REIHE INFORMATIK
1/2000

Protocol Independent Multicast and Asymmetric Routing

Thomas T. Fuhrmann
Praktische Informatik IV
University of Mannheim
68131 Mannheim, Germany

February 1, 2000

Protocol Independent Multicast and Asymmetric Routing

Thomas T. Fuhrmann

University of Mannheim, Germany

Abstract

Originally all links in the Internet were assumed to operate bidirectionally. Like many other routing protocols, PIM (protocol independent multicast) is based on this assumption: as will be explained, PIM's concept of using the routers' unicast RIBs (routing information bases) for reverse-path-forwarding is not applicable in networks with uni-directional links. If an additional bidirectional link such as a dial-up connection exists, link-layer tunnelling can overcome these basic routing deficiencies. But in order to achieve a more efficient routing and sustain scalability we argue that multicast and unicast traffic should be distinguished either by an extended link-layer tunnelling or dual RIBs.

1 Introduction

Asymmetric Routes

Satellite links were among the first links being employed for Internet routing, e.g. for a connection to INRIA's CYCLADES network in 1973. But unlike these early scenarios where satellite links were operated bidirectionally, i.e. in the same way as terrestrial cables, upcoming scenarios are likely to use a large number of low-cost receive-only devices. The high-bandwidth broadcast capabilities of these systems suggest satellites as the ideal technology to serve e.g. home-networks with multicast traffic.

However, IP generally requires nodes to *exchange* data, e.g. routing information or acknowledgements. Hence, receive-only devices cannot be operated with the regular IP stack. Although one might make up a multicast protocol that abstains from sending feedback messages and simply broadcasts a session on a satellite link, most of the modern multicast protocols for routing, congestion control and reliability require feedback from the network nodes.

Normally, hosts with satellite receivers can be assumed to have an additional bidirectional link, e.g. a dial-up connection that connects them to the Internet. Hence bidirectional communication is feasible in principle, but without further action the network will exhibit *asymmetric routing* since we want to receive packets from a link to which we cannot send packets.

In practice, this means that a packet contains the receive-only interface's IP-address as source address although it is sent from the bidirectional interface. The source address field does thus not indicate the packet's source but the destination to which replies are solicited. In Figure 1a packets to the destination 132.151.1.19 emerge from the interface 134.155.48.93 while the header indicates 192.54.168.155 as the source address. As a consequence, data and the corresponding replies will not travel on the same route.

The latter is not per se a drawback. Originally, the concept of a datagram network imposed very little restrictions on the way packets were routed through the network [11]. According to the end-to-end principle [17] each router is free to decide on a packet's next hop provided that the final delivery can be guaranteed. No state is kept in the routers, and even in cases where both hosts of e.g. a TCP connection have only one interface, data and acknowledgements need not travel on the same route [1]. Asymmetric routing is thus well in line with IP.

Before further analysing the implications of asymmetric routes on modern IP protocols, we shortly mention another important scenario where asymmetric routes occur, namely Mobile IP (cf. Figure 1b). A mobile host keeps its home address while being attached to the network of its foreign agent. While traffic sent to

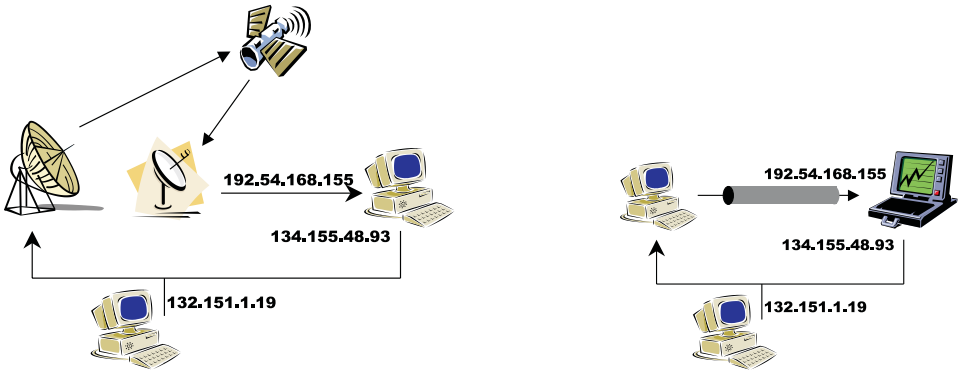

Figure 1: Two examples for the origin of asymmetric routes: a) a node with a receive-only interface b) a mobile host

the mobile host has to be tunnelled, traffic emerging from there can be routed directly to its destination. Again, packets bear another source address than the actual interface used for sending would suggest.

Implications of Asymmetric Routes

Despite these legitimate and desirable situations where packets and reply packets take different paths some protocols rely on the fact that routing paths are bidirectional:

- Ingress filtering [8] relies on the fact that the source address field of an IP packet denotes the real source and is not used as reply address only. It proposes to discard packets with presumed 'wrong' source addresses in order to restrict denial-of-service attacks. This policy is ignorant of the fact that a host might have legitimate reasons to have responses sent to a different subnet.

- Protocol-independent multicast (PIM) [3, 5, 6] builds its distribution tree for reverse path forwarding (RPF) on the metric found in the unicast routing information base (RIB). However, this only works if the assumed reverse path to the multicast sender or rendezvous-point (RP) does in fact coincide with the path originally taken by the multicast traffic. If the local unicast RIB indicates a reverse path that differs from the path originally taken by the packet in question PIM must discard the packets since they enter via the

wrong interface, i.e. via an interface this router will not use to reach the source or RP.

In the following we will examine more closely the second point. It is a genuine routing problem (the first point is merely a security problem which seems to be better addressed by authentication protocols based an cryptographical methods [12]). Note that we also do not address issues associated with very-large-scale multicast groups and the high latency of satellite links. For these problems some interesting approaches exist [2]. Additionally, new protocols based on exponentially distributed random timers have been proposed recently [14, 9].

This report is structured as follows: Section 2 analyses PIM in networks with asymmetric routing. It is explained why PIM does not work in the desired way across unidirectional links. Section 3 gives a short overview over an existing solution using tunnels and discusses its benefits and drawbacks. Section 4 presents an enhanced link-layer tunnelling that enables PIM to work more efficiently in networks that contain unidirectional links. Section 5 describes how these enhancements can be included into PIM in order to achieve a clear separation between link layer and network layer. Section 6 draws conclusions and sums up the report.

2 PIM and Asymmetric Routing

As explained in the introduction, unidirectional satellite links may well become a major source of routing asymmetry in the future Internet. We will now more deeply analyse the implications of this asymmetry on the applicability of PIM as a multicast routing protocol.

One main idea behind PIM is the use of an Internet node's unicast RIB for the *reverse path forwarding* algorithm. Shortly speaking, RPF forwards only those multicast packets that enter the router via the interface that is used to reach the packet's source. For shared trees, the same algorithm is used but a rendezvous-point (RP) takes the role of the source. Assuming symmetric routing, this will lead to a source- or RP-rooted distribution tree with two characteristic properties:

- Unnecessary packet duplication that would occur from routing loops is avoided. This is a crucial property for any multicasting algorithm.
- Each node will receive packets after a minimal number of hops (if the unicast metric is hop-based). This "shortest path" property does not hold for more general distribution trees. Especially, other schemes (e.g. Steiner trees) might optimise the total use of bandwidth rather that minimise individual delivery delays.

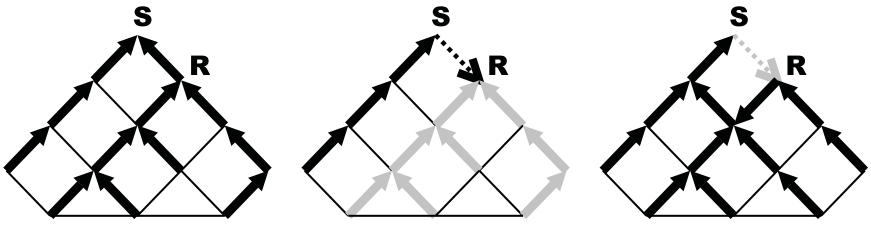

Figure 2: a) An example network with a RPF distribution tree (bold lines). Arrows indicate the unicast routing direction. b) If the bidirectional link from the source S to node R is replaced with a unidirectional link (dashed arrow) a large part of the distribution tree is cut off. c) PIM will then construct a RPF tree that does not make use of the UDL.

In networks with asymmetric links the concept of PIM to use the unicast RIB does not work any more. Consider an example network where one bidirectional link was replaced by a unidirectional link (Figure 2): Although this change of the network's topology would not affect the multicast data flow from the source S to all the interested nodes, PIM will discard all the packets that successfully arrive at the receiver R instead of forwarding them further down the tree.

The reason for this obstructive behaviour is PIM's concept of using the *unicast route towards* the multicast source or RP for the construction of RPF trees. Accordingly, PIM forwards only those packets that enter via the interface the *unicast RIB* entry points to. But since with unidirectional links packets can enter via a receive-only interface to which in principle no outbound route points, PIM cannot properly work in these settings.

On the other hand, group membership reports [7] and PIM join messages [3, 5] will similarly be propagated upstream along the *unicast route*. Multicast routers will mark the respective incoming interfaces as forwarding interfaces. As a result, PIM will automatically construct a forwarding tree that does not include the UDL. Hence, the problem described above will not occur during normal operation. Thus, even without modification PIM still works in networks with asymmetric routing, but it does not make use of (potentially high-bandwidth) unidirectional links. All traffic is forwarded along the (supposedly low-bandwidth) unicast routes.

Before we describe a working solution to the problems described above we summarise our findings: PIMs mechanism of tree construction avoids UDLs. However, forcing packets into such a link does not help either since all downstream routers would discard these packets. Both problems, tree construction and obstructive discarding of packets, emerge from the fact that PIM confuses unicast and

multicast routing: while the former is used to construct shortest paths *to* a host, the latter should find shortest paths *from* a host in order to construct valid RPF trees. In completely bidirectional settings both meanings coincide. But only one unidirectional link suffices to disturb this match in a potentially large part of the network.

3 Link-layer Tunnelling

As described above, PIM cannot make use of unidirectional links. Thus, the only possibility to use PIM without modification is to pretend that those links were bidirectional. One way of achieving that is the use of *link-layer tunnelling* [4].

Link-layer tunnelling emulates a bidirectional broadcast network even though the link is physically unidirectional. We call the node that sends packets to the unidirectional link *feed*. Packets can travel on the unidirectional link *downstream* to possibly many *receivers*. For the opposite direction all packets are tunnelled upstream to the feed. This can be done by using the underlying (bidirectional) network. Finally, the feed can send the packets downstream again. By doing this, the UDL recovers bidirectional broadcast capabilities at the cost of extensive tunnel usage.

In order not to affect the upper layers of the protocol stack, tunnelling operates at the link layer. Packets that the receiver sends to the UDL are taken out of the protocol stack at the link layer. These MAC packets are then encapsulated according to [10] and sent through the bidirectional interface towards the feed end of the UDL. There they are decapsulated again. If the packet's MAC address indicates that the packet was addressed to the feed the packet is delivered to the upper protocol layers. Otherwise the packet is sent downstream again. It is then received by the appropriate receiver node where it is processed accordingly. Link layer tunnelling is illustrated in Figure 3.

A similar approach is applied for Mobile IP. Here packets that are destined to a mobile host are encapsulated by its home agent and sent to the mobile host's care-of address [16, 15]. The mobile host's replies are either sent directly to the source of the original packet or again tunnelled back to the home agent [13]. Due to this similarity, link-layer tunnelling uses many of the techniques developed for Mobile IP.

Based on this link-layer tunnelling mechanism we can now analyse PIM again: Although routing updates are not relayed beyond the link the respective router is attached to, link-layer tunnelling creates a virtual link that is used by the unicast routing in the same way a real link would be used. Assuming that this mechanism

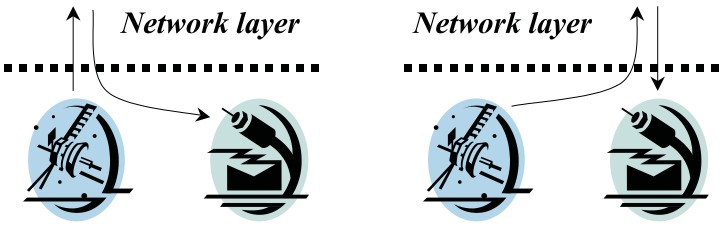

Figure 3: With classical link-layer tunnelling (left) the IP layer uses the UDL as the default link. All outbound packets have to be tunnelled through the bidirectional link. With the enhanced link-layer tunnelling proposed here (right) the BDL is the default. For the IP layer all packets seem to enter via the bidirectional interface.

produces a route towards a certain area in the network, PIM will now use the UDL for multicast traffic from this area. Since we will want to receive all multicast traffic via the UDL we have to have a default route that points to the respective interface. As a result, PIM will work without further modification.

On the other hand, this leads to inefficient routing since now all packets have be tunnelled to the feed. According to the construction of PIM, a region's multicast traffic will use the UDL if and only if unicast traffic towards this region is tunnelled to the feed. Thus, both the network load and the routing overhead at the feed are increased. In the worst case, where source and destination are neighbours, all the packets sent by a specific receiver utilise each link between this receiver and the feed twice, once in the tunnel and once outside the tunnel (cf. Figure 3).

A similar problem also occurs with bidirectional tunnelling in Mobile IP where all packets from the mobile host travel via its home agent regardless of the packets' actual destination. Whereas with Mobile IP this is a controllable drawback, here this fundamental inefficiency is critical, since it prevents this solution from being scalable. Even more, this solution is highly error-prone since it relies on the full functionality of the feed not only for multicast traffic but for all traffic.

These drawbacks are so severe that an improved mechanism has to be devised before unidirectional links can be widely employed with multicast routing. A possible improvement of the link-layer tunnelling idea is described in the following section. A more general solution will be sketched in section 5.

4 Enhanced Link-layer Tunnelling

The key concept of PIM is to send join messages to the interface on the unicast route and conversely accept multicast traffic only from this interface. Link-layer

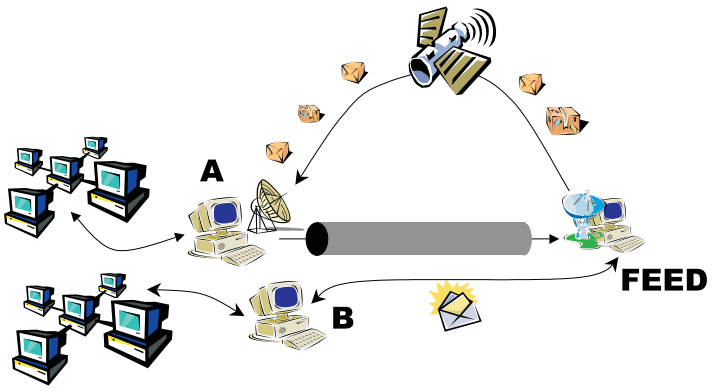

Figure 4: With link-layer tunnelling PIM requires *all* traffic to be tunnelled to the feed. For hosts A and B in neighbouring subnets this leads to considerable inefficiencies since traffic cannot be directly sent from A to B but has to be routed via the feed.

tunnelling can handle both these requirements with one general mechanism: it forces *all* the traffic to flow via the feed. Due to this trick the UDL is always on the respective unicast route towards a potential multicast source. But as explained above, this mechanism leads to a general inefficiency. If however the two objectives are addressed separately, the overall efficiency can be greatly improved:

- Firstly, an improved mechanism should not tunnel all the outbound traffic to the feed but only PIM's routing messages. This will enable PIM to build its distribution tree in the desired way while avoiding the unnecessary tunnelling overhead. However, this method will only work properly if all PIM routers in the area served by the UDL send their join messages towards the receiver end of the tunnel. This is satisfied if the respective router is the gateway for that area since then all outbound unicast routes point to that machine. This condition can normally be satisfied easily.

- Secondly, in order to have PIM accept multicast packets from the the UDL, these packets must enter the network layer via the correct interface. But unlike with classical link layer tunnelling packets received from the UDL are pretended to have entered via one of the bidirectional interfaces. Based on the address of the multicast source or the RP the link layer mechanism can retrieve the correct interface from the unicast RIB.

  With this mechanism unicast routes point as they would do without the UDL. However, PIM will accept multicast traffic for further distribution. Only the

link that was retrieved from the RIB as the correct incoming interface is excluded from the distribution. This is the bidirectional link that connects to the area from where the multicast traffic originates. Normally this would be a link to an ISP so that this exclusion is in fact the desired behaviour of the gateway. Multicast traffic from local sources could still travel across that link since it is either sent directly to the rendezvous point, or it uses a different tree that does not include the UDL and is hence not affected by this mechanism.

In practice, one could imagine to combine a satellite interface with a modem into one virtual link. With that combination only the tunnelling of PIM messages would have to implemented. All other outbound traffic would automatically use the bidirectional link while multicast traffic from the UDL is transparently presented to the network layer.

This solution requires only the router that is attached to the UDL to be modified. All other nodes can use this router as a gateway without further modification. Furthermore, PIM and other network-layer protocols do not need to be modified on any node.

Together with a feedback suppression mechanism, this solution is quite scalable and also less error-prone than the classical link-layer tunnelling described above: Even if no feed is operational unicast routing still works.

On the other hand, this combination of a unidirectional interface with a bidirectional interface into one virtual interface confuses link layer and network layer. Even though it eliminates the necessity to look up interfaces in the RIB, the tunnelling of the PIM messages still requires the inspection of packets at the link layer. Thus one should prefer to modify PIM so that it can directly handle unidirectional links. This more general solution will be presented briefly in the next section.

5 Dual RIBs

All the problems discussed in this report arise from the fact that PIM confuses routes *to* a destination-address with routes *from* that address. So a general solution for the unidirectional link problem should distinguish the two routes.

This can be achieved by implementing the method described above directly into PIM. For a UDL the RIB then contains an additional entry. It identifies the unidirectional interface as the one to accept multicast packets from. At the same time it denotes the tunnel through which PIM can send its messages to the feed.

The unicast RIB to the feed is not affected by this extra entry. So unicast packets will use the bidirectional interfaces in the usual way.

In contrast to the enhanced link layer tunnelling technique layers are now clearly separated. The network layer itself distinguishes unicast traffic that should be sent directly from PIM messages that have to be tunnelled. The link layer does not need to look into the packets any more which is a much clearer solution.

With this method PIM has to be changed only on the nodes connected to the UDL. Other nodes are not affected. This makes this method as suitable for quick deployment as the enhanced link-layer tunnelling described above.

6 Conclusions

PIM's usage of the unicast RIB is not appropriate for networks containing unidirectional links. This is due to the fact that unicast routing is concerned with shortest paths *to* a host whereas reverse path forwarding needs shortest paths *from* a host.

Link layer tunnelling [4] is an effective solution that enables PIM to operate in these networks without modifications. It is however not efficient since all packets have to be tunnelled to the UDL's feed. Even more this solution does not scale well.

We have proposed an enhanced link layer tunnelling that allows efficient unicast routing and additionally enables PIM to operate across UDLs. This is achieved by only tunnelling PIM messages to the feed. Traffic from the UDL is presented to the IP layer as if it entered the system via a BDL.

A more general solution to the problem is a modification of PIM that uses dual RIBs. They distinguish between outbound and inbound routes at the IP layer.

With the help of these modifications unidirectional links such as satellite links can provide inexpensive and widespread access to multicast services. This will help to facilitate the ubiquitous use of IP multicast.

7 Acknowledgements

The work described in this paper was to a large extend carried out during a short-term scientific mission at INRIA, Sophia-Antipolis. I would like to thank Walid Dabbous for hosting me at the RODEO group in Sophia-Antipolis. Special thanks got to Emmanuel Duros, Hitoshi Asaeda (IBM, Japan) and all the members of the RODEO group for their helpful discussions. The short-term scientific mission was financially supported by COST Action 264 of the European Commission.

References

[1] Jon C. R. Bennett, Craig Partridge, and Nicholas Shectman. Packet reordering is not pathological network behavior. *IEEE Transactions on Networking*, 7 No. 6:789–798, December 1999.

[2] Jean-Chrystome Bolot, Thierry Turletti, and Ian Wakeman. Scalable feedback control for multicast video distribution in the Internet. In *Proceedings of the ACM SIGCOMM*, pages 58–67, 1994.

[3] Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. The PIM architecture for wide-area multicast routing. In *IEEE Transactions on Networks*, pages 153–162, April 1996.

[4] Emmanuel Duros, Walid Dabbous, Hidetaka Izumiyama, Noboru Fujii, and Yongguang Zhang. *A Link Layer Tunneling Mechanism for Unidirectional Links*. Internet Engineering Task Force, June 1999. draft-ietf-udlr-lltunnel-02.txt.

[5] Deborah Estrin, Dino Farinacci, Ahmed Helmy, David Thaler, Steve Deering, Mark Handley, Van Jacobson, Ching-Gung Liu, Puneet Sharma, and Liming Wei. *Protocol independent multicast - sparse mode (PIM-SM): Protocol specification*. Internet Engineering Task Force, RFC 2362, June 1998.

[6] Deborah Estrin, Van Jacobson, Dino Farinacci, David Meyer, Liming Wei, Steve Deering, and Ahmed Helmy. *Protocol Independent Multicast - Version 2: Dense Mode Specification*. Internet Engineering Task Force, June 1999. draft-ietf-pim-v2-dm-03.txt.

[7] William C. Fenner. *Internet Group Management Protocol, Version 2*. Internet Engineering Task Force, RFC 2236, November 1997.

[8] Paul Ferguson and Daniel Senie. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. Internet Engineering Task Force, RFC 2267, January 1998.

[9] Timur Friedman and Don Towsley. Multicast session membership size estimation. *IEEE INFOCOM*, pages 965–972, March 1999.

[10] Stan Hanks, Tony Li, Dino Farinacci, and Paul Traina. *Generic Routing Encapsulation*. Internet Engineering Task Force, RFC 1701, October 1994.

[11] Christian Huitema. *Routing in the Internet*. Prentice-Hall, Inc., 1995.

[12] Stephen Kent and Randall Atkinson. *IP Authentication Header*. Internet Engineering Task Force, RFC 2402, November 1998.

[13] Gabriel E. Montenegro. *Reverse Tunneling for Mobile IP*. Internet Engineering Task Force, RFC 2344, May 1998.

[14] Jörg Nonnenmacher and Ernst W. Biersack. Scalable feedback for large groups. In *IEEE/ACM Transactions on Networking*, volume 7(3), pages 375–386, June 1999.

[15] Charles E. Perkins. *IP Encapsulation within IP*. Internet Engineering Task Force, RFC 2003, October 1996.

[16] Charles E. Perkins. *IP Mobility Support*. Internet Engineering Task Force, RFC 2002, October 1996.

[17] Jerome H. Saltzer, David P. Reed, and David D. Clark. End-to-end arguments in system design. *ACM Transactions in Computer Systems*, 2(4):277–288, November 1984.