Network Working Group
Request for Comments: 1433

J. Garrett
AT&T Bell Laboratories
J. Hagan
University of Pennsylvania
J. Wong
AT&T Bell Laboratories
March 1993

Directed ARP

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. Discussion and suggestions for improvement are requested. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

A router with an interface to two IP networks via the same link level interface could observe that the two IP networks share the same link level network, and could advertise that information to hosts (via ICMP Redirects) and routers (via dynamic routing protocols). However, a host or router on only one of the IP networks could not use that information to communicate directly with hosts and routers on the other IP network unless it could resolve IP addresses on the "foreign" IP network to their corresponding link level addresses. Directed ARP is a dynamic address resolution procedure that enables hosts and routers to resolve advertised potential next-hop IP addresses on foreign IP networks to their associated link level addresses.

Acknowledgments

The authors are indebted to Joel Halpern of Network Systems Corporation and David O'Leary who provided valuable comments and insight to the authors, as well as ongoing moral support as the presentation of this material evolved through many drafts. Members of the IPLPDN working group also provided valuable comments during presentations and through the IPLPDN mailing list. Chuck Hedrick of Rutgers University, Paul Tsuchiya of Bell Communications Research, and Doris Tillman of AT&T Bell Laboratories provided early insight as well as comments on early drafts.

1. Terminology

A "link level network" is the upper layer of what is sometimes referred to (e.g., OSI parlance) as the "subnetwork", i.e., the layers below IP. The term "link level" is used to avoid potential confusion with the term "IP sub-network", and to identify addresses (i.e., "link level address") associated with the network used to transport IP datagrams.

From the perspective of a host or router, an IP network is "foreign" if the host or router does not have an address on the IP network.

2. Introduction

Multiple IP networks may be administered on the same link level network (e.g., on a large public data network). A router with a single interface on two IP networks could use existing routing update procedures to advertise that the two IP networks shared the same link level network. Cost/performance benefits could be achieved if hosts and routers that were not on the same IP network could use that advertised information, and exchange packets directly, rather than through the dual addressed router. But a host or router can not send packets directly to an IP address without first resolving the IP address to its link level address.

IP address resolution procedures are established independently for each IP network. For example, on an SMDS network [1], address resolution may be achieved using the Address Resolution Protocol (ARP) [2], with a separate SMDS ARP Request Address (e.g., an SMDS Multicast Group Address) associated with each IP network. A host or router that was not configured with the appropriate ARP Request Address would have no way to learn the ARP Request Address associated with an IP network, and would not send an ARP Request to the appropriate ARP Request Address. On an Ethernet network a host or router might guess that an IP address could be resolved by sending an ARP Request to the broadcast address. But if the IP network used a different address resolution procedure (e.g., administered address resolution tables), the ARP Request might go unanswered.

Directed ARP is a procedure that enables a router advertising that an IP address is on a shared link level network to also aid in resolving the IP address to its associated link level address. By removing address resolution constraints, Directed ARP enables dynamic routing protocols such as BGP [3] and OSPF [4] to advertise and use routing information that leads to next-hop addresses on "foreign" IP networks. In addition, Directed ARP enables routers to advertise (via ICMP Redirects) next-hop addresses that are "foreign" to hosts, since the hosts can use Directed ARP to resolve the "foreign" next-

hop addresses.

3. Directed ARP

Directed ARP uses the normal ARP packet format, and is consistent with ARP procedures, as defined in [1] and [2], and with routers and hosts that implement those procedures.

3.1 ARP Helper Address

Hosts and routers maintain routing information, logically organized as a routing table. Each routing table entry associates one or more destination IP addresses with a next-hop IP address and a physical interface used to forward a packet to the next-hop IP address. If the destination IP address is local (i.e., can be reached without the aid of a router), the next-hop IP address is NULL (or a logical equivalent, such as the IP address of the associated physical interface). Otherwise, the next-hop IP address is the address of a next-hop router.

A host or router that implements Directed ARP procedures associates an ARP Helper Address with each routing table entry. If the host or router has been configured to resolve the next-hop IP address to its associated link level address (or to resolve the destination IP address, if the next-hop IP address is NULL), the associated ARP Helper Address is NULL. Otherwise, the ARP Helper Address is the IP address of the router that provided the routing information indicating that the next-hop address was on the same link level network as the associated physical interface. Section 4 provides detailed examples of the determination of ARP Helper Addresses by dynamic routing procedures.

3.2 Address Resolution Procedures

To forward an IP packet, a host or router searches its routing table for an entry that is the best match based on the destination IP address and perhaps other factors (e.g., Type of Service). The selected routing table entry includes the IP address of a next-hop router (which may be NULL), the physical interface through which the IP packet should be forwarded, an ARP Helper Address (which may be NULL), and other information. The routing function passes the next-hop IP address, the physical interface, and the ARP Helper Address to the address resolution function. The address resolution function must then resolve the next-hop IP address (or destination IP address if the next-hop IP address is NULL) to its associated link level address. The IP packet, the link level address to which the packet should be forwarded, and the interface through which the packet should be forwarded are then passed to the link level driver

associated with the physical interface. The link level driver encapsulates the IP packet in one or more link level frames (i.e., may do fragmentation) addressed to the associated link level address, and forwards the frame(s) through the appropriate physical interface. The details of the functions performed are described via C pseudo-code below.

The procedures are organized as two functions, Route() and Resolve(), corresponding to routing and address resolution. In addition, the following low level functions are also used:

Get_Route(IP_Add,Other) returns a pointer to the routing table entry with the destination field that best matches IP_Add. If no matching entry is found, NULL is returned. Other information such as Type of Service may be considered in selecting the best route.

Forward(Packet,Link_Level_Add,Phys_Int) fragments Packet (if needed), and encapsulates Packet in one or more Link Level Frames addressed to Link_Level_Add, and forwards the frame(s) through interface, Phys_Int.

Look_Up_Add_Res_Table(IP_Add,Phys_Int) returns a pointer to the link level address associated with IP_Add in the address resolution table associated with interface, Phys_Int. If IP_Add is not found in the address resolution table, NULL is returned.

Local_Add_Res(IP_Add,Phys_Int) returns a pointer to the Link Level address associated with IP_Add, using address resolution procedures associated with address, IP_Add, and interface, Phys_Int. If address resolution is unsuccessful, NULL is returned. Note that different address resolution procedures may be used for different IP networks.

Receive_ARP_Response(IP_Add,Phys_Int) returns a pointer to an ARP Response received through interface, Phys_Int, that resolves IP_Add. If no ARP response is received, NULL is returned.

Dest_IP_Add(IP_Packet) returns the IP destination address from IP_Packet.

Next_Hop(Entry) returns the IP address in the next-hop field of (routing table) Entry.

Interface(Entry) returns the physical interface field of (routing table) Entry.

ARP_Helper_Add(Entry) returns the IP address in the ARP Helper Address field of (routing table) Entry.

ARP_Request(IP_Add) returns an ARP Request packet with IP_Add as the Target IP address.

Source_Link_Level(ARP_Response) returns the link level address of the sender of ARP_Response.

```
ROUTE(IP_Packet)
{
Entry = Get_Route(Dest_IP_Add(IP_Packet),Other(IP_Packet));
If (Entry == NULL)  /* No matching entry in routing table */
  Return;  /*  Discard IP_Packet */
else
  {  /* Resolve next-hop IP address to link level address */
  If (Next_Hop(Entry) != NULL) /* Route packet via next-hop router */
    Next_IP = Next_Hop(Entry);
  else  /* Destination is local */
    Next_IP = Dest_IP_Add(IP_Packet);
  L_L_Add = Resolve(Next_IP,Interface(Entry),ARP_Helper_Add(Entry));
  If (L_L_Add != NULL)
    Forward(IP_Packet,L_L_Add,Interface(Entry));
  else  /* Couldn't resolve next-hop IP address */
    Return;  /* Discard IP_Packet */
  Return;
  }
}
```

Figure 1: C Pseudo-Code for the Routing function.

```
Resolve(IP_Add,Interface,ARP_Help_Add)
{
If ((L_L_Add = Look_Up_Add_Res_Table(IP_Add,Interface)) != NULL)
  {   /* Found it in Address Resolution Table */
  Return L_L_Add;
  }
else
  {
  If (ARP_Help_Add == NULL)
    {  /* Do local Address Resolution Procedure */
    Return Local_Add_Res(IP_Add,Interface);
    }
  else  /* ARP_Help_Add != NULL */
    {
    L_L_ARP_Help_Add = Look_Up_Add_Res_Table(ARP_Help_Add,Interface);
    If (L_L_ARP_Help_Add == NULL)
                          /* Not in Address Resolution Table */
      L_L_ARP_Help_Add = Local_Add_Res(ARP_Help_Add,Interface);
    If (L_L_ARP_Help_Add == NULL)  /* Can't Resolve ARP Helper Add */
      Return NULL;  /*  Address Resolution Failed */
    else
      {  /* ARP for IP_Add */
      Forward(ARP_Request(IP_Add),L_L_ARP_Help_Add,Interface);
      ARP_Resp = Receive_ARP_Response(IP_Add,Interface);
      If (ARP_Resp == NULL) /* No ARP Response (after persistence) */
        Return NULL;  /* Address Resolution Failed */
      else
        Return Source_Link_Level(ARP_Resp);
        }
      }
    }
  }
}
```

Figure 2: C Pseudo-Code for Address Resolution function.

3.3 Forwarding ARP Requests

A host that implements Directed ARP procedures uses normal procedures to process received ARP Requests. That is, if the Target IP address is the host's address, the host uses normal procedures to respond to the ARP Request. If the Target IP address is not the host's address, the host silently discards the ARP Request.

If the Target IP address of an ARP Request received by a router is the router's address, the router uses normal procedures to respond to

the ARP Request. But if the Target IP address is not the router's address, the router may forward the ARP Request back through the same interface it was received from, addressed to a Link Level Address that corresponds to an ARP Helper Address in the router's routing table. The procedures used to process an ARP Request are described via C pseudo-code below. The function Receive() describes procedures followed by hosts and routers, and the function Direct() describes additional procedures followed by routers. In addition, the following low level functions are also used:

Is_Local_IP_Add(IP_Add,Phys_Int) returns TRUE if Phys_Int has been assigned IP address, IP_Add. Otherwise, returns FALSE.

Do_ARP_Processing(ARP_Request,Interface) processes ARP_Request using ARP procedures described in [2].

I_Am_Router returns TRUE if device is a router and False if device is a host.

Target_IP(ARP_Request) returns the Target IP address from ARP_Request.

Filter(ARP_Request,Phys_Int) returns TRUE if ARP_Request passes filtering constraints, and FALSE if filtering constraints are not passed. See section 3.4.

Forward(Packet,Link_Level_Add,Phys_Int) fragments Packet (if needed), and encapsulates Packet in one or more Link Level Frames addressed to Link_Level_Add, and forwards the frame(s) through interface, Phys_Int.

Look_Up_Next_Hop_Route_Table(IP_Add) returns a pointer to the routing table entry with the next-hop field that matches IP_Add. If no matching entry is found, NULL is returned.

Look_Up_Dest_Route_Table(IP_Add) returns a pointer to the routing table entry with the destination field that best matches IP_Add. If no matching entry is found, NULL is returned.

Link_Level_ARP_Req_Add(IP_Add,Phys_Int) returns the link level address to which an ARP Request to resolve IP_Add should be forwarded. If ARP is not used to perform local address resolution of IP_Add, NULL is returned.

Local_Add_Res(IP_Add,Phys_Int) returns a pointer to the Link Level address associated with IP_Add, using address resolution procedures associated with address, IP_Add, and interface, Phys_Int. If address resolution is unsuccessful, NULL is

returned. Note that different address resolution procedures may be used for different IP networks.

Next_Hop(Entry) returns the IP address in the next-hop field of (routing table) Entry.

Interface(Entry) returns the physical interface field of (routing table) Entry.

ARP_Helper_Add(Entry) returns the IP address in the ARP Helper Address field of (routing table) Entry.

Source_Link_Level(ARP_Request) returns the link level address of the sender of ARP_Request.

```
Receive(ARP_Request,Interface)
{
If (Is_Local_IP_Add(Target_IP(ARP_Request),Interface))
  Do_ARP_Processing(ARP_Request,Interface);
else  /*  Not my IP Address  */
  If (I_Am_Router)  /*  Hosts don't Direct ARP Requests  */
    If (Filter(ARP_Request,Interface))  /*  Passes Filter Test  */
                                        /*  See Section 3.4  */
      Direct(ARP_Request,Interface);  /*  Directed ARP Procedures  */
Return;
}
```

Figure 3: C Pseudo-Code for Receiving ARP Requests.

```
Direct(ARP_Request,Phys_Int)
{
Entry = Look_Up_Next_Hop_Route_Table(Target_IP(ARP_Request));
If (Entry == NULL)  /* Target_IP Address is not a next-hop */
  {                 /*  in Routing Table */
  Entry = Look_Up_Dest_Route_Table(Target_IP(ARP_Request));
    If (Entry == NULL)  /* Not a destination either */
      Return;  /* Discard ARP Request */
    else
      If (Next_Hop(Entry) != NULL) /* Not a next-hop and Not local */
        Return;  /* Discard ARP Request */
  }
If (Interface(Entry) != Phys_Int)
                          /* Must be same physical interface */
  Return;  /* Discard ARP Request */
If (ARP_Helper_Add(Entry) != NULL)
  {
  L_L_ARP_Helper_Add = Resolve(ARP_Helper_Add(Entry),Phys_Int,NULL);
  If (L_L_ARP_Helper_Add != NULL)
    Forward(ARP_Request,L_L_ARP_Helper_Add,Phys_Int);
      /*  Forward ARP_Request to ARP Helper Address  */
  Return;
  }
else  /*  Do local address resolution.  */
  {
  L_L_ARP_Req_Add =
              Link_Level_ARP_Req_Add(Target_IP(ARP_Request),Phys_Int);
  If (L_L_ARP_Req_Add != NULL)
    {  /*  Local address resolution procedure is ARP. */
       /*  Forward ARP_Request. */
    Forward(ARP_Request,L_L_ARP_Req_Add,Phys_Int);
    Return;
    }
  else
    {  /*  Local address resolution procedure is not ARP.  */
       /*  Do "published ARP" on behalf of Target IP Address  */
    Target_Link_Level =
                    Local_Add_Res(Target_IP(ARP_Request),Phys_Int);
    If (Target_Link_Level != NULL)  /*  Resolved Address  */
      {
      Forward(ARP_Response,Source_Link_Level(ARP_Request),Phys_Int);
      }
    Return;
    }
  }
}
```

Figure 4: C Pseudo_Code for Directing ARP Requests.

3.4 Filtering Procedures

A router performing Directed ARP procedures must filter the propagation of ARP Request packets to constrain the scope of potential "ARP floods" caused by misbehaving routers or hosts, and to terminate potential ARP loops that may occur during periods of routing protocol instability or as a result of inappropriate manual configurations. Specific procedures to filter the propagation of ARP Request packets are beyond the scope of this document. The following procedures are suggested as potential implementations that should be sufficient. Other procedures may be better suited to a particular implementation.

To control the propagation of an "ARP flood", a router performing Directed ARP procedures could limit the number of identical ARP Requests (i.e., same Source IP address and same Target IP address) that it would forward per small time interval (e.g., no more than one ARP Request per second). This is consistent with the procedure suggested in [5] to prevent ARP flooding.

Forwarding of ARP Request packets introduces the possibility of ARP loops. The procedures used to control the scope of potential ARP floods may terminate some ARP loops, but additional procedures are needed if the time required to traverse a loop is longer than the timer used to control ARP floods. A router could refuse to forward more than N identical ARP Requests per T minutes, where N and T are administered numbers. If T and N are chosen so that T/N minutes is greater than the maximum time required to traverse a loop, such a filter would terminate the loop. In some cases a host may send more than one ARP Request with the same Source IP address,Target IP address pair (i.e., N should be greater than 1). For example, the first ARP Request might be lost. However, once an ARP Response is received, a host would normally save the associated information, and therefore would not generate an identical ARP Request for a period of time on the order of minutes. Therefore, T may be large enough to ensure that T/N is much larger than the time to traverse any loop.

In some implementations the link level destination address of a frame used to transport an ARP Request to a router may be available to the router's Directed ARP filtering process. An important class of simple ARP loops will be prevented from starting if a router never forwards an ARP Request to the same link level address to which the received ARP Request was addressed. Of course, other procedures such as the one described in the paragraph above will stop all loops, and are needed, even if filters are implemented that prevent some loops from starting.

Host requirements [5] specify that "the packet receive interface between the IP layer and the link layer MUST include a flag to indicate whether the incoming packet was addressed to a link-level broadcast address." An important class of simple ARP floods can be eliminated if routers never forward ARP Requests that were addressed to a link-level broadcast address.

4. Use of Directed ARP by Routing

The exchange and use of routing information is constrained by available address resolution procedures. A host or router can not use a next-hop IP address learned via dynamic routing procedures if it is unable to resolve the next-hop IP address to the associated link level address. Without compatible dynamic address resolution procedures, a router may not advertise a next-hop address that is not on the same IP network as the host or router receiving the advertisement. Directed ARP is a procedure that enables a router that advertises routing information to make the routing information useful by also providing assistance in resolving the associated next-hop IP addresses.

The following subsections describe the use of Directed ARP to expand the scope of ICMP Redirects [6], distance-vector routing protocols (e.g., BGP [3]), and link-state routing protocols (e.g., OSPF [4]).

4.1 ICMP Redirect

If a router forwards a packet to a next-hop address that is on the same link level network as the host that originated the packet, the router may send an ICMP Redirect to the host. But a host can not use a next-hop address advertised via an ICMP Redirect unless the host has a procedure to resolve the advertised next-hop address to its associated link level address. Directed ARP is a procedure that a host could use to resolve an advertised next-hop address, even if the host does not have an address on the same IP network as the advertised next-hop address.

A host that implements Directed ARP procedures includes an ARP Helper Address with each routing table entry. The ARP Helper Address associated with an entry learned via an ICMP Redirect is NULL if the associated next-hop address matches a routing table entry with a NULL next-hop and a NULL ARP Helper Address (i.e., the host already knows how to resolve the next-hop address). Otherwise, the ARP Helper Address is the IP address of the router that sent the ICMP Redirect. Note that the router that sent the ICMP Redirect is the current next-hop to the advertised destination [5]. Therefore, the host should have an entry in its address resolution table for the new ARP Helper Address. If the host is unable to resolve the next-hop IP

address advertised in the ICMP Redirect (e.g., because the associated ARP Helper Address is on a foreign IP network; i.e., was learned via an old ICMP Redirect, and the address resolution table entry for that ARP Helper Address timed out), the host must flush the associated routing table entry. Directed ARP procedures do not recursively use Directed ARP to resolve an ARP Helper Address.

A router that performs Directed ARP procedures might advertise a foreign next-hop to a host that does not perform Directed ARP. Following existing procedures, the host would silently discard the ICMP Redirect. A router that does not implement Directed ARP should not advertise a next-hop on a foreign IP network, as specified by existing procedures. If it did, and the ICMP Redirect was received by a host that implemented Directed ARP procedures, the host would send an ARP Request for the foreign IP address to the advertising router, which would silently discard the ARP Request. When address resolution fails, the host should flush the associated entry from its routing table.

For various reasons a host may ignore an ICMP Redirect and may continue to forward packets to the same router that sent the ICMP Redirect. For example, a host that does not implement Directed ARP procedures would silently discard an ICMP Redirect advertising a next-hop address on a foreign IP network. Routers should implement constraints to control the number of ICMP Redirects sent to hosts. For example, a router might limit the number of repeated ICMP Redirects sent to a host to no more than N ICMP Redirects per T minutes, where N and T are administered values.

4.2 Distance Vector Routing Protocol

A distance-vector routing protocol provides procedures for a router to advertise a destination address (e.g., an IP network), an associated next-hop address, and other information (e.g., associated metric). But a router can not use an advertised route unless the router has a procedure to resolve the advertised next-hop address to its associated link level address. Directed ARP is a procedure that a router could use to resolve an advertised next-hop address, even if the router does not have an address on the same IP network as the advertised next-hop address.

The following procedures assume a router only accepts routing updates if it knows the IP address of the sender of the update, can resolve the IP address of the sender to its associated link level address, and has an interface on the same link level network as the sender.

A router that implements Directed ARP procedures includes an ARP Helper Address with each routing table entry. The ARP Helper Address

associated with an entry learned via a routing protocol update is NULL if the associated next-hop address matches a routing table entry with a NULL next-hop and NULL ARP Helper Address (i.e., the router already knows how to resolve the next-hop address). Otherwise, the ARP Helper Address is the IP address of the router that sent the routing update.

Some distance-vector routing protocols (e.g., BGP [3]) provide syntax that would permit a router to advertise an address on a foreign IP network as a next-hop. If a router that implements Directed ARP procedures advertises a foreign next-hop IP address to a second router that does not implement Directed ARP procedures, the second router can not use the advertised foreign next-hop. Depending on the details of the routing protocol implementation, it might be appropriate for the first router to also advertise a next-hop that is not on a foreign IP network (e.g., itself), perhaps at a higher cost. Or, if the routing relationship is an administered connection (e.g., BGP relationships are administered TCP/IP connections), the administrative procedure could determine whether foreign next-hop IP addresses should be advertised.

A distance-vector routing protocol could advertise that a destination is directly reachable by specifying that the router receiving the advertisement is, itself, the next-hop to the destination. In addition, the advertised metric for the route might be zero. If the router did not already have a routing table entry that specified the advertised destination was local (i.e., NULL next-hop address), the router could add the new route with NULL next-hop, and the IP address of the router that sent the update as ARP Helper Address.

4.3 Link State Routing Protocol

A link-state routing protocol provides procedures for routers to identify links to other entities (e.g., other routers and networks), determine the state or cost of those links, reliably distribute link-state information to other routers in the routing domain, and calculate routes based on link-state information received from other routers. A router with an interface to two (or more) IP networks via the same link level interface is connected to those IP networks via a single link, as described above. If a router could advertise that it used the same link to connect to two (or more) IP networks, and would perform Directed ARP procedures, routers on either of the IP networks could forward packets directly to hosts and routers on both IP networks, using Directed ARP procedures to resolve addresses on the foreign IP network. With Directed ARP, the cost of the direct path to the foreign IP network would be less than the cost of the path through the router with addresses on both IP networks.

To benefit from Directed ARP procedures, the link-state routing protocol must include procedures for a router to advertise connectivity to multiple IP networks via the same link, and the routing table calculation process must include procedures to calculate ARP Helper Addresses and procedures to accurately calculate the reduced cost of the path to a foreign IP network reached directly via Directed ARP procedures.

The Shortest Path First algorithm for calculating least cost routes is based on work by Dijkstra [7], and was first used in a routing protocol by the ARPANET, as described by McQuillan [8]. A router constructs its routing table by building a shortest path tree, with itself as root. The process is iterative, starting with no entries on the shortest path tree, and the router, itself, as the only entry in a list of candidate vertices. The router then loops on the following two steps.

1. Remove the entry from the candidate list that is closest to root, and add it to the shortest path tree.

2. Examine the link state advertisement from the entry added to the shortest path tree in step 1. For each neighbor (i.e., router or IP network to which a link connects)

   - If the neighbor is already on the shortest path tree, do nothing.

   - If the neighbor is on the candidate list, recalculate the distance from root to the neighbor. Also recalculate the next-hop(s) to the neighbor.

   - If the neighbor is not on the candidate list, calculate the distance from root to the neighbor and the next-hop(s) from root to the neighbor, and add the neighbor to the candidate list.

The process terminates when there are no entries on the candidate list.

To take advantage of Directed ARP procedures, the link-state protocol must provide procedures to advertise that a router accesses two or more IP networks via the same link. In addition, the Shortest Path First calculation is modified to calculate ARP Helper Addresses and recognize path cost reductions achieved via Directed ARP.

1. If a neighbor under consideration is an IP network, and its parent (i.e., the entry added to the shortest path tree in step 1, above) has advertised that the neighbor is reached via the same link as a network that is already on the shortest path

tree, the distance from root and next-hop(s) from root to the neighbor are the same as the distance and next-hop(s) associated with the network already on the shortest path tree. If the ARP Helper Address associated with the network that is already on the shortest path tree is not NULL, the neighbor also inherits the ARP Helper Address from the network that is already on the shortest path tree.

2. If the calculated next-hop to the neighbor is not NULL, the neighbor inherits the ARP Helper Address from its parent. Otherwise, except as described in item 1, the ARP Helper Address is the IP address of the next-hop to the neighbor's parent. Note that the next-hop to root is NULL.

For each router or IP network on the shortest path tree, the Shortest Path First algorithm described above must calculate one or more next-hops that can be used to access the router or IP network. A router that advertises a link to an IP network must include an IP address that can be used by other routers on the IP network when using the router as a next-hop. A router might advertise that it was connected to two IP networks via the same link by advertising the same next-hop IP address for access from both IP networks. To accommodate the address resolution constraints of routers on both IP networks the router might advertise two IP addresses (one from each IP network) as next-hop IP addresses for access from both IP networks.

5. Robustness

Hosts and routers can use Directed ARP to resolve third-party next-hop addresses; i.e., next-hop addresses learned from a routing protocol peer or current next-hop router. Undetected failure of a third party next-hop can result in a routing "black hole". To avoid "black holes", host requirements [5] specify that a host "...MUST be able to detect the failure of a 'next-hop' gateway that is listed in its route cache and to choose an alternate gateway." A host may receive feedback from protocol layers above IP (e.g., TCP) that indicates the status of a next-hop router, and may use other procedures (e.g., ICMP echo) to test the status of a next-hop router. But the complexity of routing is borne by routers, whose routing information must be consistent with the information known to their peers. Routing protocols such as BGP [3], OSPF [4], and others, require that routers must stand behind routing information that they advertise. Routers tag routing information with the IP address of the router that advertised the information. If the information becomes invalid, the router that advertised the information must advertise that the old information is no longer valid. If a source of routing information becomes unavailable, all information received

from that source must be marked as no longer valid. The complexity of dynamic routing protocols stems from procedures to ensure routers either receive routing updates sent by a peer, or are able to determine that they did not receive the updates (e.g., because connectivity to the peer is no longer available).

Third-party next-hops can also result in "black holes" if the underlying link layer network connectivity is not transitive. For example, SMDS filters [9] could be administered to permit communication between the SMDS addresses of router R1 and router R2, and between the SMDS addresses of router R2 and router R3, and to block communication between the SMDS addresses of router R1 and router R3. Router R2 could advertise router R3 as a next-hop to router R1, but SMDS filters would prevent direct communication between router R1 and router R3. Non-symmetric filters might permit router R3 to send packets to router R1, but block packets sent by router R1 addressed to router R3.

A host or router could verify link level connectivity with a next-hop router by sending an ICMP echo to the link level address of the next-hop router. (Note that the ICMP echo is sent directly to the link level address of the next-hop router, and is not routed to the IP address of the next-hop router. If the ICMP echo is routed, it may follow a path that does not verify link level connectivity.) This test could be performed before adding the associated routing table entry, or before the first use of the routing table entry. Detection of subsequent changes in link level connectivity is a dynamic routing protocol issue and is beyond the scope of this memo.

References

[1] Piscitello, D., and J. Lawrence, "The Transmission of IP Datagrams over the SMDS Service", RFC 1209, Bell Communications Research, March 1991.

[2] Plummer, D., "An Ethernet Address Resolution Protocol - or - Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", RFC 826, Symbolics, Inc., November 1982.

[3] Lougheed, K. and Y. Rekhter, "A Border Gateway Protocol 3 (BGP-3)", RFC 1267, cisco Systems and IBM T. J. Watson Research Center, October 1991.

[4] Moy, J., "OSPF Version 2", RFC 1247, Proteon, Inc., July 1991.

[5] Braden, R., editor, "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, USC/Information Sciences

Institute, October 1989.

[6] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.

[7] Dijkstra, E. W., "A Note on Two Problems in Connection with Graphs", Numerische Mathematik, Vol. 1, pp. 269-271, 1959.

[8] McQuillan, J. M., I. Richer, and E. C. Rosen, "The New Routing Algorithm for the ARPANET", IEEE Transactions on Communications, Vol. COM-28, May 1980.

[9] "Generic System Requirements In Support of Switched Multi-megabit Data Service", Technical Reference TR-TSV-000772, Bell Communications Research Technical Reference, Issue 1, May 1991.

Security Considerations

Security issues are not discussed in this memo.

Authors' Addresses

John Garrett
AT&T Bell Laboratories
184 Liberty Corner Road
Warren, N.J. 07060-0906

Phone: (908) 580-4719
EMail: jwg@garage.att.com

John Dotts Hagan
University of Pennsylvania
Suite 221A
3401 Walnut Street
Philadelphia, PA 19104-6228

Phone: (215) 898-9192
EMail: Hagan@UPENN.EDU

Jeffrey A. Wong
AT&T Bell Laboratories
184 Liberty Corner Road
Warren, N.J. 07060-0906

Phone: (908) 580-5361
EMail: jwong@garage.att.com