Incentive Based Inter-domain Routeing

Richard Mortier[1] and Ian Pratt[2]

[1] Microsoft Research Ltd., 7, JJ Thomson Avenue, Cambridge CB3 0FB, UK*
mort@ieee.org, Tel.: +44 1223 479830, Fax.: +44 1223 479999
[2] University of Cambridge Computer Lab, 15, JJ Thomson Avenue, Cambridge CB3 0FD, UK
ian.pratt@cl.cam.ac.uk, Tel.: +44 1223 334639

Abstract. The Internet's inter-domain routeing system has evolved to keep pace with the Internet's rapid growth, from a few co-operatively managed administrative domains to a large number of competetive domains. This growth has brought to light one of the Internet's shortcomings: lack of support for efficient control and management of traffic, particularly between domains. This paper presents an extension to BGP, the inter-domain routeing protocol, that enables congestion to drive route selection and thus allows economic incentives to play their part in traffic distribution. Implementation in a deployed BGP stack is discussed and a simple simulation presented, showing better traffic distribution.

1 Introduction

The Internet has evolved from a small number of co-operatively managed interconnected networks. It now consists of approximately 16 000 networks, or Autonomous Systems (ASs), with a similar number of competitive administrative domains. Rather than rely on co-operation, interconnecting networks enter into Service Level Agreements (SLAs) that contractually specify the parameters of the services to be provided. The process of managing the allocation of network resources to meet such obligations is known as *traffic engineering* [1].

Unfortunately, control and management protocols in the Internet have struggled to evolve to manage this increased complexity. The only mechanism available to operators to manage the distribution of traffic throughout the Internet is the inter-domain routeing protocol, Border Gateway Protocol v4 (BGP) [2,3]. This is used to advertise connectivity to particular Internet Protocol (IP) prefixes, since either the network advertising the prefix owns that prefix, or the network is providing transit for traffic belonging to a prefix owned by another network.

By controlling the distribution of adverts for these prefixes, based on the adverts' properties such as the originating AS, BGP provides for separation between topology and policy. However, the mechanisms it currently provides are generally not easily automated, requiring substantial manual intervention on the part of operators. Received wisdom has long held that BGP configuration is highly prone to errors, and there is increasing evidence that such errors both occur *and* significantly impact the operation

* The work presented in this paper was carried out while the author was a Ph.D. student at the University of Cambridge Computer Lab, UK.

B. Stiller et al. (Eds.): NGC/ICQT 2003, LNCS 2816, pp. 308–317, 2003.

of the Internet [4]. Furthermore BGP does not provide any facility for changing routes based on more dynamic metrics such as the current performance of different routes. This makes it difficult to correctly implement policies such as multi-homing, where a customer connects via multiple providers for performance and reliability reasons.

This paper presents an incremental modification to BGP allowing ASs to advertise a load-based price for carrying traffic from their peer ASs. This provides two significant benefits: it provides a globally valid metric that should improve the stability properties of BGP; and it allows route selection to be made incentive-compatible and more dynamic. This has a number of benefits: it allows more flexible traffic engineering policies to be implemented; it increases the potential for automation of Internet network management; and it increases the likely performance of the network.

Section 2 describes Internet routeing and related work using BGP and its mechanisms for Internet traffic engineering. Section 3 discusses issues surrounding the design of the pricing extension to BGP and its implementation. Sections 4 and 5 present a simulator built around a deployed BGP implementation, and a simple simulation demonstrating that stability of routes and a more even distribution of load can be achieved. Finally, Section 6 concludes with a pointer to some discussion of issues surrounding deployment.

2 Internet Routeing and Related Work

Traffic in the Internet is routed by each router matching the destination address of a packet against the longest (and so most specific) address prefix known to that router. The corresponding routeing table entry tells the router on which interface the packet should be transmitted in the hope that it will progress towards its destination. Information about which networks and routers 'own' prefixes is disseminated via routeing protocols.

In a given network there are effectively two classes of prefix, and so two classes of routeing protocol. The Interior Gateway Protocol (IGP) disseminates information about those prefixes that belong to the network in question. Common examples of such protocols include the Intermediate System-Intermediate System protocol (IS-IS) [5,6] and the Open Shortest Path First protocol (OSPF) [7]. The Exterior Gateway Protocol (EGP) disseminates information about those prefixes that are external to the AS in question; the only currently deployed example is the Border Gateway Protocol v4 (BGP) [2,3].

BGP is a *path-vector* routeing protocol, an extension of a distance-vector routeing protocol. Each node distributes to its neighbours, or peers, its current preferred routes to the destination prefixes of which it is aware along with *path attributes* that apply to those prefixes. BGP is used in two ways: as internal-BGP (iBGP), to readvertise externally learnt prefixes within the enclosing AS; and as external-BGP (eBGP), to advertise prefixes (both internal and external) to neighbouring ASs. A session is iBGP if both ends use the same AS number, and the effect is that all the path attributes are trusted.

When a node receives adverts for the same destination prefix from multiple neighbours, it executes some preference function to decide which it will use and continue to advertise. In this way nodes build up information about their current best *next hop* choices for the destination prefixes available in the network. Key path attributes include:

the mandatory AS-PATH attribute which lists the ASs on the path that this advert has taken to reach the current router, enabling loop detection and also application of policy; the first applied path attribute, the LOCAL-PREF, a simple locally valid preference; and the ultimate tie-breaker, selecting the peering router with numerically lowest IP address.

Existing work has addressed the problems of resource allocation and effective route choice in IGPs [8,9]. The key point about such work is that the problems are all contained within a single network and thus a single administrative domain. Consequently, more co-operation can be expected from network elements, and it is more reasonable to assume that implementation of a single consistent policy is desired. However, as BGP is used to distribute prefixes between ASs and thus administrative domains, there will be multiple policies to be implemented in a given AS based on those to which it connects, and little co-operation between competitors may be assumed.

To deal with these problems, BGP provides two basic mechanisms to enable operators to express policy. The first is *filtering*, controlling the routes that individual routers will accept and advertise based on properties of the peer or advert in question. The second is based on use of *path attributes*, associated by routers with each set of advertised prefixes, and used to influence the preference function at the receiving router. There are approximately fifteen different types of path attribute that may be used, and their treatment is standardised by the IETF [2].

The sheer number of available path attributes itself causes problems. For example, it makes understanding their interaction difficult, increasing the chances of misconfiguration, and can lead to problems with persistent oscillation [10,11]. Since correctly managing so many attributes is difficult, operators tend to use just a small well-understood subset. Two of the most commonly used are the AS-PATH and COMMUNITY attributes.

COMMUNITY attributes are opaque 32-bit tags associated with adverts that have semantics defined on a pairwise ad hoc basis between operators. They are commonly used to control the application of filters to adverts, or the modification or assignment of other path attributes to the advert. For example, the number of terms in the AS-PATH is used as part of the route preference function (shorter AS-PATHs being preferred), so by inserting multiple copies of their own AS number, operators gain some control over the routes their peers will prefer. The number of copies to insert is commonly controlled using different COMMUNITY attribute values as agreed between the operators.

In conclusion, BGP currently provides no mechanism allowing simple and consistent inter-domain traffic management. The mechanisms it does provide are ad hoc and difficult to make consistent between operators. Although extensions have been proposed to make particular functions consistent [12], they are limited in scope, and do not provide a generic mechanism for consistent inter-domain traffic engineering. The remainder of this paper presents a new path attribute that should enable consistent network-wide traffic engineering policies to be implemented, and also simplify the management process for the operator. Furthermore, it can be used to increase the control the operator has over traffic entering and leaving their network, allowing more flexible service differentiation.

3 Inter-AS Pricing

The principal aim of inter-AS pricing is to give greater control over traffic distribution to operators offering and receiving transit services. The current measured network load is transformed first into a per-AS *price*, and this is then transformed into a per-customer *charge* and advertised to customers. Note that in this context, an operator's *customers* refers to other network operators, both its clients and peers.

In fact, current peering policies are already influenced by network load: before operators decide to peer they attempt to calculate their traffic matrices and decide appropriate ratios for ingress to egress traffic on their respective networks. However, such decisions and measurements are taken over very long timescales, typically on the order of months. The mechanism presented in this paper should allow these timescales to be shortened to hours or perhaps even minutes.

Furthermore, those who desire higher quality service should have some mechanism to express this, for both transmitted *and* received traffic. Conversely, those operators providing transit services should have some mechanism allowing them to encourage or discourage customers (i.e. other operators) from routeing traffic toward them. Prices should be based on network load since that is the major cause of service variation in an operational network. Retaining the separation between price and charge allows arbitrary policy to be imposed by operators, while basing the network's route selection process on its load.

The natural mechanism to implement inter-AS pricing is as a new path attribute for BGP. Such a path attribute should be *optional* and *non-transitive* (i.e. not all BGP implementations need support it, and it need not always be communicated to peers). This preserves compatibility with prior versions of BGP whilst enabling incremental deployment. Inter-AS pricing can be split into three parts: *measuring congestion*, performed by the routers in the AS, based on metrics such as round-trip time estimates, packet drop rates, or packet mark rates; *calculating prices*, based on aggregated congestion measurements from the network; and *charging customers*, allowing policies to be applied by the provider based on customer, time-of-day, etc.

Dividing inter-AS pricing in this way achieves a number of goals. The principal from a technical point of view is that operators are given a rational mechanism to select between available routes. More nebulous effects include the easing of network management since usage based charging in this way gives a basis for automated settlement of bilateral peering arrangements; potential structural changes to the network, both technical and economic; and the possibility for operators to influence the route taken by traffic destined for them. This allows operators to begin to offer differentiated service in the Internet which can include some form of statement about the treatment of traffic in networks other than their own: a movement toward true end-to-end differentiated services.

3.1 Calculation of Prices

There are a number of design decisions implicit in the above description which will now be discussed in turn: the measure of congestion on which the price is based; the

association of the price with a node and not a link; and the constraints on the calculated prices.

Using marking information provided by routers seems a good basis for calculating a useful AS-wide measure of congestion as it takes into account both the remaining capacity and queueing delay on links. The price will thus be based on, and usually related to, the congestion that the router is experiencing. Effective schemes for smoothing and utilising the information available from packet marking are not discussed in this paper but are the subject of ongoing work.

Three constraints on the calculated price are identified here. The first two are fairly straightforward: the price should be positive[1] and should increase as the measure of congestion increases. The third is that the price should become less sensitive to changes in the load as the load increases.

As the network becomes excessively congested, route stability becomes more important as changing routes has a progressively more disruptive effect. In particular, route stability is likely to be a more important constraint than maximising the revenue generated by these price-based mechanisms: operators have other means to generate revenue, and there is not much that the routeing protocol can do to deal with a network which is simply overloaded. When the network does become overloaded, measures such as admission control and the end-to-end congestion control mechanisms of the transport protocols must play their part to reduce congestion. Such measures might be implemented through pricing visible to end users, but it is not within the remit of the routeing protocol to calculate or advertise these prices.

3.2 Expression of Policies

Before advertising the calculated price it is transformed according to local policy into a *charge*, then used by peers to calculate a `LOCAL-PREF` value as input to their own route selection process. This serves two purposes: (*i*) allowing the AS advertising the route to influence the route selection process of those receiving the advert; and (*ii*) allowing the AS receiving the advert to express more complex route selection policies. Such policies may be separated into two categories: static and dynamic.

Static policies include current BGP configurations such as 'always choose AS_i over AS_j' for a given prefix. For such policies the price path attribute acts purely as an accounting mechanism, simplifying the construction, parameterization and settlement of SLAs; it plays no direct part in the distribution of traffic through the network, and hence should not affect routeing stability.

Dynamic policies are more interesting and enable more expressive semantics but are harder to understand and predict in detail. Perhaps the most obvious such policy would be 'pick the cheapest route.' More complex policies could be implemented if the operator could measure the load neighbouring ASs were experiencing.

When calculating the price, a node has knowledge of the load its links are experiencing and the charges advertised to it from its peers. The price, p_i, at a node, N_i, may

[1] This is perhaps debatable if this mechanism were to be used to give discounts to peers, but such use is not considered further here.

thus be viewed as a function, $p_i = p(l_i^j, c_j^i) \quad \forall j \neq i$ where l_i^j is the load between nodes N_i and N_j, and c_j^i is the charge node N_j advertises to node N_i.

Using such load measurements, the operator could implement policies such as 'pick the highest quality route,' 'pick the cheapest route j such that $c_j^i l_i^j < C$' or 'pick the highest quality route j such that $c_j^i l_i^j < C$,' for some total cost C, perhaps itself time varying. The implementation of such remote monitoring facilities is not covered here; route server *looking glasses* already allow queries of the routes available to particular destinations, and these might be extended if the facility was considered desirable. In fact, companies offering such Internet performance measurement services now exist [13,14].

3.3 Settlement of Bills

Schemes for *settlement*, the process of converting into bills the charges associated with traffic, may be arbitrarily complex. The simplest and most obvious is to use traffic volume. This has a number of advantages: it is straightforward to understand and to measure; it is generally slowly varying between ASs, allowing operators to make relatively accurate predictions about future bills; and many operators already have to collect such information in order to police the SLAs into which they have entered.

Of course, scope exists for more complex settlement schemes. For example, if suitable feedback could be arranged, settlement might be performed based on the number of packets marked. Although this links the final bill more closely to congestion (since charges will not be levied unless congestion is occurring and hence packets being marked), such a scheme is more complex to understand and predict, and requires more infrastructure to support.

4 Simulation

Existing network simulators typically simulate at a per-packet level, making them inappropriate for BGP (and routeing in general) simulation: in such cases it is the macroscopic properties of the protocol that are of interest. Furthermore, they tend to use models of protocol behaviour rather than actual protocol implementations; since routeing protocols are notoriously difficult to implement correctly, such modelled versions must be viewed with some scepticism.

To avoid these problems, a simulator was developed based on a deployed implementation of routeing code: the GNU Zebra protocol suite [15]. This provides a number of dæmons which operate individual routeing protocols, and multiplex forwarding table updates to the kernel via a further *Zebra* dæmon. This suite was modified in three ways: (*i*) the Zebra dæmon was restricted to provide only logging and not to modify kernel forwarding tables; (*ii*) the BGP dæmon was modified to enable multiple copies to run on the local machine[2]; (*iii*) a simple harness was written that scheduled among the many BGP instances and provided simulation of load.

[2] Essentially by binding the BGP connection's local socket to a virtual IP interface on the local machine.

Realistic load simulation is complex and so for the initial simulator it was assumed that (*i*) the network is homogeneous in capacity, and (*ii*) each node sources equal amounts of traffic to all known prefixes. Although both are untrue in practice, they allowed some basic protocol properties to be investigated; furthermore, the Internet core currently uses technologies with commensurate, if not identical performance characteristics.

Under these two assumptions, load at a node is then calculated as the sum of two terms: (*i*) one unit per prefix advertised in the network, representing the load sourced by the node itself; and (*ii*) one unit per prefix per source for which the node's neighbours are using the node as next hop. For example, in a network of n nodes each advertising a single prefix, stub nodes should carry $2n - 1$ units: n units representing the load that the stub node is itself generating, and $n - 1$ units representing the load generated for it by every other node in the network.

Finally, code was added to implement the price path attribute, mimicking the usual way that path attributes are implemented. The result is a BGP simulator where the basic BGP code is essentially unmodified from a genuine deployed BGP implementation, and the extension under test is implemented in the same way that it would be implemented in reality. Although the simulation of load makes simplifying assumptions about the behaviour of traffic in the network, the state machine and routeing protocol behaviour are not simplified in any way.

5 Results and Discussion

The results in Figure 1 are intended as a simple illustration of this approach; more complete evaluation has yet to be carried out. Each AS contains a single router which introduces a single prefix to the system; the price is equal to the load incident to an AS; and the simplest non-trivial charging policy is uniformly applied: prefer the cheapest route. The nodes are numbered in the numeric order of their IP addresses with T_x nodes being transit nodes and S_x nodes being stub nodes.

In all simulations the stub nodes carry 11 units of load as expected. With unmodified BGP, the deterministic default tie-breaker[3] makes node T_1 preferred for transit to node T_2, which is then preferred to node T_3. The result is that nodes (T_1, T_2, T_3) carry (17, 15, 13) units of load respectively. With the BGP price attribute in place, a much more even distribution of load is achieved: all transit nodes carry 15 units of load each. The cost of this better balancing of load is an approximate doubling in the convergence time.

It should be noted that Zebra version 0.91a on which the simulator was based separately modifies the standard BGP route selection process. This version of Zebra changes the default tie-breaker to prefer the first-received route, in an attempt to reduce route-flap. The result using otherwise unmodified BGP is that the load distribution is then based on the precise ordering of message arrivals at nodes. This allows situations with evenly balanced load to arise at the price of the system becoming non-deterministic. Such a modification is actually required when using the price path attribute to prevent permanent oscillation: after the system equalizes load through the transit ASs, stub ASs

[3] All other things being equal, select the next hop with the numerically lowest IP address.

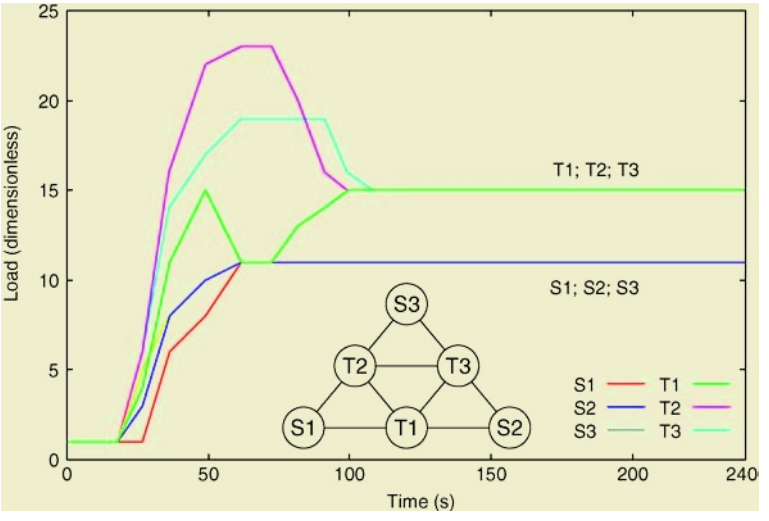

Fig. 1. Multi-homing with a small core. The embedded figure displays the simulated topology. The graph of results with unmodified BGP is omitted for space.

see equal prices; however, by default they would tie-break on lowest IP address, causing one transit AS to become overloaded and the pricing mechanism once again to take effect.

Two other mechanisms were also introduced to avoid route change synchronization leading to permanently oscillating routes. First, the number of routes that may have their `LOCAL-PREF` altered on the basis of a change in price is limited to one. This implements the most conservative load shedding policy; changes in price are still re-advertised as soon as they are processed. Second, the `LOCAL-PREF` may only be modified after a delay proportional to the maximum `AS-PATH` length in the network.

These changes attempt to ensure that changes in price have a chance to propagate throughout the network so that routes are not changed on the basis of out-of-date prices. An alternative, less pessimistic, scheme would be to choose the delay randomly from $[0, n]$ where n is proportional to the diameter of the network; this should decrease convergence times while still preventing synchronisation.

We remark here that several issues did become clear even with such a simple simulation. First, the number of BGP messages appears to increase when pricing is applied, as changes in load cause changes in prices which must be readvertised, potentially causing further changes in load. The magnitude of the increase in number of messages seems dependent on the topology and initial condition. With more realistic sizes of network and routeing tables this might become a problem and so deserves further investigation.

Second, correct choice of which routes to move to the cheaper AS can be difficult. If an AS advertises a reduction in its charge, the natural reaction is to cause as many routes as possible to use that AS as transit. However, doing so can increase the load on that AS to the extent that the price reduction is destroyed, and replaced by a price

increase. This can cause the AS receiving the advert to now choose to move its routes back, resulting in needless route flap.

There are two straightforward responses to this problem of increased route flap. First, the assumption that each AS sources traffic from only one prefix means that BGP has no flexibility over how much traffic to shift: it must move all or nothing. In a real deployment, a single AS is unlikely to both source sufficient traffic and do so toward a single prefix to cause this effect – where such a situation occurs, dynamic SLAs can be considered inappropriate without application of other techniques such as prefix disaggregation. Second, route flap damping [16] can be used to rate limit adverts in such situations.

However, it is clear that further work must include more detailed investigation of stability properties, particularly using more complex topologies and realistic load distributions coupled with different load redistribution policies. Mechanisms for providing useful measures of network load that can deal with Internet phenomena such as flash-crowds also need investigation, as do ways to turn such measures into prices. In particular, appropriate ways to aggregate mark information from routers to calculate AS-wide prices has not been addressed. Finally, given the freedom such a system potentially gives to operators to measure load and calculate prices, the interactions between different such mechanisms is yet another area requiring investigation.

6 Conclusion

This paper presented an incentive-based approach to routeing using the Internet's inter-domain routeing protocol, BGP. It presented the design of a new path attribute for BGP that enables choice between routes to be made on a more rational and incentive-compatible basis than currently possible. It continued with discussion of a BGP simulator written to test the behaviour of the new path attribute and presented a result from this implementation. Although detailed evaluation is beyond the scope of this paper, the simulation served to illustrate the ideas outlined here, and the simulator may prove useful in the future. Even such a simple simulation provided some directions where further work is needed to make pricing for BGP in this way viable.

The other area for further investigation not touched upon in this paper concerns more operational details of BGP: the behaviour of iBGP with pricing, and algorithms for combining iBGP advertised prices to achieve a price for the AS should be studied. Implementing more complex dynamic policies involving 'quality' estimates of neighbouring ASs, interaction between ASs applying different policies, and interactions between static and dynamic policies should also all be studied further. More details may be found in [17].

References

1. Xipeng Xiao, A. Hannan, B. Bailey, and L.M. Ni, "Traffic engineering with MPLS in the Internet," *IEEE Network Magazine*, vol. 14, no. 2, pp. 28–33, March/April 2000.
2. Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, IETF, Mar. 1995.

3. J.W. Stewart III, *BGP4 Inter-Domain Routing in the Internet*, Addison Wesley Longman, 1999.
4. R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proceedings of ACM SIGCOMM 2002*, Aug. 2002.
5. "OSI IS-IS Intra-domain Routing Protocol," RFC 1142, IETF, Feb. 1990.
6. R.W. Callon, "Use of OSI IS-IS for routing in TCP/IP and dual environments," RFC 1195, IETF, Dec. 1990.
7. J. Moy, "OSPF Version 2," RFC 2328, IETF, Apr. 1998.
8. B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, Mar. 2000.
9. C. Villamizar, "OSPF optimized multipath (OSPF-OMP)," in *Proceedings of the Forty-Fourth Internet Engineering Task Force*. IETF, Mar. 1999, available as Internet Draft `draft-ietf-ospf-omp-02`.
10. T. Griffin and G.T. Wilfong, "An analysis of BGP convergence properties," *Computer Communication Review (CCR)*, vol. 29, no. 4, pp. 277–288, Oct. 1999, Proceedings of ACM SIGCOMM 1999.
11. C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *Computer Communication Review (CCR)*, vol. 30, no. 4, pp. 175–187, Oct. 2000, Proceedings of ACM SIGCOMM 2000.
12. L. Gao, T. Griffin, and J. Rexford, "Inherently safe backup routing with BGP," in *Proceedings of IEEE Infocom 2001*, Anchorage, Alaska, Apr. 2001, pp. 547–556.
13. Keynote.com, "Keynote.com," `http://www.keynote.com/`, 2001.
14. Matrix.net, "Matrix.net," `http://www.matrix.net/`, 2001.
15. DML Networks, Inc., "The GNU Zebra Routeing Protocol Suite," `http://www.zebra.org/`, 2002.
16. C. Villamizar, R. Chandra, and R. Govindan, "BGP Route Flap Damping," RFC 2439, IETF, Nov. 1998.
17. Richard Mortier, "Internet traffic engineering," Tech. Rep. UCAM-CL-TR-532, University of Cambridge, Computer Laboratory, JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom, phone +44 1223 763500, Apr. 2002.