

IDN-9

Neighbors and Next-Door-Neighbors

Robert Hinden

October 1983

I want to discuss neighbors, next-door-neighbors, and neighbor discovery. The current gateways send pings and routing updates to their neighbor gateways. Gateways are said to be neighbors if they share a network. Gateways may be neighbors on more than one network. The current gateways suffer with two problems regarding neighbors.

The first is that all gateways ping all their neighbor gateways and themselves in order to tell if they are reachable. A ping consists of sending a GGP echo to a gateway and receiving a GGP echo reply from it. On a network with few gateways, this does not create any problems, but on a large network with many gateways, the pinging generates large amounts of traffic. The number of pings sent by gateways on a network is equal to N^2 , where N is the number of gateways.

There are currently about 20 gateways on the Arpanet which send a ping to themselves and each other gateway every 15 seconds. That results in 400 pings per 15 seconds, or about 2.3 million pings per day. If the number of gateways should grow to say 40, which is not too far fetched considering the growth in the past year, there would be 9.2 million pings per day. The Arpanet is currently averages about 20 million messages per day. If there are 40 gateways, then the number of messages sent attributed to pings (a ping is equal to two messages) would almost equal the total amount of traffic the Arpanet sends today. Clearly this is an important problem.

The other problem the current gateways have regarding neighbors is neighbor discovery. The problem is that there isn't any. In order for a gateway to know about another gateway, one of the gateways must have been created with a neighbor table containing that gateway. The tables must be kept up to date and assembled in all of the gateways. Gateways will become neighbors with another gateway which it did not previously know about if that gateway starts sending GGP to it. This makes it reasonable for a new gateway to be added to the Internet system without reloading all of the other gateways. The problem comes up with

gateway that do not try to become neighbors with all other gateways. This includes Dumb gateway and EGP gateways.

I think it would be much better if the gateways could discover all other gateways on a network. The nicest solution would be for the network to tell the gateway about other gateways on the network. This, of course, might be possible on some networks someday, but will never be available on all networks. The next best solution is that if gateway A knew about gateway B and gateway B knew about Gateway C, then gateway A and C should be able to discover each other from information supplied by gateway B.

This will require a gateway to know about a few other gateways when it first comes up, but that is much easier to deal with than having to know about all other gateways. Once it begins communicating with these gateways it will learn of other gateways it can become neighbors with.

If we use gateway based routing in the new gateway, then the routing updates will contain information about gateways that are neighbors to the gateway that is sending the update. It should be easy for a gateway to discover another gateway by noticing

that it is mentioned in a routing update sent from another gateway. This will require that the updates contain the internet address in addition to the gateway number of the neighbor gateway in order to provide enough information for other gateways to do neighbor discovery. This will cause the updates to get very large if there are lots of neighbor gateways on a large network. I think the solution to this problem is that routing updates should not contain all neighbor gateways, but only the gateways next-door-neighbors.

A next-door-neighbor is a subset of a gateways neighbors. I think that a gateway should become next-door-neighbors with two of its neighbors on each of its attached networks. The gateway will only send pings to it self and to its next-door-neighbors. The number of pings will now be $3*N$. This is much better than the N squared of the current gateways.

There should not be a problem with the size of routing updates if they only contain next-door-neighbors. The number of neighbors in the updates will be a maximum of two times the number of networks that a gateway attaches to. It is probably reasonable to assume that a gateway will not connect to more than

16 networks, so I do not think that the routing updates will become too large to traverse networks with small message sizes.

The gateways send Pings to themselves to test that there interface to a network still works and if they can reach the network. This is not necessary for networks that provide a reliable indication of if they are up or down or when there are low level protocols (e.g HDLC) which are also providing this function. The gateway should be able to assume that if it is communicating with its next-door-neighbors that it can reach the network. It is probably only necessary to ping itself if it doesn't have any neighbors or they are all down. If we can eliminate interface pinging then the number of pings will be $2*N$, which is better yet.

If the routing updates only contain the next-door-neighbors, then a gateway will have to receive updates from most of its neighbors in order to learn about all of its neighbors. This is not a problem because the gateway must receive routing updates from all other gateways, not just its neighbors, in order to compute its routing tree. The gateway will add neighbors as it learns about them from the routing updates.

Now comes the question of how should a gateway pick its next-door-neighbors. It is important that a gateway never become isolated because the other gateways don't agree to become neighbors with it. The strategy used should insure that all gateway will come to the same conclusion as who should become next-door-neighbors with whom. I am not sure that it is necessary for gateways to agree on who should be next-door-neighbors. It might be alright if A is a next-door-neighbor with B, but B is not a next-door-neighbor with A, as long as all gateways have next-door-neighbors and no gateway is left out, but I think it is probably better if they agree.

A simple strategy to pick next-door-neighbors would be to pick the gateways that are up with the gateway number closest a gateway. For instance, if there are gateway numbered 1, 2, 3, 4, 5, and 6, then gateway 5 would pick gateways 4 and 6 as next-door neighbors, gateway 6 would pick gateways 5 and 1, etc. This strategy works but does not optimize for network topology or delay. We might devise better strategies that pick the nearest gateway, based on hops and/or delay, to become next-door-neighbors, but I think that the simple strategy is fine for a first cut.

The next area I want to discuss is the procedures a gateway uses to acquire next-door-neighbors and how to switch next-door-neighbors. First, I should mention the ways that a gateway can learn about new neighbor gateways. These are:

- 1) Receiving a Routing update from a gateway on an attached network.
- 2) Information in routing updates sent from other neighbor gateways.

Note that the SPF routing updates are received from all other gateways, not just the gateways neighbor gateways. It is only the updates received from neighbor gateways which will provide new neighbor information.

When a gateway learns about a new neighbor it should add it to its neighbor table and recompute who its next-door-neighbors should be. in order for a gateway to pick next-door-neighbors it must first find neighbor gateways that are up. The above mentioned ways insure that the neighbors are up because the information is supplied from the neighbor itself or from another gateway which is communicating with that gateway. If the new neighbor gateway becomes the next-door-neighbor, then the gateway

should try to bring it up as a next-door-neighbor. I don't think it necessary that two gateway are required to agree to become next-door-neighbors. A gateway sends pings to its next-door-neighbors to determine if they are up and sends routing updates describing its next-door-neighbors. As long as the procedure to determine a gateways next-door-neighbor gives consistant results based on the same input and the the routing updates containing the neighbor information are flooded quickly and reliably to provide that input, the gateways should agree who their next-door-neighbors are.

When a gateway comes up it should know about few gateways on each of its attached networks. For the gateway to break partitions it must know about gateways on each side itself. There may be some ways around this, but for now I think we should assume it as a given. Also note that networks which support logical addressing or a multicast type of broadcast make this much simpler. In the first case, the gateway coming up can send to a "generic" gateway address on that network to find a neighbor and in the second case it can send to a "generic" address to send to all gateways on that network.

The gateway coming up should send a message to one or more of its neighbor gateways which I will call "I am a Gateway". It does this to inform the neighbor gateway that it is a gateway and that it wants to receive routing updates to learn about its neighbors and the internet. When a gateway receives this message it knows that a new gateway is coming up. I don't think that it should install it as a neighbor when it receives this message or immediately send it routing updates. The gateway receiving this message should execute a coming up procedure to see if the new gateway is really up and it can be reached reliably. This should probably be "K out of N" pings, but the details are not important here.

If the new gateway does not come up successfully, then it should be forgotten about. It is the responsibility of the new gateway to make contact again. If the new gateway does come up, then the gateway receiving the "I am a Gateway" message should send it a complete set of routing updates. I think the new gateway should not be installed as a neighbor until it begins to send routing updates itself. This will insure that traffic will not begin to flow until the gateway is really up.

After the new gateway receives routing updates, it can pick its next-door-neighbors and start send routing updates. When the other gateways receive these routing updates they will learn about the new gateway and decide if it is a new neighbor and next-door-neighbor. If it is the latter, then the gateways should drop the old next-door-neighbor and start using the new one.

When a gateways next-door-neighbor goes down, it should mark that gateway as down and pick a new next-door-neighbor. The simple procedure which I described earlier has the characteristic that the gateways which declare their next-door-neighbor down, will become next-door-neighbors with each other.

I think that if a gateway does not have a new next-door-neighbor to pick when one goes down (i.e. there are three or less gateways on a network), then it should continue to try and bring it up. This should help to quickly break partitions.