On Routes and Multicast Trees in the Internet

Jean-Jacques PANSIOT

Dominique GRAD

Université Louis Pasteur - LSIIT URA-CNRS 1871
Computer Science Department
7, rue Descartes 67084 Strasbourg Cedex, France
{pansiot, grad}@dpt-info.u-strasbg.fr
http ://dpt-info.u-strasbg.fr/~{pansiot, grad}

***Abstract* : Multicasting has an increasing importance for network applications such as groupware or videoconferencing. Several multicast routing protocols have been defined. However they cannot be used directly in the Internet since most inter-domain routers do no implement multicasting. Thus these protocols are mainly tested either on a small scale inside a domain, or through the Mbone, whose topology is not really the same as Internet topology. The purpose of this paper is to construct a graph using actual routes of the Internet, and then to use this graph to compare some parameters - delays, scaling in term of state or traffic concentration - of multicast routing trees constructed by different algorithms - source shortest path trees and shared trees.**

***Key words* :** Routing, routes, Internet, multicast, shortest path trees, centered trees

Introduction

Multicast routing is an active research area. The problem is to transmit a data packet from one source to K receivers. A solution, at the network level, is to construct a multicast tree. A packet enters the tree at some node, and is propagated towards all leaves, corresponding to receiving nodes. Multicast routing protocols such as DVMRP [WPD 88], PIM-DM [DEF 97] or MOSPF [Moy 94], are based on a tree for each source, and are mostly limited to a small scale, although DVMRP is used throughout Internet via the Mbone. These protocols have a high cost since one tree must be maintained for each pair (source, group), and routers that are not part of the tree must also be involved (flooding). However it should be noted, that in the Mbone, tunnels may transparently get through some routers which have therefore no state information to maintain. Newer protocols, usable on a larger scale are now developed. Some are based on a unique centered tree per group, such as CBT [BFC 93], others may also include source rooted trees, such as PIM-SM [EFD 97]. In these two cases, routers not part of a tree do not incur any cost for maintaining trees. On the other hand, intermediate routers with degree 2 in the multicast tree must maintain tree state and signaling, although their role is only to forward multicast packets in much the same way as unicast packets. Solutions [GPZ 96] have been proposed to free these degree 2 nodes from any cost in maintaining multicast trees.

The goal of this paper is twofold. Firstly to get some experimental data on the shape of multicast trees one can actually obtain in Internet: node degree, route length,... These data could be used in particular to calibrate tree and graph generators used to simulate or validate network protocols. Secondly to get more directly usable information for people working on multicast tree construction. For example, are there many nodes of degree 2 ? Are trees rooted in different sources in the same graph very different ? In the following, we are interested in sparse groups, that is groups where the average distance between members is high, and with membership ranging up to a few thousands.

In Section 1, we describe how we constructed a graph from actual Internet routes. We mention some problems we found in tracing routes, and we discuss the realism of the graph we obtained. In Section 2, we analyze more precisely the structure of our graph. In Section 3, we compare different types of multicast trees such as source rooted shortest path trees (SPT) or shared trees (ST), in terms of scalability. We compare for example the average delay, and the amount of state information for ST and SPT. We also show that in our graph, traffic concentration for many groups is not significantly higher for ST than for SPT. In section 4, we introduce the notion of reduced multicast trees, and compare their characteristics with trees analyzed in the previous section. We show that reduced trees need much less state information they allow.

1. Graph construction

1.1 Collecting routes

The first phase of this study took place in the summer of 1995. It consisted in building a large base of actual routes between scattered points in the Internet. To do this, we selected IP addresses in our network accounting database. These addresses correspond to hosts having previously communicated with us, therefore it is plausible that they may constitute a multicast group. A first step was to eliminate either undeclared addresses (more precisely, addresses not declared in inverse maps, so that the corresponding domain name cannot be retrieved), or hosts not responding to packets sent using Loose Source Routing and Recording (LSRR). We have then constructed a route from our network to 5000 remaining hosts.

These hosts are distributed in 54 different top domains (see Figure 1.).

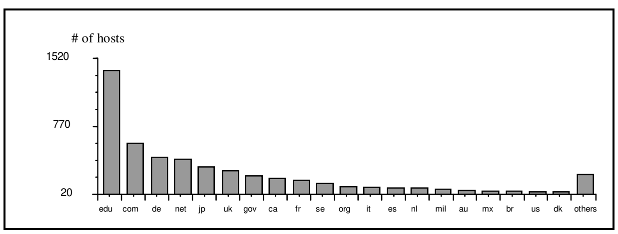


Figure 1: Domain hosts distribution

We may consider that our sample is highly geographically distributed in around 50 countries, and in different types of networks (academic, commercial,...). The average length of these 5000 routes is 20.49, where the length is the usual hop count. This is the only cost indication we can easily retrieve remotely. We have then selected 11 more hosts among the 5000, to use them as new sources of routes. These sources were chosen in the main top domains, and such that they allow to forward source routed packets [Pos81a]. We selected 3 sources in *edu*, and 1 in each of the domains *au, ca, com, de, fr, gov, jp*, *net* and *org*. These 12 sources are thus quite scattered throughout the Internet. We then traced routes from the 11 new sources. In practice some routes could not be traced, either because of transient routing problems, or, more frequently, because some intermediate routers filter LSRR packets.

1.2 From routes to graph

Globally we traced 30788 routes, with an average length of 17.63 for all sources. The source in domain *net* has the smallest average distance with 12.8, and the source in domain *au* the largest with 21.49. One may observe that even if the Internet has no real center, some nodes are, in the average, much closer to the rest of the Internet than others. Here the difference is more than 40%.

In the sequel, we will only consider destinations we could trace from the 12 sources. This produces a base of 15322 routes towards 1270 destinations. The maximal length of these routes is 35. We then consider each hop of a route as an edge in our global graph, consisting of 3888 nodes and 4857 edges.

Remark: traced routes and calculated routes.

Our goal is to construct trees similar to multicast trees with a source and K destinations. It should be noted that a set of routes traced from one source to K destinations gives a graph but not necessarily a tree for several reasons:

- because of policy routing, routes from node A to two nodes B and C may share a common node D, without being the same route from A to D,
- there may be several routes with the same cost between 2 points, and routers may use them alternatively,
- routes being computed at different times, may change temporarily or definitively.

In this three cases, cycles may appear in the graph.

1.3 The graph

Let $G = \{ N, E, D \}$ be the non oriented, connected graph obtained from our route database, where N is the set of nodes, E the set of edges and D is a cost (distance) function for edges.

We will consider that an edge obtained from a route (a hop) is always bi-directional, even if it appears only in one direction in our database. This is probably the case of most terrestrial links of Internet, even if policy routing implies non symmetrical routing. We have

$$\mathrm{card}\,(N) = 3888,$$

$$E \subset N \times N,$$

$$\forall\, n, m \in N,\ (n,m) \in E \Rightarrow (m,n) \in E$$

We assume that all edges have the same unitary cost, since we did not collect more precise information on routes. We extend this distance function to any pair of nodes (n,m) in N, by computing the shortest path in G from n to m.

In order to classify nodes of G, we define for each node n:

- Deg(n) is the degree of n that is, the number of edges incident to n
- Dmax(n) is the maximum distance, by the shortest path, between n and any other node in G.

1.4 Graph representativeness

To verify if data we collected were a sufficiently good sample of a part of Internet, we have already shown (see 1.1) that destinations are widely spread over the world. To see if our sample is sufficiently dense, we did the following: for any given source s among the 12 sources, we count how many nodes and edges are added with s. We observe that in the average (over s), only 0.7% (± 0.5%) edges and 0.4% (± 0.2%) nodes are added. Moreover, almost 90% of all nodes and edges are already there with only 6 sources, whatever these 6 sources. We may deduce that adding a new source would not change our graph significantly. Of course this does not mean that this graph contains all actual links between nodes.

1.5 Tracing routes: tools and problems

1.5.1 Traceroute

The first part of our problem was to compute Internet routes. We used the well known tool *traceroute* from Van Jacobson. This software allows to trace a route from a point A to a point B, possibly going through an intermediate point C (option *-g* : loose source routing) [Cal83]. Since the goal is to collect a large number of routes, we had to improve the time to trace one route. Depending on the length of the route and on network load, tracing a route may take in the order of a minute. Note also that standard *traceroute* makes 3 queries for each hop. This number may become insufficient in case of congestion of some link. But if we increase the query number, for example to 5, the time to trace a route is increased by the same factor (e.g. 2/3). So we added a new option to *traceroute*, called -u. In this case, for a given TTL, *traceroute* stops querying after the first successful answer (or when the maximum number of queries is reached). In many cases, one query is sufficient.

Another optimization concerns source routing. Since we have to trace a large number of routes going through a given node S (one of the 11 distant sources), it is useless to trace each time the route from our local host to S, it is sufficient to start at node S. For this we introduce a new option *-M min_ttl* that allows to start with a TTL equal to *min_ttl* (for example the distance of S) instead of 0. In this manner the time to trace a route is in the average divided by two. After the tracing of a route is completed, it is stored as a list of nodes. Routes that were not « correct » (one hop did not respond, the destination did not respond, the same node appeared at two different distances) were discarded.

1.5.2 Identifying nodes

Traceroute basically produces the list of IP addresses (and when this is possible, domain names) of routers along the route. For leaves of the graph (that is sources and destinations), we considered only nodes whose domain name was known. However for intermediate nodes, we also kept nodes known only by their IP address. In practice, over more than 10 000 different IP addresses, more than 1000 (10%) remained anonymous (failure of the inverse DNS query). A more serious problem is to determine if two identifiers (name or address) correspond to the same node or not. One may assume that if two different addresses have the same name, they correspond to the same node (via different interfaces). Unfortunately, the converse is not true, two different names (such as border2-hssi1-0.chicago.mci.net and border2-fddi-0.chicago.mci.net) may correspond to two different interfaces of the same host. Worse, for two different addresses, one cannot tell a priori if they correspond to the same host.

In theory, a solution could be to query all addresses using SNMP to discover the address of other interfaces. In practice this is not generally feasible, in particular because routers do not permit SNMP access from everywhere. We have adopted a partial solution, based on the fact that when a router sends an ICMP message [Pos81b], it generally uses as source address the address of the emitting interface, rather than the address of the interface where the original packet arrived. Therefore, we have sent an UDP packet with an unused port number (same principle as *traceroute*) to all IP addresses obtained by *traceroute*.

We then verified if the source address of the ICMP Port Unreachable message (say A) was the same as the destination address of the UDP packet (say B). If this is not the case, A and B are two addresses of the same node. Note that this is likely to occur since we trace routes using source routing. In the above example, A is the interface of the normal route to the router, whereas B is the incoming interface of a source route. With this method around 200 synonyms (different addresses of the same host) were found. Obviously this method is not perfect, and in our resulting graph, some apparently different nodes are actually the same.

Concerning edges, note that distinct edges do not necessarily correspond to distinct physical links. For example, 3 edges connecting 3 routers on the same broadcast network (e.g. fddi), correspond to the same link.

2. Graph analysis

2.1 Classification of nodes

Using some characteristics of nodes, we have established a classification that allows to structure our graph into several hierarchical levels starting from leaves (nodes of degree 1). The following classes have been identified:

L Leaf These nodes have degree 1. Therefore they cannot be an intermediate node in a route, and must have been used as a destination (sources may also be of degree 1, but anyway, they were also used as destinations). Moreover the last hop to these destinations must be the same whatever the source. These nodes are at the periphery of the graph.

C Center These nodes belong to strongly connected components of the graph (that is nodes on at least one cycle) or to paths connecting these components.

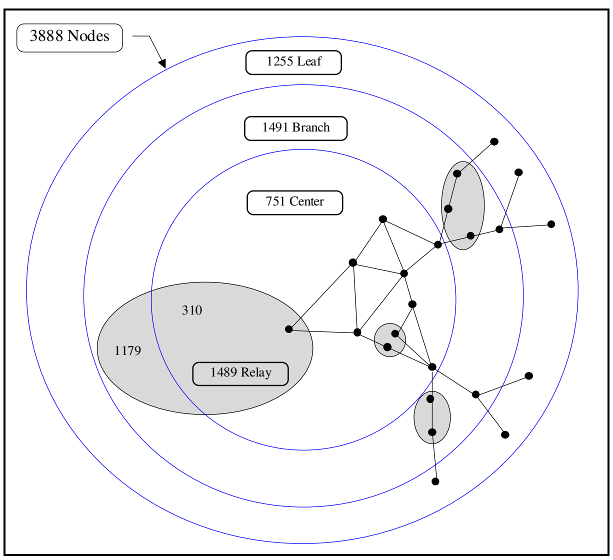


Figure 2 : Node classification

B Branch These nodes are neither Leaf nodes, nor Center nodes. For each Branch node *b*, there is at least a Leaf node *l*, such as *b* is on all paths from any Center node to *l*.

R Relay These nodes have degree 2. They are member of class Center or Branch (see shaded areas in Figure 2). Note that if we have edges (x,y) and (y,z) with y a relay node, we could remove node y and create an edge (x,z) with cost 2, as far as route length in the graph is concerned.

S Source These nodes are the twelve nodes used as sources of routes. They can be no longer distinguished in the graph, so this class must be explicitly enumerated. Since our graph is not oriented, a source is an endpoint of a route similarly to a destination

2.2 Graph statistics

2.2.1 Sources

It appears that among source nodes, only two are leaves, all others have degree at least two. This can be explained by several phenomena. A destination may be on the same LAN than a source, giving two edges from this source (one to the local destination, one to the gateway router). A second possibility is that there are two (or more) gateway routers for a source. A third possibility is obviously multi-homed sources.

An interesting result we can see in the following table is the importance of routing policy on the length of routes. For the 12 sources, we may compare the distance to a destination using either the actual Internet route (obtained by *traceroute*) or the shortest path computed from the graph, bypassing any routing policy. We see that the average (over all sources) of the longest route is 30.2 whereas the average longest path is only 21.8. If we consider the average length of all routes, it is 17.4, compare to an average distance of 13.1. Routing policy increases by 45% the maximum distance and by 32% the average distance.

			Graph		Routes	
Name	**Class**	**Deg**	**Dmax**	**avg Dist**	**Dmax**	**avg Dist**
carina.rp.csiro.au	CRS	2	25	15,6	35	21,7
iserver.istc.org	LS	1	25	16,6	30	19,1
escher.u-strasbg.fr	LS	1	23	14,3	34	21,0
enterprise.mtl.t.u-tokyo.ac.jp	CRS	2	25	15,0	28	16,8
wapiti.uwyo.edu	CRS	2	20	11,4	28	15,9
mark.ucdavis.edu	CS	4	22	13,3	32	18,3
tito.hyperlink.com	CRS	2	20	10,6	32	15,5
linus.erin.utoronto.ca	CRS	2	22	13,1	29	18,4
elmer.harvard.edu	CRS	2	22	13,2	29	19,4
eo-dns.ku-eichstaett.de	CRS	2	21	12,3	31	15,8
mon2.pressimage.net	CRS	2	19	10,9	30	13,9
thidwi.gsfc.nasa.gov	CS	4	18	10,5	24	13,0
Average		***2,2***	***21,8***	***13,1***	***30,2***	***17,4***

Table 1 : Source characteristics

2.2.2 Nodes and distances

Name	*Deg*	*Dmax*	*Class*
mae-east-plusplus.washington.mci.net	25	16	C
icm-dc-2b-s4/0-1984k.icp.net	16	16	C
sl-mae-e-f0/0.sprintlink.net	11	16	C
gsfc8.nsn.nasa.gov	14	16	BC
mae-0.uscyber.net	6	16	BC
mae-connect.interpath.net	9	16	C
mae-east-rt2.es.net	6	16	C
mae-east.digex.net	8	16	BC
vienna1.va.alter.net	14	16	C
icm-fix-e-h2/0-t3.icp.net	5	16	C
umd-rt1.es.net	5	16	C
mae-east.ans.net	9	16	C
icm-fix-e-f0.icp.net	4	16	C
nordu.net	10	16	C
sura2.nsn.nasa.gov	14	16	C
mae-east-rt1.es.net	2	16	RC
castor.himeji-tech.ac.jp	1	31	L
chaos.bio.sci.osaka-u.ac.jp	1	31	L
c58.ucs.usl.edu	1	31	L

Table 2 : Nodes with maximum and minimum Dmax values

From maximum distances computed for each node, we can deduce the diameter, radius, and center of the graph, defined as follows:

$$DiameterG = \max(\, Dmax(n)\,,\ n \in N\,) = 31$$

$$RadiusG = \min(\, Dmax(n),\ n \in N\,) = 16$$

$$CenterG = \{\, n \in N \,/\ Dmax(n) = RadiusG \,\}$$

In Table 2 we give the main characteristics of the 16 nodes belonging to CenterG. Obviously, they are all in the class Center. We give also characteristics of the 3 nodes with maximal Dmax (= 31).

2.2.2.1 Distribution of nodes by distance

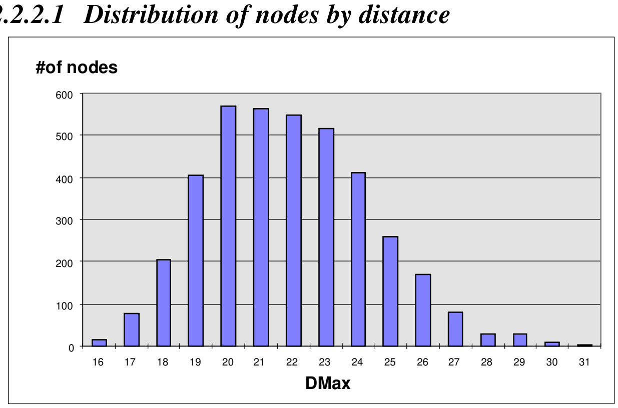


Figure 3 : Nodes and Dmax distribution

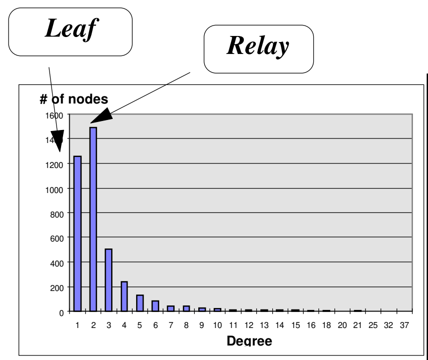
From Figure 3, we can see that the maximal distance Dmax of nodes has almost a normal distribution between RadiusG and DiameterG. We compute:

$$DmaxAvg = avg(\, Dmax(n)\,,\ n \in N\,) = 21.8$$

$$\sigma Dmax = 2.5$$

2.2.3 Degree of nodes

In Figure 4 we give the distribution of nodes by degree. It can be seen that more than 70% of nodes have degree 1 or 2 and belong to classes Leaf or Relay.



Degree	*# of nodes*
9	25
10	17
11	11
12	8
13	11
14	11
15	7
16	3
18	3
20	1
21	2
25	1
32	1
37	1

Figure 4 : Nodes and degree

From these characteristics, we compute the average and maximum degree over all nodes of G:

$$DegMax = \max(\, Deg(n)\,,\ n \in N\,) = 37$$

$$DegAvg = avg(\, Deg(n)\,,\ n \in N\,) = 2.5$$

In Table 3, we give some information on nodes with maximal degree. These nodes are in class Center. Their high degree does not necessarily correspond to a high number of physical interfaces, but sometimes to a connection to a multiple access network (broadcast networks or non broadcast networks). It should be noted that the average degree of Center nodes is 4.

Name	*Deg*	*Dmax*	*Class*
ipgw.ku-eichstaett.de	37	20	C
sl-dc-8-f0/0.sprintlink.net	32	17	C
mae-east-plusplus.washington.mci.net	25	16	C
gw.ulcc.ja.net	21	19	BC
sl-dc-6-f0/0.sprintlink.net	21	17	C
sl-fw-3-f0/0.sprintlink.net	20	18	BC
crc-cisco.u-strasbg.fr	18	22	BC
smds-gw.ulcc.ja.net	18	19	C
sl-stk-5-h1/0-t3.sprintlink.net	18	18	C

Table 3 : Maximal degree nodes

If we look back at our original data, containing all routes before destination selection (see 1.2), we find nodes with higher degrees : 45 (node connected to the German academic X25 network Win) and 37 (node connected to the English academic SMDS network Janet). These networks use IP over a switched circuit technology. All routers connected to such networks are potential direct neighbors at the IP level. Therefore there is almost no limit on the degree of a node even if the number of physical interfaces is limited. This phenomenon may become even more common with the widespread use of ATM networks in large network backbones. More generally graph edges may correspond to:

- a point to point link between two nodes
- a link within a broadcast network, such an Ethernet or Fddi LAN. Note that these LANs may be found not only on user's sites, but also within backbones for router interconnection.
- a link within a non broadcast multiple access (NBMA) network, such as X25, SMDS, Frame relay or ATM. It could be also a pure switched circuit network such as the phone network.

3. Multicast trees

3.1 Introduction

We will consider multicast groups whose members are chosen in the Leaf class of G. Multicast communication inside a group requires the construction of a multicast tree. Each member node may be a source of data packets that are forwarded along tree edges, until reaching all receivers (tree leaves). Using our graph, we constructed a « virtual » routing table, indicating for each node the next hop along the shortest path toward every other node. Note that this routing table bypasses routing policies usually used in interdomain routing. Using this table, we will construct and compare different types of trees. We distinguish two main classes of trees:

- *Shortest Path Tree (SPT).* This tree is used by only one source (the root of the tree) to reach all members. It can be considered as a directed tree, since packets flow only from root to leaves. Obviously this kind of tree minimizes delays (if we assume that the cost of an edge corresponds to a delay). The construction of such a tree is rather straightforward. In theory, using SPT, traffic should be spread over different trees, hence different links.
- *Shared Tree (ST).* Such a tree is constructed for a group and is shared by all sources. The delay between a given source and receivers is not minimal. Flows from all sources are concentrated on the same links. Obviously there are many ways to construct a ST for a group, and many heuristics have been proposed, using available information (usually shortest path to other nodes).

In Figure 5, we illustrate these two classes of trees for a group of six members. We give an example of a SPT tree with source S and five receivers, and a ST tree shared by six sources/receivers. The maximum delay is 4 for the SPT tree and 5 for the ST tree. Note that if all 6 members wish to be sources, we must construct and maintain six SPT trees.

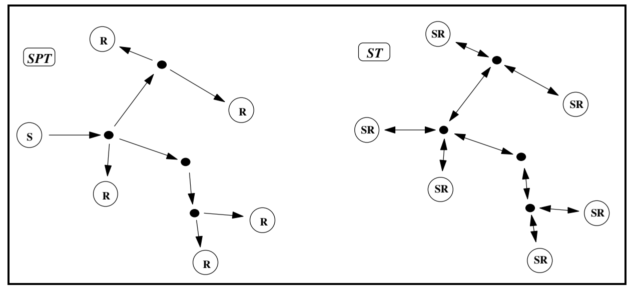


Figure 5 : Shortest Path and Shared Trees

In order to evaluate these two types of trees, and to compare several parameters (delay, traffic concentration, state information), we construct, using our graph G and randomly chosen groups, SPT trees and several kinds of ST trees. Recall that we use shortest path information in the graph. In practice, shortest path information is not always known or used in a router, in particular for inter domain routing.

3.2 Statistics on delays and weights

These statistics have been computed for groups of size M (M=10,20,50,100,200,500), and 200 randomly chosen groups for each size M. For each group we consider that each member is a potential source, and we construct M SPT trees and several ST trees using different heuristics (see SST, SCT and KMB below). To measure the cost of each approach, we compute for each kind of tree, the delay (maximum distance between a source and a receiver) and the weight (total number of edges).

Shortest Path Trees (SPT) for each source:

SPT Shortest Path Tree → For a group we construct M trees. The SPT delay is defined as the average over the M trees of the maximum delay. Similarly, the SPT weight is the average of the weight of M trees.

Shared Tree for each group:

SST *Shared Source Tree* → This is a shared tree, with the following simple heuristics: we randomly choose one member, and we use as a shared tree the

SPT tree rooted at this source. We define the delay as the maximum distance between any two leaves of the tree since we consider every member as a source. The weight is just the number of edges. Obviously the choice of the member will influence tree characteristics. We will consider the average, minimal, and maximal values of delay and weight corresponding to the choice of the SPT used as a SST.

SCT Shared Centered Tree → In this heuristics, we choose randomly a node in the Center class of G, say R, and construct the SPT tree rooted at R and spanning group members. Delay and weight are defined as for SST. Since there are many choices for R, we will consider the average, minimal and maximal of delay and weight over M random choices of R.

KMB KMB Tree → We use the KMB heuristics [KMB81] which computes an approximation of the Steiner tree minimizing the weight of a tree spanning all members. Delay and weight are defined as for SST.

Since SPT trees minimize delays, in Figure 6 we give the ratio between average delays of SST, SCT, KMB trees and SPT trees, for groups of size 10 to 500.

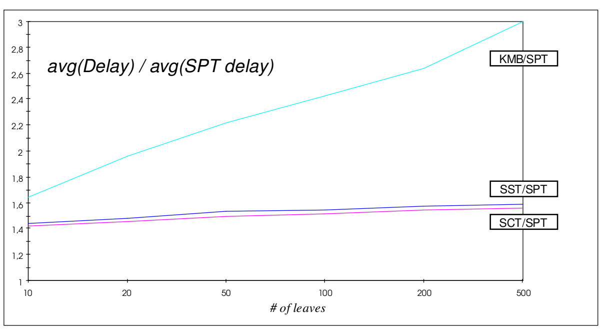


Figure 6 : Trees and average delays

We observe that the average delays for SST and SCT are almost identical and 45% greater than for SPT. Moreover for KMB which is designed for minimizing weight, delay is more than twice SPT delays for groups larger than 20, and more than three times for larger groups.

Since SST and SCT tree heuristics require a random choice either of a source or a center, we will now consider the best, average and worst choice in terms of delay for SST and SCT. Since these heuristics require only one tree for a given group, we compute the ratio between these delays and the maximum delay for the M SPT trees of the same group. We see in Figure 7 that for the average choice, SST and SCT delays are 25% greater than SPT delays, and for the best choice, delays are 10% greater than SPT's.

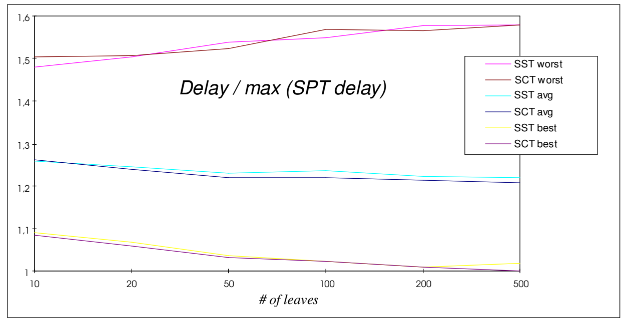


Figure 7 : SST et SCT delays

In the previous figures, we have considered, for a given tree, the delay as the maximum delay between the source and any receiver. Now we consider the average delay between a source and all receivers. We define the delay in a SCT tree as the average delay between any pair (source, receiver) of the group. Considering the average choice for the center, we see that the delay is increased by 54% compared to SPT, independently of group size. For the best center, delay is increased by 20% for large groups (500 members).

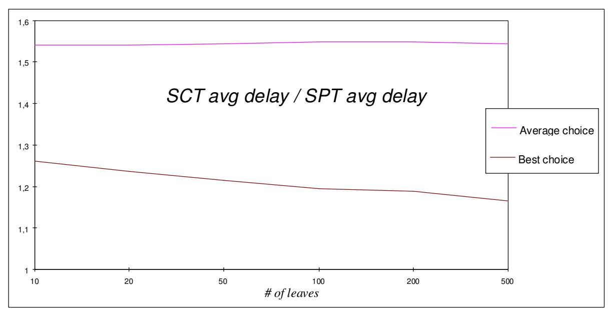


Figure 8 : SCT et SPT average delays

Weights of SST, SCT, and KMB trees converge on the average SPT tree weight when group size increases. KMB trees have, as expected the minimal weight for group size greater than 100. Note however that the benefit in weight for KMB trees seems rather small.

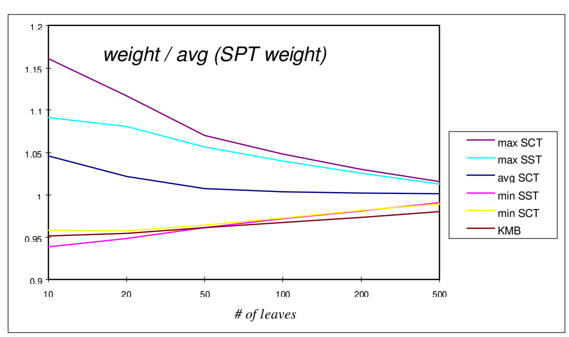


Figure 9 : Shared tree weights

3.3 Statistics on tree resources

We have shown that the average maximal delay in a SCT tree, with a center randomly chosen in the Center is around 45% higher than SPT's, and this independently of tree size. We are now going to compare resources necessary to maintain trees, using both SCT and SPT trees. Such trees require that each node maintains information on neighbors in the tree (or equivalently on interfaces used by the tree), we call this state information. A multicast tree requires also bandwidth resources on links used by one or more flow from sources to receivers.

3.3.1 State information in nodes

Globally the number of entries in routing tables (tree tables) necessary to maintain the tree is proportional to the number of nodes (or edges) in the tree, since each node must have one entry for each edge (neighbor).

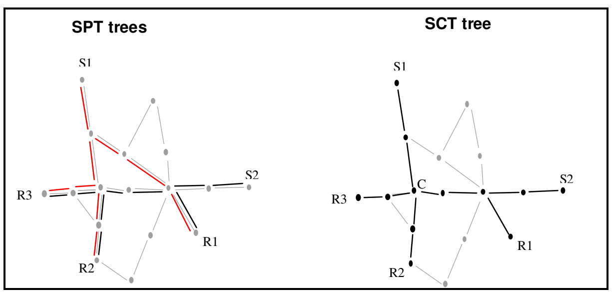


Figure 10 : SPT and SCT trees

In the example of Figure 10, we see that with a group containing two sources S1 and S2, and three receivers, R1, R2 and R3, the two SPT trees rooted at S1 and S2 require 12 intermediate nodes (6 for each tree). Some nodes are common to both trees, but in this case they must memorize some information twice. The unique SCT tree requires only 7 intermediate nodes.

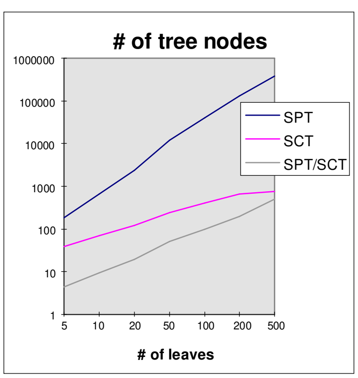


Figure 11 : SPT and SCT tree nodes

We counted the number of intermediate nodes in SCT and SPT trees for groups of size 5, 10, 20, 50, 100, 200 and 500 members. We have shown previously that weights of SCT and SPT trees are quite similar, and so is the number of nodes. Therefore the number of state information is roughly proportional to the number of trees necessary for each approach. In Figure 11, we represent, with a logarithmic scale, the number of state information necessary for SPT and SCT, together with the ratio SPT/SCT.

One may estimate that for a group of M hosts, the number of entries necessary in the SPT approach is M times greater than the number of entries for SCT. This can be seen in Figure 11, where for a group of size 100, the number of entries for SPT is 100 times bigger than for SCT. Obviously, if only a fraction (say 10%) of members are also sources, this ratio will diminish in the same proportion (e.g. 10% of M instead of M).

3.4 Statistics on traffic concentration

In order to evaluate traffic concentration in both SPT and SCT approaches, we have simulated the effect of 5000 multicast flows, consisting in either 500 groups of 10 members, or 100 groups of 50 members, or 50 groups of 100 members, or 25 groups of 200 members, or 10 groups of 500 members. Then we considered 50 000 flows (5000 groups of 10 members, 1000 groups of 50 members, 500 groups of 100 members). For each group, we considered each member as a source, hence constructed as many SPT trees, and counted how many trees share the same edge.

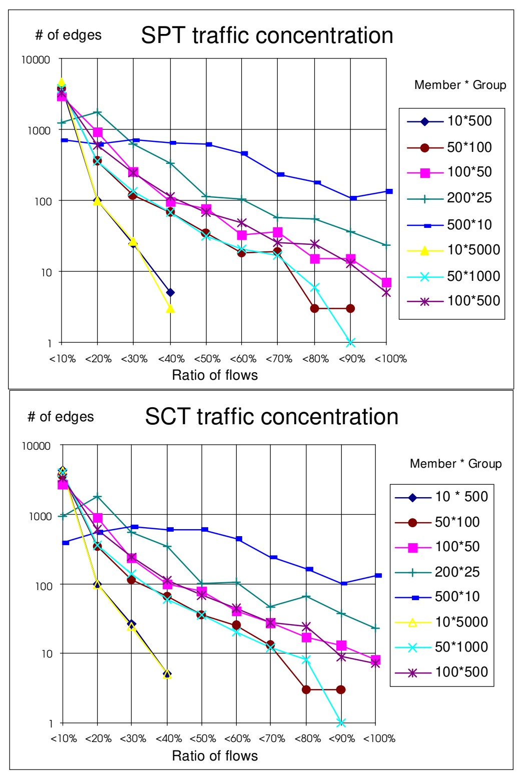


Figure 12 : Traffic concentration

For the SCT approach and each group, we chose randomly a center in class Center, constructed the SCT tree, and for each edge we counted as many flows as there are sources (since the flow of each source must go through this edge, in one direction or another). It should be noted that in this first simulation, we considered edges as non oriented, and did not distinguish the direction of flows. For each edge, we have computed the percentage of flows using it.

In Figure 12 we represent the distribution of edges for SPT and SCT (that is, how many edges support less than 10% of flows, between 10% and 20%, and so on). It is evident from the figure, that in our graph, traffic concentration is very similar for SPT and SCT, for 5000 and 500000 flows, contrary to what might be expected. Also, in both cases, concentration is higher (for a given number of flows) with large groups.

We did a second simulation with 500 flows, using a perhaps more realistic hypothesis, namely that only 10% of members of a group (randomly chosen) are actual flow sources. We use 500 groups of 10 members (1 source), 100 groups of 50 members (5 sources), 50 groups of 100 members (10 sources), 25 groups of 200 members (20 sources) and 10 groups of 500 members (50 sources).

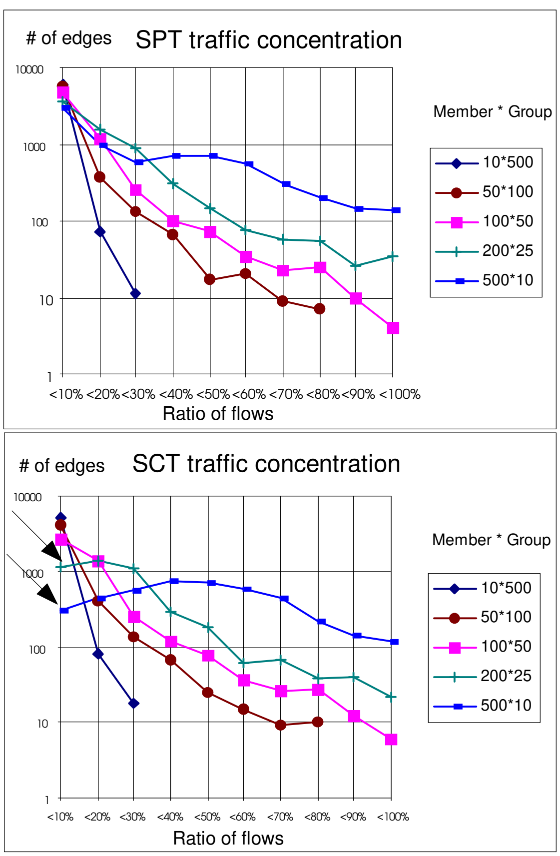


Figure 13 : Traffic concentration with 10% sources

Moreover we consider now that edges are oriented. Figure 13 represents the distribution of oriented edges for SPT and SCT. We notice a small difference in node distribution. In the case of large groups (200 and 500 members), there is less concentration for SPT than SCT, the ratio of edges with a small proportion of flows (10 % to 30%) being larger than with SCT. In any case these two approaches give very similar results in term of traffic concentration, provided that the root of the SCT tree is chosen randomly (hence is a priori different for each tree). We conclude that concentration should not be an important criterion to choose between SCT and SPT.

4. Reduced trees

4.1 Definitions

In the framework of our Logical Addressing and Routing (LAR) architecture [PGZ95], we have presented an optimization of SPT or SCT trees we call **reduced trees**. Let $T = (V, E)$ be a tree with V the set of vertices and $E \subseteq V \times V$ the set of edges. The set V can be partitioned into 3 disjoint subsets M, D and R, where M is the set of members (of degree 1), D the set of duplicating nodes (of degree at least 3), and R the set of relay nodes (of degree 2). A reduced tree has no relay nodes. Given an arbitrary tree $T = (V, E)$, with $V = R \cup D \cup M$, we can construct the corresponding reduced tree $T_r = (V_r, E_r)$ where $V_r = D \cup M$, and $(v, w) \in E_r$ if and only if there exists a path $v, u_1, u_2, \ldots u_n, w$ in T (n possibly null) with $u_i \in R$. In other words, an edge of E_r corresponds to a path in T whose intermediate nodes are relay nodes. We call an edge of T_r a **logical edge**. In terms of routes in a multicast tree, a logical edge can be seen as a unicast route without intermediate duplicating nodes.

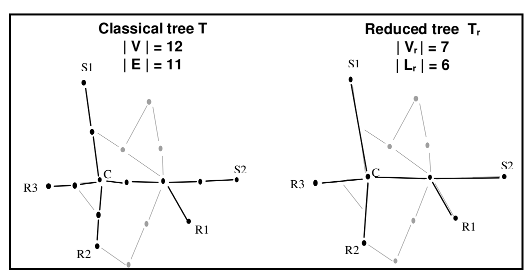


Figure 14 : SCT reduced tree

In Figure 14, we notice that for a group with 2 sources S1 and S2, and 3 receivers R1, R2 and R3, the SCT tree with root C requires 7 intermediate nodes for 5 members, where the corresponding reduced tree requires only 2 (duplicating) intermediate nodes for the same members.

4.2 Statistics on reduced trees

We counted the number of relay nodes in SCT and SPT trees for 200 groups of M members (M = 5, 10, 50, 100, 200, 500). Figure 15 gives the average ratio of relay nodes among intermediate nodes (not counting leaves), for SPT and SCT trees.

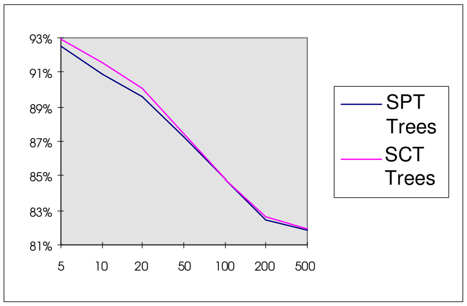


Figure 15 : Ratio relay /intermediate nodes

Whatever the size of the group, relay nodes constitute more than 80% of all intermediate nodes of SPT or SCT trees (in our graph). A relay node in a multicast tree must forward through only one interface a packet received by one interface, in much the same way as a unicast packet. So it would be interesting if these nodes were not an explicit part of the multicast tree, saving a lot of state information (in the order of 80%), as suggested in the LAR approach.

5. Conclusion

In this paper, we have shown some characteristics of a graph constructed from more than 15000 actual Internet routes. This graph appears to be a realistic model of a (sparse) part of Internet, at the time of its computation (summer 1995). With this graph we have constructed several types of trees and compared their characteristics, such as delay, weight, state information and traffic concentration. We have seen that these trees, centered or not, are quite similar in terms of delay and weight, because the core of Internet is meshed and represents only one third of nodes. We have shown that for applications that are not strongly delay sensitive, centered trees are a better solution, since they require much less state information (not to mention signaling, depending on the multicast routing protocol used), for an average delay increased by about 25%. Rather surprisingly, they do not seem to cause much traffic concentration, probably because different centered trees tend to use different links. We have also shown that reduced multicast trees require about 20% of the information needed by normal multicast trees.

Finally it should be noted that the Internet topology is constantly evolving, and it would be interesting to see which characteristics change and which (if any) are stable.

Bibliography

[BFC 93] **A. Ballardie, P. Francis, J. Crowcroft,** *Core Based Trees (CBT) An Architecture for scalable Inter-domain Multicast Routing,* SIGCOMM'93, San Francisco, USA, Sep. 93, pp. 85-95.

[Cal 83] **R. Callon,** *Internet protocol*, Proceedings of the IEEE, 71(12), Dec. 1983.

[DEF 97] **S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, L. Wei,** *Protocol Independent Multicast Version 2, Dense Mode Specification,* Work in progress, Internet Draft, draft-ietf-idmr-pim-dm-05, May. 97, 12 p.

[EFD 97] **D. Estrin, D. Farinacci, S. Deering and al.,** *Protocol Independent Multicast-Sparse mode (PIM-SM) : Protocol Specification,* Work in progress, Internet Draft, draft-ietf-idmr-PIM-SM-specv2-10, Sep. 97, 52 p.

[GPZ 96] **D. Grad, JJ. Pansiot, S. Marc-Zwecker,** *Distributed Computation of Reduced Multicast Trees,* Proceedings of TDP'96, La Londe les Maures, Jun.96, pp. 91-107.

[KMB 81] **L. Kou, G. Markowsky, L. Berman,** *A fast algorithm for Steiner trees*, Acta Informatica 15, pp. 141-145, 1981.

[Moc 87] **P. Mockapetris,** *Domain Names - Concepts and facilities*, Network Information Center, Request for Comments, RFC 1034, Nov. 87, 55 P.

[Moy 94] **J. Moy,** *Multicast routing extensions for OSPF*, CACM, Vol. 37, Aug. 94, pp. 61-66.

[PGZ 95] **J-J. Pansiot, D. Grad, S. Marc-Zwecker,** *Towards a Logical Addressing and Routing Sublayer for Internet Multicasting*, Proceedings of PROMS'95 Salzburg, Austria, Oct. 9-12 95, pp.521-535.

[Pos 81a] **J. Postel,** *Internet Protocol,* Network Information Center, Request for Comments, RFC0791, Jan. 91, 45 p.

[Pos 81b] **J. Postel,** *Internet Control Message Protocol,* Network Information Center, Request for Comments, RFC0792, Jan. 91, 21 p.

[Wax 88] **B.M. Waxman,** *Routing on Multipoint Connections,* IEEE Journal of Selected Areas in Communications, Vol. 6, N° 9, Dec. 88.

[WPD 88] **D . Waitzman, C. Partridge, S. Deering,** *Distance Vector Multicast Routing Protocol,* Network Information Center, Request for Comments, RFC 1075, Nov. 88.