Title
Modifying Shortest Path Routing Protocols to Create Symmetrical Routes

Authors
Ghosh, Rajib
Varghese, George

Publication Date
2001-09-11

Peer reviewed

Modifying Shortest Path Routing Protocols to Create Symmetrical Routes

Rajib Ghosh, Citibank
George Varghese, UCSD

Abstract—**We describe a new mechanism to deal with asymmetries that arise in routing protocols. We show how to avoid route asymmetries (due to non-unique shortest paths) for any shortest path protocol by adding random integer link costs. We show in detail how RIP can be modified to avoid route asymmetry with high probability, without affecting either its efficiency or performance metrics such as convergence time.**

I. Introduction

Computing routes between source and destination end-nodes is a fundamental task in any large computer network. This paper deals with the issue of *route asymmetries*, and a method to avoid such asymmetries with high probability. Our ideas apply to other datagram routing protocols and even to route calculation in virtual circuit networks.

Symmetry in routes and link costs is generally assumed to hold. However, Paxson's classic study of Internet routing [Pax96a], [Pax96b] reveals that approximately half of the measured Internet routes include asymmetric paths that visit at least one different city. The two most natural symmetry assumptions in routing are *symmetrical link costs* and *symmetrical routes.* It is common to assume that the cost of a link from node A to neighboring node B is the same as the cost of the link from node B to node A. Similarly, if A and B are not neighbors, and the route from A to B is $A, R_1, \ldots, R_n, B$, it is natural to assume that the route from B to A is the reverse route $B, R_n, \ldots, R_1, A$.

Neither of these assumptions is true. Many links consist of two simplex links in each direction, with each simplex link having potentially different characteristics. Examples include cable and satellite links. Even if link costs were symmetrical, most routing protocols make no effort to compute symmetrical routes: if there are multiple shortest path routes, arbitrary tiebreakers can lead to asymmetrical routes. In some cases, routing protocols deliberately compute asymmetrical routes because of policy constraints [Pax96a].

Most standard routing protocols (i.e., Distance-Vector, Link-State) deal well with asymmetrical link costs for computing routes between a single source and a single destination. However many *Multicast* routing algorithms tacitly assume link asymmetry. A lack of link symmetry can lead to the calculation of suboptimal multicast trees. This can be avoided by modifying existing routing algorithms to calculate multicast "From Trees"[1] in addition to "To Trees".

Clearly link asymmetry leads to route asymmetry. In Section II we assume link symmetry, and isolate the impact of routing protocols on route asymmetry. We start by describing why symmetrical routes are desirable, and what are the major causes of route asymmetry. We then describe a new technique for avoiding asymmetries due to non-unique shortest paths. The main idea is to add an additional random cost component to each link so that, with high probability, there is only one shortest path between every source and destination. We validate our scheme by a theoretical analysis (Section II-B) and by simulations on real Internet domains that have asymmetrical routes (Section II-C). In Section III, we discuss possible modifications to a common Interior Gateway Protocol, RIP, to provide symmetrical routes within a domain.We conclude in Section IV by discussing the application of our ideas to other routing protocols.

II. Route Asymmetry

We begin this section by discussing why route symmetry is desirable, and the impact of asymmetry on routing protocols and measurements. We then analyze the two main causes for route asymmetry: non-unique shortest paths and policy routing.

Importance of Routing Symmetry: While route asymmetry is not as pernicious as some of the other pathologies described in [Pax96a] (e.g., routing loops),

[1] This idea was invented by Steve Deering and Christian Huitema

it does affect several protocols. For example, the Network Time Protocol, NTP [Mil85], approximates one-way propagation time as half of the round-trip time between two hosts when synchronizing clocks between widely separated hosts. If the routes are asymmetric, this assumption breaks down. In such a case the two hosts can keep consistent time internally but not between each other.

A second example [Pax96a] is protocols by which connection end-points infer network conditions from the pattern of packet arrivals they observe (e.g., by timing the arrival of acknowledgments [Kes91]). If routing is symmetric the bandwidth observed in the arrival of acks is the same as the bandwidth of the outgoing link. This could allow servers to determine the link bandwidth available for replying to client requests. If routing is not symmetric then the server cannot determine the correct value. Routing symmetry is also necessary if routers are to set up *anticipatory flow state* for replies to requests which pass through them. For example, when A sends a connection request to B through router R, then R might set up flow state for the reply from B to A. However if the route is not symmetric and the reply path does not contain R, then the anticipatory flow to A is wasted.

Sources of Routing Asymmetry: The two most important causes of routing asymmetry are the absence of non-unique shortest paths and Policy Routing. Routers use Bellman-Ford or Link State routing [Per92] algorithms to calculate routing tables. When multiple routes have the same cost, each router picks a route using an arbitrary tiebreaker. This leads to asymmetry in routes between two nodes A and B if the router closest to A and the router closest to B pick different routes. We describe an example below. We show how we can avoid this asymmetry by modifying route selection algorithms to include a random integer link cost which serves to arbitrate in case of ties.

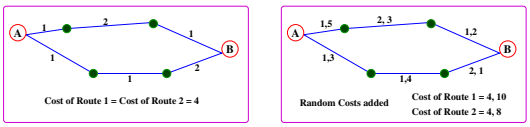
The Internet today consists of domains inter-connected by ISPs (*Internet Service Providers*). These ISPs are commercial organizations which charge for the service they provide. This structure of the Internet leads to two important classes of protocols: intra-domain, those that are used within a domain, e.g. RIP [Hed88], and inter-domain, those that used across domains, e.g. BGP [RL95].

While the lack of a unique shortest path is probably an important source of route asymmetry within domains, the principal source of asymmetries in backbone routers is policy routing. For example [Pax96a] in *Hot Potato* routing, suppose host A in ISP_A wants to send a packet to host B in ISP_B which is, say, at the other end of the US, and both ISP_A and ISP_B provide connectivity across the US. Then ISP_A might like to "drop" the packet to ISP_B as soon as possible because it might like ISP_B to carry the packet along the costly trans-country link. For reverse traffic, ISP_B will return the favor. A third cause of routing asymmetry is *Adaptive Routing* in which a router shifts traffic from a highly loaded link to a less loaded one, or load balances across multiple paths.

A. Avoiding Routing Asymmetry

The only way to prevent asymmetries due to Policy Routes is to have policies that ensure symmetrical routes based on some agreement between ISPs (implausible). We can, however, prevent asymmetry arising from non-unique shortest paths in intra-domain routing protocols by using random link costs (described later). This is useful for creating symmetrical intra-domain routes. For example, protocols such as NTP [Mil85] can benefit when symmetrical routing is used within a domain.

One way to ensure route symmetry is to use a symmetrical tiebreaker. For example, we could use a sorted list of node IDs in a path as a tiebreaker (treated as a string for lexicographic comparison) for choosing among several equal-cost paths. However, the overhead of sorting node IDs increase the complexity of Dijkstra's algorithm from $O(N \log N)$ to $O(N^2 \log N)$, where N is the number of nodes. For distance vector protocols, there will be the additional message overhead of carrying sorted path lists.



(a) (b)

Fig. 1. Random costs can break route asymmetry

We introduce a new technique that is simpler and much more efficient than the sorted list method. The idea is to use an additional (small) random number component along with the usual link cost (Figure 1). Each link between two nodes gets a small integer random cost assigned either manually by a manager, or automatically by a leader node elected on the link. The

sum of the random costs serves as a tiebreaker. When there are two or more paths with the same cost (between nodes *A* and *B* in Figure 1(a)), then the path with the least random cost (route 2, Figure 1(b)) is selected.

Note that adding random costs does not change the convergence behavior of Bellman-Ford.[2] The random costs are used only for breaking ties between two equal cost paths. The actual and random costs are added and stored separately, and only the actual cost is compared to *infinity* when checking for convergence. See Section III for details.

This method leaves the complexity of route computation unchanged and only requires that routing messages carry the extra random component along with the normal link cost. Conceptually routing is unchanged except that instead of using a single number for the cost of a link or path we need to use a tuple (c, r) to represent the cost (Figure 1(b)); c is the usual cost and r is the random cost. We compare tuples lexicographically: $(c, r) < (c', r')$ if $c < c'$ or $c = c'$ and $r < r'$. Except for this change, route computation in Bellman-Ford or Dijkstra algorithms remains unchanged.

Our new method of adding random costs raises an interesting question. How big should the random numbers be to make the probability of non-unique shortest paths sufficiently low? Below, we present a theoretical analysis followed by simulation results which show that 10 bit random numbers produce good results.

B. Probabilistic Analysis of the Random Cost Algorithm

Consider an arbitrary graph with arbitrary (but symmetric) link costs. Suppose we choose random link costs uniformly in the range $\{1, ..., c\}$. What is the probability that there is a non-unique shortest path? In its fullest generality this is a very hard problem, because even enumerating the set of shortest paths for a given graph is difficult.

Our key insight, which allows us to bound the required probability, is to use a powerful *Isolating Lemma* due to Mulmuley, Vazirani, and Vazirani [MR95]. Originally, the Isolating Lemma was invented to calculate the probability that a graph would contain a unique *perfect matching* given a random assignment of *node* weights. We, however, use it to calculate the probability that a graph will contain *all unique shortest paths* given a random assignment of *edge* weights.

Consider any set X of m elements. Suppose that each element is assigned an integer random cost chosen uniformly and independently in the range $\{1, ..., 2m\}$. Consider any family F of subsets of X. The cost of a subset $S \in F$ is the sum of the weights of the elements of X contained in S. The Isolating Lemma states that the probability that there is a unique minimum weight subset in F is at least $1/2$. This lemma is surprising because it works regardless of the way we choose the subsets F of X. A proof can be found in [MR95].

To use the Isolating Lemma, we let X be the set of all links. Thus m is the number of links in the graph. We take F to be the the set of all shortest paths between a certain pair of nodes. Our objective is to find the probability of a unique minimum cost(weight) path in F for the source-destination pair in question. It follows directly from the Isolating Lemma that if we choose random integer link costs for each link in the range $\{1, ..., 2m\}$, then the probability of a shortest path is at least $1/2$.

Since $1/2$ is too high a probability of failure, we generalized the Isolating Lemma slightly. The generalization shows that if the random link costs are chosen uniformly and independently in the range $\{1, ..., km\}$, the probability of a unique minimum weight shortest path is at least $1-1/k$. This shows that that we can make the probability of a unique shortest path as high as we like by increasing the number of bits allocated to the random cost. For example, the analysis indicates that in a 128 link network, we can get roughly 90 percent probability of a unique shortest path using 10 bit random costs.

While it is satisfying to obtain a theoretical estimate of the probability that is completely independent of the network topology, we note that the the theoretical result is a gross underestimate of the real success probability. For example, we are dealing only with a specific family of subsets (shortest cost paths between a particular source-destination pair) and not all possible subsets. Our simulation results, presented below, show much better results.

C. Simulation Results

We simulated *Bellman-Ford* route computation, as used by the *RIP* protocol, over a few sample Internet topologies. We used a subset of the topology of Internet sites used for routing measurement in [Pax96a],

[2]The convergence of Bellman-Ford often depends on the maximum cost of a route. Thus, changing the cost metric could, without care, affect convergence time.

[Pax96b]. This sample topology (Figure 2) consisted of 23 nodes, 13 from North America, 7 from Europe, 2 from Australia, and 1 from Asia. The random integer costs were chosen from a space of 1 through the number of nodes. We also used a subset of the Swiss Academic & Research Network to simulate a reasonably complex domain. The Swiss topology (Figure 3) consisted of 19 nodes. Table I lists the location of the sites in our sample Internet topology (Figure 2). By using real Internet topologies, we hope to make our analysis and results pertain to the Internet as closely as possible.

The Distance-Vector algorithm was simulated several times over these two topologies, and we measured the number of asymmetrical routes in each topology. Table II shows the percentage of symmetrical routes obtained for the different topologies with random costs. With no random numbers, our algorithm yielded about 8 route asymmetries for the Internet topology and about 12 for the Swiss one. The use of 10 bit random numbers eliminated all route asymmetries.

We also looked at a few local domain topologies but most of them had a *tree* topology, and hence no asymmetries. Thus, we believe that 10 bit random numbers should suffice for most topologies. The above link cost modification can be used to make any shortest path routing algorithm symmetric. In the next section we describe in detail how we can modify the popular intra-domain protocol RIP [Hed88].

III. Modifications to Routing Information Protocol

RIP is a Bellman-Ford (or Distance-Vector) protocol in which each router calculates the shortest cost to other routers, distributes the new costs to its neighbors, with the process continuing till all costs stabilize. In this section we show how we can modify RIP to incorporate random costs. Figure 4 shows the message format of RIP version 2 as described in RFC [Mal94] except for one small modification.

In standard RIP, the cost metric for routes is a 32 bit field. However the value of *infinity* is only 16, which requires 5 bits. Therefore we propose reducing the cost metric to 16 bits (Figure 4). We use the lower order 16 bits for the random costs. By reading the two fields as a single 32 bit integer, cost comparisons still take a single step. Since the actual cost is in the higher 16 bits, the actual cost will get preference when comparing costs; the lower 16 bits of random cost will break ties. However our scheme requires us to redefine the value of *infinity* to $16 * 2^{16}$ because we have left shifted the cost metric by 16 bits. Note that we have not affected convergence times. To prevent overflow between random and actual cost fields, we use up to 12 bit random link costs (our analysis suggested 10 bits was sufficient); given a maximum of 16 hops, the random link cost field will never overflow.

A simple way to ensure interoperability and backward compatibility is to break the original domain into two domains connected by (say) a BGP router. We gradually enlarge the domain containing the new RIP by reprogramming one router at a time in the domain containing the old RIP routers. When the transition is complete, we return to a single domain.

IV. Conclusions

While there is little one can do to prevent asymmetries due to policy routes unless policy makers cooperate, we have shown that a simple new technique of adding random cost tiebreakers can avoid non-unique shortest paths in shortest path protocols. Such a technique could be used within a domain to improve the performance of NTP, packet-pair and flow state set up protocols within domains. However, the major contribution of this paper is not to argue that symmetry is a pressing need but to show that, perhaps contrary to popular belief, routing symmetry can be added quite simply to shortest path protocols.

References

[Hed88] C. Hedrick. Routing Information Protocol. *STD 34, RFC 1058*, June 1988.

[Kes91] S. Keshav. A Control-Theoritic Approach to Flow Control. In *Proceedings of the SIGCOMM '91 Symposium*, pages 3–15, September 1991.

[Mal94] G. Malkin. RIP Version 2 Carrying Additional Information. *Xylogics Inc., RFC 1723*, November 1994.

[Mil85] D.L. Mills. Network Time Protocol (NTP). *RFC 958*, September 1985.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Pax96a] Vern Paxson. *An Analysis of End-to-End Internet Dynamics*. (Partial Draft) ftp://ftp.ee.lbl.gov/.vp-routing.long.ps.Z, 1996.

[Pax96b] Vern Paxson. End-to-End Routing Behavior in the Internet. In *Proceedings of the ACM SIGCOMM '96 Symposium*, volume 26, number 4, pages 25–38, August 1996.

[Per92] R. Perlman. *Interconnections: Bridges and Routers*. Addison-Wesley, 1992.

[RL95] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). *RFC 1771*, March 1995.

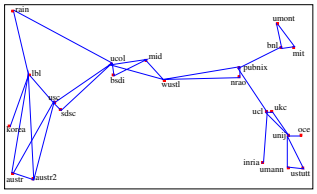


Fig. 2. A Sample **Internet** topology consisting of 23 nodes.

Sl.	Name	Location	Sl.	Name	Location
1	rain	Portland, Oregon	13	umont	Montreal, Canada
2	lbl	Berkeley, CA	14	ucl	London, U.K.
3	usc	Los Angeles, CA	15	ukc	Canterbury, U.K.
4	sdsc	San Diego, CA	16	unij	Nijmegen, The Netherlands
5	ucol	Boulder, CO	17	oce	Venlo, The Netherlands
6	bsdi	Colorado Springs, CO	18	umann	Mannheim, Germany
7	mid	Lincoln, Nebraska	19	ustutt	Stuttgart, Germany
8	wustl	St. Louis, MO	20	inria	Sophia, France
9	pubnix	Fairfax, VA	21	korea	Pohang, South Korea
10	nrao	Charlottesville, VA	22	austr	Melbourne, Australia
11	mit	Cambridge, MA	23	austr2	Newcastle, Australia
12	bnl	Brookhaven, NY			

TABLE I

List of Sites used in the above sample **Internet** topology

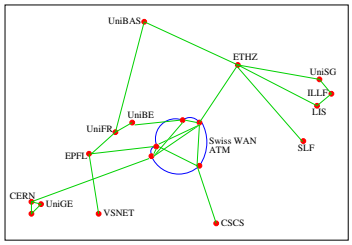


Fig. 3. A **Swiss Network** topology consisting of 19 nodes selected from the Swiss Academic & Research Network.

Random Number Space	Topology	
	Sample Internet	**Swiss Network**
19	-	88.2
23	98.1	-
8 bits	99.95	98.95
10 bits	100	100

TABLE II

Percentage of symmetrical routes obtained for different topologies. 10 bit random numbers appear to be sufficient for most topologies.

Command(1)	**Version(1)**	**unused(2)**
Address Family Identifier(2)		**Route Tag (2)**
IP Address (4)		
Subnet Mask (4)		
Next Hop (4)		
Metric (2)		**Random Cost (2)**

Fig. 4. Proposed RIP Message Format; note the change in the last word. The value of *infinity* is left shifted by 16 bits to keep convergence unchanged. The random cost is also prevented from overflowing onto the actual cost fields.