IDN-10

Routing to Exterior Networks in the Next Generation Gateway

Eric Rosen

October 1983

Let's look at what is required in order to make our interior routing algorithm deal properly with exterior nets.

Consider what happens when a packet destined for an exterior net reaches its entry (interior) gateway. Remember, the destination net is considered to be an exterior net just in case there is no interior gateway whose most recent routing update indicates that the gateway is interfaced to that network. There are three cases to consider:

a - The gateway has an exterior neighbor which is an appropriate first hop for this exterior net. Then the packet can just be forwarded to the exterior neighbor (using the local protocol of the directly connected net). If the source host is on the net over which the interior and exterior gateways are neighbors, a redirect should be sent to the source host.

b - The gateway does not know any way of reaching that network. An ICMP destination net unreachable should be sent to the source host. There is no distinction here between exterior and interior net.

c - The gateway has no exterior neighbor which is an appropriate first hop for this exterior net. However, there is at least one interior gateway somewhere remote in this autonomous system which does have an exterior neighbor which (according to its EGP NR message) is an appropriate first hop for this exterior net. Then the entry gateway should choose one of these other interior gateways as the exit gateway, and route the packet to it. (Of course, it should try to choose an interior gateway which is up.)

Note that the only use of the exterior net info is to enable the entry gateway to map exterior destination nets to an interior exit gateway. The data is then routed according to the ordinary interior routing algorithm to the exit gateway, which then forwards it to one of its exterior neighbors.

This method of routing pictures the interior exit gateway as being "directly connected" to the exterior net. That is, from the perspective of the interior routing algorithm, there is no real difference between the interior nets to which a gateway is really interfaced, and the exterior nets for which it has exterior neighbors who claim to be appropriate first hops.

Why then do we make any distinction at all between interior and exterior nets? We have a commitment to perform routing to the interior nets in a timely, accurate, and responsive manner. We have much less of a commitment to the exterior nets. Since we do not have gateways of our own system connected to the exterior nets, we cannot guarantee the accuracy of any information we receive about them. We cannot guarantee that data directed to exterior nets will actually arrive there. We cannot control the amount of overhead that would be needed to keep very responsive information about those networks. If a particular interior net keeps moving from gateway to gateway, we are committed to being always able to figure out how to reach it, and to using a path which is chosen so as to satisfy some sort of "optimality" requirement. There is no such commitment with respect to exterior nets. EGP is not a routing algorithm.

We need to discuss the way in which interior gateways will learn that other interior gateways are potential exit gateways for packets addressed to certain exterior nets. First, some terminology. We will say that an interior gateway IG "borders" an exterior net EN if it is an exterior neighbor (direct or indirect) of a gateway EG which claims via EGP to be an appropriate first hop to EN. If IG and EG are exterior neighbors over net IN, then we say that IG borders EN via IN.

Note that only the gateways which border a particular exterior net actually have to know how to address the exterior gateways which are used to reach that network. Gateways which do not border this net have no need for this knowledge. All they really have to know is which interior gateways border which exterior nets.

The simplest mechanism would be to have the routing updates from gateway G list not only the (interior) nets that it is interfaced to, but also the exterior nets that it borders. This method has one insuperable problem, however. The number of exterior nets can grow almost without bound, and we certainly don't want our routing updates to be able to grow arbitrarily large. Hence we need some separate protocol.

If we can assume the transitivity of the "is a neighbor of" relation, we would like to ensure that whenever two interior gateways are neighbors on net N, that they both border exactly the same set of exterior nets. (This prevents data from having to travel extra gateway hops on the same network.) This follows immediately if we can assume that both gateways have the same set of exterior neighbors. Of course, a given exterior gateway might be a "direct" neighbor of one of the interior gateways and an "indirect" neighbor of the other. If some of our gateways are to be indirect neighbors of exterior gateways, then we need a protocol between interior neighbors which passes information similar to that in the EGP NR message.

What is really needed here is for any gateway IG which has a DIRECT exterior neighbor EG on net N to to inform all its interior neighbors on net N that they have EG as an indirect exterior neighbor. IG also has to inform its interior neighbors as to the exterior nets for which EG is an appropriate first hop. That is, a gateway needs to send to all its interior neighbors a list of its direct exterior neighbors and the exterior nets to be reached via those exterior gateways. Call this an Indirect Neighbor Update (INU). Note that INU updates are not part of the EGP protocol, but rather of an interior updating protocol, and they contain less information than NR messages might contain. These INU updates need to be acknowledged by each interior neighbor, and should probably just be sent directly to them; there doesn't appear to be any need for flooding, or any reason to treat next door neighbors differently than other neighbors. Sequencing and duplicate detection are needed though. The most recently used sequence number should be saved in non-volatile memory. When a gateway sees a neighbor come up, it should send it its most recent INU update, if the neighbor has not already acknowledged it.

Under this procedure, a gateway with no direct exterior neighbors will never send any INU updates.

INU updates should probably be sent at about the same maximum rate as EGP polls. However, they should be sent only when necessary, to report some change detected as a result of NR messages.

When a gateway sees a new neighbor (or when it comes up), it needs to acquire the INU information, just as it needs to acquire the interior routing information.

It is possible that if two neighboring interior gateways G1 and G2 are both direct exterior neighbors with the same exterior gateway G3, their INU updates may contradict each other. This can happen if their INU polling cycles of the exterior neighbor are not in sync, as in general they will not be. Eventually, G1 and G2 should come to agreement, but there may be some period of minutes during which one has more up-to-date information than the other. (Of course, if the information in G3's NR message is in constant flux, the interior gateways may never get consistent information, but that is the fault of G3 itself.) We ought to adopt the rule that a gateway receiving an INU update ignore any information in that update about exterior gateways with which it is a direct neighbor. This prevents "direct" information from being overwritten by "indirect". However, it doesn't help an interior gateway G4 which is not a direct exterior neighbor of G3 to determine whether to believe G1 or G2. We have two options here. Either G4 should believe whichever of G1 or G2 has sent it an INU update most recently, or it should believe one of them according to some arbitrary rule. Probably it doesn't make too much difference.

The just discussed "INU update" procedure ensures that all interior neighbors on net N border the same set of exterior networks. Now in order for remote interior gateways to be able to route to exterior nets, it is only necessary that some ONE of the gateways on net N send to all interior gateways not on net N a list of the exterior nets which can be reached via exterior gateways which are on net N. These are the exterior nets for which that gateway (and its interior neighbors on net N) are potential exit gateways. This list of exterior nets will be called an Exterior Network Update (ENU). The remote gateways can assume that if two interior gateways are neighbors on net N, then they are potential exit gateways for the same set of exterior networks.

This sort of update message really just consists of a gateway identifier, the network number of the net which has the exterior neighbors, and a list of network numbers of exterior nets. If a particular gateway has exterior neighbors on two nets, then it may have two such update messages to send. We should probably consider them to be two separate messages, with separate sequence number spaces.

Of course, every gateway could send one of these update messages, but that would use a lot of overhead and would only result in sending a lot of redundant information. We could decide, say, that the gateway on net N that believes it has a smaller identification number than any of its neighbors on that net should be the one to send this update. If net N is partitioned, this algorithm would cause one net N gateway from each segments to send the ENU, which is just what we need.

Should all the gateways which receive an ENU from G1 figure out for themselves which other gateways border the same exterior nets as G1, or should the ENU itself carry a list of the other gateways which border the same nets? If we are really willing to make the commitment that all interior gateways which are neighbors on net N border the same set of exterior nets via net N, and I think this commitment is important to avoid unnecessary extra hops, then the former procedure is both lower in overhead and more responsive, a rare combination of virtues.

It is possible that some race condition will cause two net N gateways to send an ENU, even in the absence of a partition. Unless the ENUs happened to be identical, we would have to decide which to believe. Since there is no rational way to choose, I suggest believing the union. That is, assume that all the gateways neighboring the source of the ENUs border all the exterior nets which are mentioned in either.

How should we disseminate these ENUs, which might in general be quite lengthy?

If we are really going to start talking of having to handle tens of thousands of nets, we should be thinking in terms of having one or more "name servers", which get these updates and store the information in them. Then when an entry gateway gets a packet destined for an exterior net, it could query the name server for the identity of the exit gateway. Each gateway could keep a cache of the exit gateways for the most recently used exterior nets. For reliability, the name servers could be gateways, and each gateway could have the ability to become a name server. The need to do a query would amount to a "connection setup delay".

If we are talking of a total of less than 1000 nets, we would probably have trouble selling this connection-oriented mechanism to the ICCB. When the number of networks gets really huge, the overhead involved in keeping every gateway informed of the potential exit gateway for every net would be prohibitive. Until the overhead gets prohibitive, it may be felt that the additional delay of using name servers is not worthwhile. (However, I don't see any way around that in the future. I find it somewhat worrisome that the number of internet addresses that need separate treatment in routing is growing so rapidly, and there seems to be little realization that this causes a problem in the gateways.)

It is probably fair to require that no gateway initiate an ENU update more often than once every few minutes (maybe as much as 10 minutes?). After all, one of the points of EGP is to enable an autonomous system to control its own overhead, which might mean not sending immediate updates every time some exterior net or gateway flaps.

Probably the best thing is to use a flooding protocol, similar to that used for the interior routing updates. However, the situation is complicated by the fact that the exterior updates may be of arbitrary length, and we can't count ever on their fitting into a single packet. However, there is no reason why an entire exterior net table needs to be in a single packet.

If we fragment exterior updates, we can use IP reassembly to put them back together before processing them, or we use a technique similar to that which I originally proposed for EGP, i.e., process each fragment separately. If we are committed to performing IP reassembly anyway, maybe we should just use the former, since that could make the fragmentation/reassembly more or less transparent to the actual updating protocol. That is, the fragmentation and reassembly would be done below the level of the updating protocol, which itself would be very similar to that used with the interior updates. (Though we would probably want to run the interior protocol at a higher priority.)

Alternatively, we could process the fragments of a particular update as they arrive, since each fragment is independently process-able. That is, we could use a technique similar to that which I first proposed for EGP. Each exterior update would then have both an update sequence number (similar in function to that of the exterior updates) and a fragment number. To make this work, however, there needs to be some fixed rule about how to divide an exterior update into fragments, so that retransmissions cause the same set of fragments to get retransmitted. This could turn out to be very tricky.

General IP reassembly is also not without its subtleties, however. If a couple of neighbors are sending us fragments of the same update, we may get one copy of the update reassembled before the other. Then we should ack it to both neighbors, and remember to discard the unused fragments we have been holding.

In general, if we are doing IP reassembly in the gateway, we need to decide when to throw the unreassembled fragments away. However, this is probably the simplest thing, if we are going to implement reassembly anyway. It may not be very efficient or very rapid, but, after all, it is only the exterior net information that may be held up awaiting reassembly; it's not as if our interior routing information is held up. The only problem that arises if the exterior net information is delayed is that some users won't be able to reach some exterior nets for awhile. As we start to see hundreds of networks, we are just going to have to face the fact that not every gateway in the world will be able to get timely information about every network, or else there will be no internet bandwidth left for user data.

One might wonder why the updates cannot be made smaller by having them contain incremental information, i.e., by just indicating which exterior nets have come or gone since the last update. However, if some incremental update failed to reach some gateway, then it would never know what was in that update. If we require complete information in each update, then a gateway receiving an update can just wipe out all information previously received from the source of that update, and believe only what is in the current update; this is a more robust procedure.

When a gateway sees a neighbor come up, it should, by analogy with the interior updating algorithm, send the neighbor all the updates it has received but has not been able to transmit to the neighbor. To avoid an enormous spike in overhead, I think we have no choice but to send these updates relatively slowly. This means that when a gateway comes up, or a partition ends, the system is relatively slow pass around information about how to reach exterior networks. This is unfortunate, and will doubtless cause some number of user gripes, but I don't really see any alternative.

There are some things we can do to ameliorate the general non-responsiveness of exterior routing information. For example, suppose a gateway G1 receives an ENU which indicates that gateway G2 no longer borders some exterior network EN which it previously bordered. (Since the updates simply list the networks which G2 does border, G1 comes to this conclusion by noting that EN was listed in a previous update from G2, but is not listed in the current update.) G1 should not immediately obliterate all knowledge of EN from its memory. Rather, it can just mark it as "probably down", or something like that. Then if G1 gets a packet destined for EN, and there is no other way it knows of to reach EN, it can send it to G1 anyway. The worst that can happen is that ICMP deads get sent back to the source host. On the other hand, if EN has come back up and is bordering G1 again, this enables faster restoration of service.

This will not work so well if a particular exterior network keeps moving around, now bordering one set of interior gateways, now bordering another. Providing responsive routing to a network which appears to be moving like that will require interior routing. I think these procedures do accord well with the main goal of EGP, namely to provide service through stub gateways.

A further improvement in exterior routing service can be made as follows. Suppose entry gateway G1 routes a packet for exterior net EN via exit gateway G2, but that when the packet gets there, G2 is not bordering net EN. Then G2 must send an ICMP net unreachable back to the source host. Gateway-directed routing allows G2 to route the ICMP dead message back via the original entry gateway G1. G1 can intercept this message before sending it on to the source host, and mark EN as not reachable via G2. This indication can be timed out after a few minutes, so that G1 will try G2 periodically. This procedure might significantly increase the responsiveness of the exterior routing, without requiring large increases in overhead.

The same sort of procedure can be applied more generally to ICMP destination host dead messages. This would enable additional host dead messages referencing the same host to be sent by the entry gateway, rather than by the exit gateway, resulting in a further reduction in overhead and preventing useless traffic from entering the system.

Here is a different sort of issue to consider. Suppose entry gateway G1 thinks that G2 borders exterior net N, and routes packets for N to G2. However, G2 does not believe it has any exterior neighbor which is an appropriate first hop for network N. G2 could just drop the packet and respond with an ICMP destination net unreachable, or it could try to re-route the packet, if it thinks it knows of another possible exit gateway. I tend to favor the former procedure (drop the packet), since it avoids the possibility of routing loops if the exterior net information is not completely consistent among all the interior gateways. It is not that hard to imagine that for a period of time, G1 and G2 each believe the other to be the appropriate exit gateway for net N, while in fact neither is. If each gateway starts re-routing packets for net N via the other, we are in trouble.