by BGP's importance is of use to us in designing an IPVP.

4.1. Neighbour Discovery, Adjacencies and Path Selection

Routers can discover their neighbours in a variety of ways: manual configuration, a "hello" protocol or other broadcast advertisement, or indeed they can simply transmit routing information towards potential neighbours without concerning themselves with its arrival at a destination. A hybrid approach is appropriate for IPVP: in broadcast networks, routers speaking IPVP advertise themselves by pinging a defined multicast address and waiting for their neighbours to initiate peering with them; in non-broadcast environments, it is reasonable to assume that the details of neighbours' IP addresses will be manually configured in a router's configuration.

Adjacencies can be permanent or more ephemeral. Maintaining permanent, reliable relationships with neighbours is an overhead that can be worthwhile if it reduces routing control traffic in the long run.

The approach chosen for IPVP is that neighbouring routers make TCP connexions with each other, negotiate IPVP peering details such as hold timers, and then exchange routing information. Apart from small, regular keep-alive messages between peers, no other communication is necessary as long as the network remains stable. In LAN environments, a *route reflector* can be used as a focal point for all peerings, receiving each router's advertisements and relaying them to the other routers in the broadcast domain, thereby reducing the number of peerings to n from $O(n^2)$, analogous to the notion of designated router in OSPF and IS-IS.

Traditional DV protocols use a simple hop count mechanism for choosing between alternative paths to a destination network. This is insufficient for a modern IGP, however, where longer paths of greater bandwidth can be preferable to paths of fewer hops but limited capacity. Choosing a cost based on a link's bandwidth and using the number of hops as a discriminator only in the event of equal-cost alternative paths, when used consistently throughout the network, leads to a provably-convergent shortest path network, as we shall show in Section 4.3.

4.2. Minimising Convergence Delay

In Section 2, we reviewed the problems inherent to a PVP. In this section, we show, in detail, an approach to speed convergence and minimise routing control traffic. The practice of marking each routing update with a tag representing the causal, or reporting, node of a failure was first suggested by Cheng *et al.* [3]. Pei *et al.* [9] propose combining the identity of the node adjacent to the failing link with the use of sequence numbers to address the issue of multiple failures overlapping in time, and it is this approach we follow.

A root cause node in IPVP is a router adjacent to a link whose change in status causes the paths of downstream routers to change. For each destination network, d, known to a router u, the router maintains a sequence number for each of the intermediate routers in the path to d. When u receives an update message from a neighbour v specifying a change in v's path to d, the update includes the IPVP identifier of

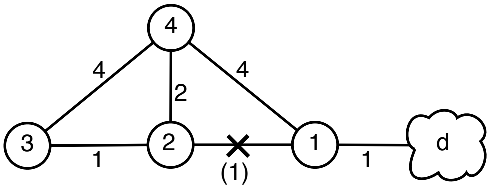

Figure 4. Root Cause Notification in action.

Round	Update	Sender's Seq. No.	Router Path
I	$2 \longrightarrow 3$	1	ϵ
	$2 \longrightarrow 4$	1	ϵ
II	$4 \longrightarrow 2$	1	4 1 d
	$4 \longrightarrow 3$	1	4 1 d
III	$2 \longrightarrow 3$	2	2 4 1 d
	$3 \longrightarrow 2$	1	3 4 1 d
IV	$3 \longrightarrow 2$	2	ϵ

Table 1. Sequence of updates resulting from state depicted in Fig. 4, a T_{long} event.

the root cause node c as well as c's new sequence number for d. Router u can examine its Adj-RIBs-in for entries related to d and, for each entry that includes c in the router path, delete entries where c's sequence number is older than that in the latest update message. If the sequence number of the root cause node in the received update message is not newer than those stored in u's Adj-RIBs-in, then the update message can be discarded. The sequence number at u for d increases when u's path to d changes.

Root cause notification (RCN) can be seen more clearly through studying the effects of the link between routers 1 and 2 failing in Figure 4. Before the break, router 3 will have known of two paths to d: (2 1 d) and (4 2 1 d) and will have chosen the path (2 1 d) because of its lower cost. When the link between 2 and 1 fails, router 2 will advertise its unreachability of d to routers 3 and 4 (it no longer has a route since 4's best path was also through 2), will set the RCN of the update to 2 and increase its sequence number for d. When 3 checks its Adj-RIBs-in for alternative paths to d, it will see that the other available path, (4 2 1 d), also includes router 2 and that the sequence number for 2 in that path is lower than the sequence number for 2 in the recently received update message for d. It can therefore ignore this alternative and wait for the update from router 4 that will contain an advertisement of the path (4 1 d). Clearly, the number of update messages when RCN is used is less than when the bouncing effect is allowed to proceed unhindered.

In Table 1, for simplicity, we make the assumptions that there is an equal propagation delay of update messages at each router, i.e. that updates flow in waves from the source through progressively more distant routers, that routers process received advertisements before sending any updates themselves, and that sequence numbers are all zero before the failure of the link between 1 and 2. We can see that each router's sequence number for d increases with its new choice of path. All updates in this sequence will show that router 2 is the root cause node.

4.3. Convergence of IPVP

João Luís Sobrinho provides an algebraic framework for the examination of path vector protocols [11]. In doing so, he allows us to more easily prove whether a given protocol will converge by examining the protocol's monotonic and isotonic properties, where monotonicity means that the weight, or cost, of a path does not decrease when the path is extended and isotonicity means that the relationship between the weights of two paths with the same origin is preserved when both paths are extended to the same node.

The algebra is a seven-tuple of $(W, \preceq, L, \Sigma, \phi, \oplus, f)$. W is a set of weights of paths which are ordered by the relation $\preceq$. L is a set of labels where each link in the network has a unique label, so that the label $l(u, v)$ is the label of the link between u and v. Σ is the set of path signatures, with the special signature ϕ representing an unusable path. Signatures are defined inductively, where $s \in \Sigma$ is defined as:

$$s(uv \circ Q) = l(u, v) \oplus s(Q),$$

for the non-trivial path $uv \circ Q$ and $l \in L$. The operation $\oplus$ has domain $L \times \Sigma$ and range Σ and allows for a single link to be added to a path. The function f relates signatures and weights, so that $f(s(P))$ is the weight of the path P.

Monotonicity can therefore be expressed as:

$$\forall_{l \in L}, \forall_{\alpha \in \Sigma}, f(\alpha) \preceq f(l \oplus \alpha),$$

and isotonicity as:

$$\begin{aligned} \forall_{l \in L}, \forall_{\alpha, \beta \in \Sigma}, f(\alpha) &\preceq f(\beta) \\ \Rightarrow f(l \oplus \alpha) &\preceq f(l \oplus \beta). \end{aligned}$$

In addition, strict monotonicity is defined as:

$$\forall_{l \in L}, \forall_{\alpha \in \Sigma - \{\phi\}}, f(\alpha) \prec f(l \oplus \alpha).$$

Sobrinho shows that a path vector protocol converges to local-optimal path in-trees if and only if the protocol's algebra is monotone and that it converges to optimal path in-trees if and only if the algebra is isotone as well. (An optimal path from a node u to a destination d is a usable path with weight less than or equal to any other path from u to d according to the operation $\preceq$. A local-optimal path is optimal with respect to the set of paths originating in out-neighbours of node u when those paths are extended to u. This corresponds to a temporally optimal path while a set of routers is in the process of learning the optimal path of the converged state.)

It suffices for us to show that the IPVP described in this section is both monotonic and isotonic to see that it will converge in finite time. A suitable weight (W), or cost, of a usable link in an IGP is a positive real number inversely proportional to the link's bandwidth. The unusable path, ϕ, is represented by a cost of ∞; the operator $\oplus$ is addition, $+$. The following values correspond to the conventional shortest path, shown by Sobrinho to converge:

$$\begin{aligned} W &\equiv R_0^+ \cup \infty, \\ \oplus &\equiv +, \\ \phi &\equiv \infty, \qquad \text{and} \\ \preceq &\equiv \leq . \end{aligned}$$

5. Simulation and Analysis

We simulated the protocol described above, with some minor simplifications, using the SSFnet simulator [1]. In all cases, we simulated IPVP in regular topologies of varying sizes: *rings*, where each router peers with two neighbours; *grids*, where routers peer with two, three or four others, depending on whether the router is at a corner, an edge or in the core of the grid, respectively; and *cliques*, fully-meshed environments where each router peers with every other router. We find that convergence occurs in all cases and, for each simulation, look at the number of update messages required for convergence to occur after each of the events T_{up}, T_{down}, T_{long} and T_{short}. (A direct comparison with simulations of OSPF was not practical as the focus of ours was on numbers of messages per topology, in order to examine the scalability of the protocol.)

Figure 5 shows the numbers of update messages for the T_{long} and T_{short} events in a ring topology. (T_{up} and T_{down} both require the same number as T_{short}.) The results are intuitive: updates emanate from the causal router and are forwarded in both direction around the ring; a T_{long} is effectively a T_{down} sourced from one router followed by a T_{up} from another router.

In Figure 6, the number of update messages and the number of updates processed per router for T_{long} and T_{down} events are shown for the fully-meshed clique topology. (T_{up} and T_{short} are similar to T_{down}.) The T_{down}, T_{up} and T_{short} events lead to a total of $(n-1)^2$ messages, or just two messages per peering session in the network. The T_{long} event leads to $O(2n^2)$ messages, less than double the number required for the simpler events. Again, this is consistent with T_{long} being a combination of T_{down} and T_{up}.

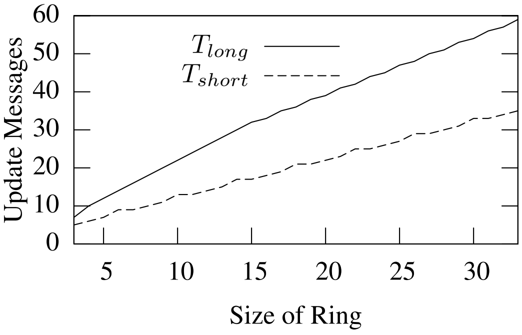

Figure 5. Numbers of T_{long} and T_{short} `update` messages in ring topologies.

Figure 7 shows the numbers of update messages required in grids. In this topology, T_{up} was similar to T_{short}. Grids provide a variety of equal-cost paths to a destination (when the costs of all hops are equal) which present a challenge to a DV protocol to minimise the number of messages that would be caused by the counting to infinity or bouncing effects. The number of messages exchanged for a grid to converge is dependent on the tie-breaking method employed where otherwise equally long paths are encountered. We chose using the lowest of the advertising neighbours' IPVP identifiers as the discriminator in such cases, which is deterministic and thereby prevents loops being formed.

Of the three topologies studied, grids are by far the most demanding for a path vector protocol, even when root cause notification is employed. This is due to the number of equal-cost paths availiable at any point in time during convergence following T_{up}, T_{short} or T_{long}. Grid-like networks, where multiple equal-cost paths exist at a variety of distances from a given destination, present a worst-case situation for path-vector protocols and are fortunately rare in the field. In other regular environments, we have seen that the number of update messages processed per peering varies between 2 and 3.5, with no exponential trends.

Analysis Two IPVP messages are of interest when analysing the protocol's run-time behaviour: updates and keepalives. Keepalive messages are transmitted from peer to peer at 10s intervals and form a 19 octet payload for a TCP packet, which equates to 34 bps when the TCP header is taken into consideration. In a steady state, i.e. while the network is converged, this is the only traffic resulting from IPVP and represents the base line for comparison with other protocols. In contrast, the link state protocols perform regular flooding of their databases throughout the network, although this does usually not have an undue impact on bandwidth availability for other applications.

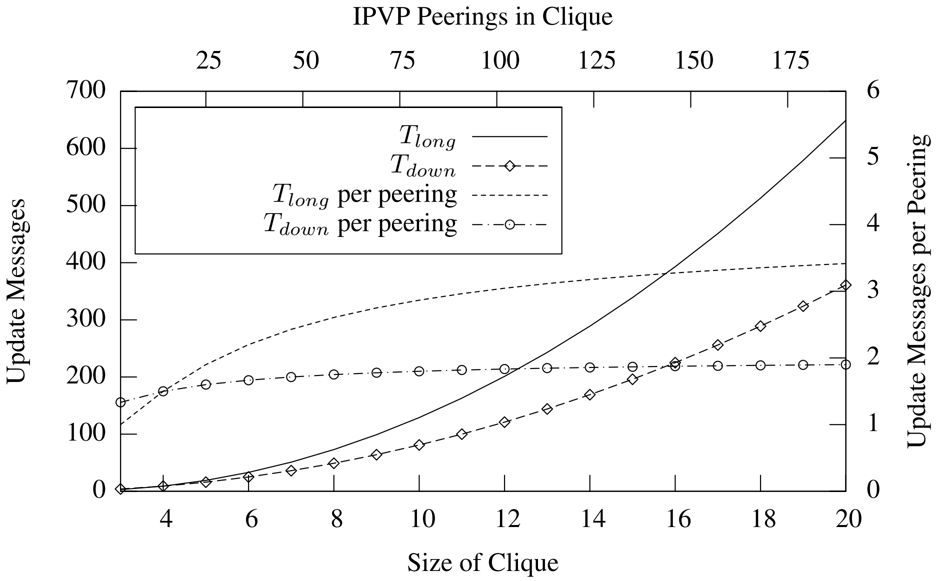

Figure 6. Numbers of T_{down} and T_{long} `update` messages in clique topologies.

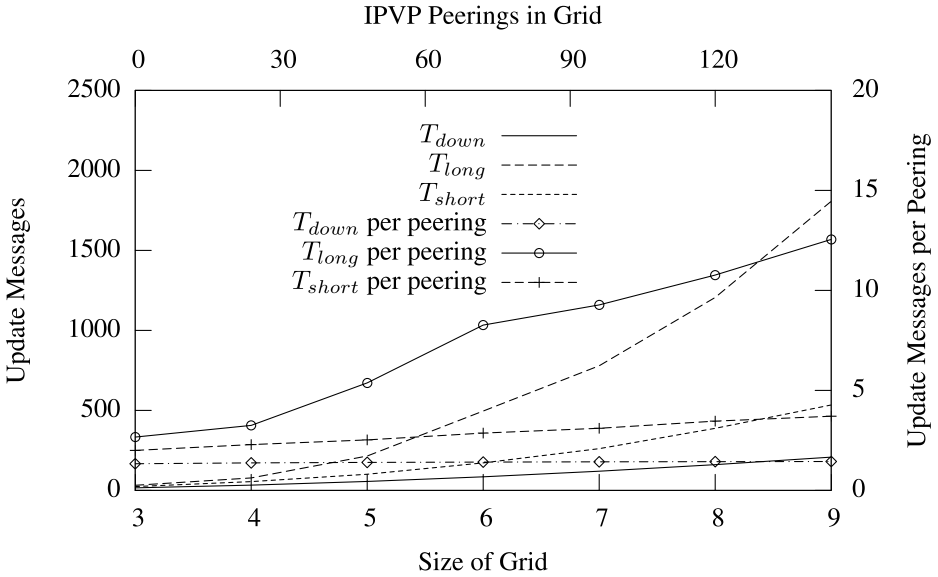

Figure 7. Numbers of T_{down}, T_{long} and T_{short} `update` messages in grid topologies.

Upon receipt of an update message, a router checks the the corresponding routing information base (RIB) to ascertain whether the event is a T_{up}, T_{down}, T_{short} or T_{long}, which is, at worst, an $O(n)$ search. A comparison is also made with the contents of the local RIB ($O(n)$) to ascertain if the event could lead to a change in the best path. In some cases, all RIBs will need to be examined to find the new best path to the destination, an operation related to the number of the router's peers and the number of networks. Overall, there is no task that requires processing that is not linearly directly proportional to the number of destinations in the network.

Each router has a RIB for each of its peers. There will be at least one entry in these RIBs for each destination in the network and, depending on a router's location in the network, a greater or lesser likelihood of a destination simultaneously existing in multiple RIBs, since multiple peers might advertise the same destination. An entry for a destination requires a similar amount of memory to the size of the `update` message that generates the entry: at a minimum, 36 octets per destination network plus 6 additional octets per intermediate router to the destination. The local RIB contains exactly one entry per destination in the network.

As an illustration, a router that has five peers and is in a network of 10,000 distinct destinations will require $10,000 * (30 + 10 * 6) * 5 = 4.5$MB of memory, in the worst case in a netowrk of diameter 10.

The processing required on a router for a single update has been shown to be a finite series of $O(n)$ tasks and we have seen that the memory footprint of IPVP is well within the capabilities of current hardware, even for networks consisting of thousands of destinations and routers. Furthermore, while networks are in a steady state, no processing other than managing keepalive messages with directly connected neighbours is required by a router running IPVP.

6. Conclusion

Current IGPs, such as IS-IS, OSPF and EIGRP, perform well in networks whose size would have been considered large when the protocols were being developed. Nowadays, however, it is not uncommon for large network service providers or global corporations to have networks that consist of tens of thousands of routers. We have seen that the link state protocols have built-in hierarchy that allows them to be scaled to a certain extent, but the most common resolution of the scalability problem today is to create a backbone of BGP within the enterprise that allows separate instances of IGPs to run within their measure.

In this paper, we have presented an interior gateway protocol that combines the path vector paradigm for route dissemination and loop avoidance with the ability to choose paths that are optimal according to the usual IGP standards of maximising bandwidth or minimising latency. We have seen that its footprint on a router is small in terms of memory requirements and that it presents no load that does not scale linearly with the size of the network in which it is running. Although IS-IS and OSPF will be faster to converge in smaller networks, it is expected that IPVP will be scalable to several thousand nodes without suffering from the computational overheads and limitations of link state protocols. Such scalability offers network operations the possibility of running a single protocol across large networks, thereby avoiding the problems associated with hierarchies, route redistribution and managing metrics from different protocols.

Indeed, the prospect exists of being able to include the full Internet routing table in the IGP for the first time since the mid-1990's. Combining EGP and IGP costs to optimise egress points from large networks to the Internet is an intriguing area of study that has not been included in the present work.

References

[1] *The Scalable Simulation Framework Network Project*, 2005. http://www.ssfnet.org/.

[2] A. Bremler-Barr, Y. Afek, and S. Schwarz. Improved BGP Convergence via Ghost Flushing. In *Proc. IEEE INFOCOM*, Apr. 2003.

[3] C. Cheng, R. Riley, S. Kumar, and J. Garcia-Luna-Aceves. EIGRP - A Fast Routing Protocol Based On Distance Vectors. In *Proc. ACM Sigcomm*, Aug. 1989.

[4] J. Garcia-Lunes-Aceves. Loop-free routing using diffusing computations. In *IEEE/ACM Transactions on Networking*, volume 1, Feb. 1993.

[5] T. Griffin and B. Premore. An Experimental Analysis of BGP Convergence Time. In *Proceedings of ICNP*, Nov. 2001.

[6] T. Griffin, F. Shepherd, and G. Wilfong. The Stable Paths Problem and Interdomain Routing. In *IEEE/ACM Transactions on Networks*, volume 10, pages 232–243, 2002.

[7] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proc. ACM SIGCOMM*, Aug. 2000.

[8] C. Labovitz, G. Malan, and F. Jahanian. Internet Routing Instability. In *Proc. ACM SIGCOMM*, Aug. 1997.

[9] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP-RCN: Improving BGP Convergence through Root Cause Notification. *Computer Networks*, 48:175–194, June 2005.

[10] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. Improving BGP Convergence Through Consistency Assertions. In *Proc. IEEE INFOCOM*, June 2002.

[11] J. Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proc. ACM SIGCOMM 2003*, pages 49–60, Aug. 2003.

[12] K. Varadhan, G. R., and D. Estrin. *Persistent Route Oscillations in Inter-Domain Routing*. University of Southern California, Mar. 1996.