D. Ooms
W. Livens
Alcatel

INTERNET DRAFT
<draft-ooms-cl-multicast-01.txt>

October, 1999
Expires April, 2000

Connectionless Multicast

Status of this Memo

Abstract

This draft proposes a new mechanism for multipoint-to-multipoint (mp2mp) communication in IP networks, namely Connectionless Multicast (CLM). Instead of a group address, a list of member addresses is encoded in the data packets.

The traditional Host Group Model [DEER] for IP multicast requires a globally unique address for each session. To support this model, multicast routing protocols create state per group in the routers. Like any connection oriented protocol, it suffers from scalability problems in backbone networks where the number of groups to be maintained can be huge. CLM does not have this problem, and additionally has some other advantages. Its limitation lies in the number of members per multicast session, not in the number of sessions. CLM will not replace the traditional multicast model. CLM

offers an alternative for mp2mp communication in the cases that traditional multicast becomes problematic.

The pros and cons of CLM are described and suggestions are made to alleviate the disadvantages. Furthermore different modes of operation are described: an end-to-end mode in close conjunction with SIP (Session Initiation Protocol [HAND]), as well as an interworking mode with PIM-SM and Simple Multicast. Both IPv4 and IPv6 are considered.

Table of Contents

Table of Abbreviations

CLM	ConnectionLess Multicast
HALE	Hierarchical Address List Encoding
IRC	Internet Relay Chat
mp2mp	multipoint-to-multipoint
PIM-SM	Protocol Independent Multicast Sparse Mode
RUN	Receiver Update Notification
SM	Simple Multicast

1. Introduction

The main goal of multicast is to avoid duplicate information flowing over the same link. By using traditional multicast instead of unicast, bandwidth consumption decreases while the state and signalling per session increases. Apart from these two dimensions, we identify a third one: the header processing per packet. This three dimensional space is depicted in Figure 1.

```
          state&signaling
            per session
             in router
                 ^
                 |
                 |
                ....
               B |  ....
               . |      ....
              .  |          ....
             .   |              ....
            .    +------------------..---> processing
           .    /               .... C     per packet
          .   /            .....           in router
         .  /         .....
        . /      .....
       ./   .....
      /A....
    /
  /
 link bandwidth
```

Figure 1

A first method to deliver identical information from a source to n destinations is to unicast the information n times (A in Figure 1). A second method, the traditional multicast model (B in Figure 1) sends the information only once to a multicast address. In the third alternative (CLM), which is the subject of this draft, the information is sent only once, but the packet contains a list of destinations (point C). For a detailed description of CLM we refer to section 3.

The three points A, B and C define a plane (indicated with dots in Figure 1): a plane of conservation of misery. All three approaches have disadvantages: the link bandwidth is a scarce resource in the access network, while state&signaling/session and processing/packet encounter limitations in the core of the network.

Traditional multicast can move a little bit along the line A-B by

switching between source and shared trees. However, this flexibility is very limited. Thus state&signaling/session is, amongst others (see section 2) a problem for the traditional multicast model in the backbone.

Also pure CLM encounters limitations in the core of the network (processing/packet), but the nice and differentiating feature of CLM is that it gives the router the choice to make its own tradeoffs. CLM has three mechanisms built in (caching, premature cloning and optimized address list encoding) that allows the router to move in this plane of conservation of misery, according to its own needs, which could be e.g. its location in the network. Caching allows a router to move from C to B, while premature cloning allows a shift from C to A.

It is often argumented that it is not worthwhile to use multicast for mp2mp communication with a limited number of members, and use unicast instead. This is definitely untrue as following example illustrates: assume n residential users set up a video conference. For e.g. xDSL, GPRS or cable modem access technologies a host has no problem receiving n-1 basic 100kb/s video flows, but the host is not able to send its own video data n-1 times at this rate. Because of the limited and often asymmetric access capacity, some type of multicast is mandatory.

In CLM, a host sends a packet with the addresses of the other n-1 members. The first router beyond the access link can probably process the packets in pure CLM mode since the link speed is relatively small here. Further in the network, where link speeds increase, routers can decide to maintain a cache entry for this session. Once arrived in a backbone network, where bandwidth is plentiful, the CLM packets could be prematurely cloned into multiple unicast packets to avoid the heavy header processing in the core routers.

A simple but important application of CLM lies in bridging the local loop for mp2mp communication. The host sends the CLM packet with the list of unicast addresses and the first router prematurely clones the CLM packet to multiple normal unicast packets.

We believe that the flexibility of CLM to adapt to local network conditions makes CLM an excellent multicast forwarding mechanism in the heterogeneous internetworks, as we currently know them.

2. The critique on the traditional multicast model

The characteristics of the traditional IP multicast model are

determined by its two components: the Host Group model [DEER] and a Multicast Routing Protocol. Both components add to the difference in nature between unicast and multicast.

In the Host Group model, a group of hosts is identified by a multicast group address, which is used both for subscriptions and forwarding. This model has two main disadvantages:

- Multicast address allocation: The creator of a multicast group must allocate a multicast address which is unique in its scope (scope will often be global). This issue is being addressed by the Malloc working group, which is proposing a set of Multicast Address Allocation Servers (MAAS) and three protocols (MASC, AAP, MADCAP).

- Destination unawareness: When a multicast packet arrives in a router, the router can determine the next hops for the packet, but knows nothing about the ultimate destinations of the packet, nor about how many times the packet will be duplicated later on in the network. This complicates the security, accounting and policy functions.

In addition to the Host Group model, a routing algorithm is required to maintain the member state and the delivery tree. This can be done using a (truncated) broadcast algorithm or a multicast algorithm [DEER]. Since the former consumes too much bandwidth by unnecessarily forwarding packets to some routers, only the multicast algorithms are considered. These multicast routing protocols have the following drawbacks:

- Connection state scalability: The multicast routing protocols exchange messages that create state for each multicast group in all the routers that are part of the point-to-multipoint tree. This can be viewed as a kind of signaling that creates multicast connection state, yielding huge multicast forwarding tables that have to be maintained in the backbone. The name Connectionless Multicast refers to the elimination of this connection state.

- Source advertisement mechanism scalability: Multicast routing protocols provide a mechanism by which members get 'connected' to the sources for a certain group without knowing the sources themselves. In sparse-mode protocols (CBT, PIM-SM), this is achieved by having a core node, which needs to be advertised in the complete domain. On the other hand, in dense-mode protocols (PIM-DM, DVMRP) this is achieved by a "flood and prune" mechanism. Both approaches raise additional scalability issues.

Multicast address allocation, destination unawareness and scalability issues are delaying widescale deployment of multicast, leading to a

search for other multicast schemes. Recently, several changes to the Host Group Model were proposed to address these drawbacks:

- Simple Multicast [PERL] uses the couple of core and multicast addresses as the group identifier. In this way core advertisement and multicast address allocation can be avoided. Furthermore, state can be avoided in parts of the network by creating tunnels. Note however that these tunnels are point-to-point and if e.g. a link has n tunnels for a certain group it will still carry n times the same data.

- In Express [HOLB], a multicast channel is identified by source and multicast addresses. Thus address allocation is not required and since there is no core, there is no core advertisement either. Creating a multicast session with multiple changing sources is implemented by using a session manager and a set of multicast channels. Note that each on-tree node still keeps state for each source of the multicast session.

3. Description

In the Host Group Model the packet carries a multicast address as a logical identifier of all group members. In Connectionless Multicast (CLM) the packet carries all the IP addresses of the multicast session members (in the context of CLM the term 'multicast session' will be used instead of 'multicast group' to avoid the strong association of multicast group with multicast address in traditional IP multicast).

In each router the next hop for each destination is determined and for every distinct next hop a new header is constructed. This header only contains the destinations for which that next hop is on the shortest path to these destinations. When there is only one destination left the CLM packet turns into a normal unicast packet, which can be unicasted along the remainder of the route.

Although not restricted to this type of multicast session, the most beneficial application of CLM is for sessions with a limited number of members (super-sparse sessions).

For sessions with a large number of members (broadcast applications), CLM can be used as an interdomain multicast routing mechanism between a limited number of domains which run either PIM-SM or Simple Multicast.

So, CLM will not replace the traditional multicast model, but offers an alternative for mp2mp communication where traditional multicast

becomes problematic.

In practice, traditional multicast routing protocols impose limitations on the number of groups and the size of the network in which they are deployed. For CLM these limitations do not exist.

CLM has the following advantages:

1) No multicast address allocation required.

2) Routers do not have to maintain state per session (or per source-session).

3) Easy transition phase. The destination address field of the IPv4 header will carry one of the addresses of the list of receivers. This enables a gradual introduction of CLM in the network. If a router does not understand CLM, it forwards the CLM packet as a normal unicast packet. The downstream CLM-upgraded routers will then take care of the branching of the tree. The minimal condition to make CLM work in the current Internet is that the designated router (= first router connected to host) of all the destinations participating in a CLM session are CLM-upgraded. When at least these routers are CLM-upgraded correct packet delivery is assured.

4) No symmetrical paths required. Traditional multicast routing protocols assume (for optimal functioning) symmetrical paths, i.e. the shortest path from A to B is the same as the shortest path from B to A. This is a false assumption in the current Internet and it is expected that this statement will become even more false when traffic engineering and more policy routing is introduced in the Internet.

5) Automatic reaction to unicast reroutes. CLM will react immediately to unicast route changes. In traditional multicast routing protocols a communication between the unicast and the multicast routing protocol needs to be established. In many implementations this is on a polling basis, yielding a slower reaction to e.g. link failures.

6) Easy security and accounting. In contrast with the Host Group Model, in CLM all the sources know the members of the multicast session, which gives the sources the means to e.g. reject certain members or count the traffic going to certain members quite easily. Not only a source, but also a border router is able to determine how many times a packet will be duplicated in its domain. It becomes also more easy to restrict the number of senders or the bandwidth per sender.

7) Heterogeneous receivers. Besides the list of destinations, the

packet can also contain a list of DiffServ bytes. While traditional multicast protocols have to create separate groups for each service class, CLM incorporates the possibility of having receivers with different service requirements within one multicast session.

8) Policy routing via BGP. No need for a specific multicast policy routing protocol (extension).

9) Unicast traffic engineering can be applied, no need for multicast traffic engineering.

However, CLM has a number of disadvantages as well:

1) Overhead. Each packet contains all remaining destinations, but the total amount of data is still much less than for the unicast (payload is only sent once). A method to compress the list of destination addresses might be useful (section 7).

2) More complex header processing. Each destination in the packet needs a routing table lookup. So a CLM packet with n destinations requires the same number of routing table lookups as n unicast headers. Additionally, a different header has to be constructed per next hop. Remark however that:

a) Since CLM will typically be used for super-sparse sessions, there will be a limited number of branching points, compared to non-branching points. By using a simple caching mechanism (see section 6) in these non-branching points, the classical forwarding method can be used and therefore the same performance achieved.

b) Among the non-branching points, a lot of them will contain only one destination. In these cases normal unicast forwarding is applied.

c) When the packet enters a region of the network where link bandwidth is not an issue anymore, the packet can be prematurely cloned. This avoids more complex processing downstream (section 5).

d) The forwarding in the branching points can also be enhanced by a caching mechanism (section 6).

e) By using a hierarchical encoding of the list of destinations in combination with the aggregation in the forwarding tables the forwarding can be accelerated (section 7).

3) Multi-access networks. Since e.g. Ethernet does not support CLM, this multi-access network will carry duplicate packets, one for each next hop.

4. Working modes

The preceding sections mainly focused on the data plane, this section will focus on the control plane of CLM. CLM is mainly applicable to super-sparse multicast sessions. Besides this, we also describe how CLM can be used as an interdomain multicast routing protocol, connecting e.g. Simple Multicast or PIM-SM domains.

4.1 CLM as an end-to-end multicast routing solution

If CLM is used end-to-end, it is especially useful for super-sparse sessions, e.g. video conferences.

In the current Internet three approaches to establish multipoint-to-multipoint sessions can be identified:

- Internet Relay Chat (IRC) [OIKA]. In this approach a set of servers keeps track of the created channels and the members of these channels. The servers are also responsible for emulating the mp2mp data delivery, which relies on a unicast forwarding.

- Current IP Multicast Group Establishment (IGMP, Multicast Protocol, Multicast Address Allocation). In this approach the creation of the group, mainly the multicast address allocation, is the responsibility of a set of servers, while the member state is distributed over the network. The data delivery is distributed over the network and relies on a multicast forwarding.

- Session Initiation Protocol (SIP) [HAND]. A host takes the initiative to set up a session. With the assistance of a SIP server a session is created. The session state is kept in the hosts. Data delivery can be achieved by several mechanisms: meshed unicast, bridged or multicast. Note that for the establishment of multicast delivery, a multicast protocol and communication with Multicast Address Allocation Servers (MAAS) are still required.

Both CLM and SIP address super-sparse mp2mp sessions. It turns out that CLM (a very flexible data plane mechanism) can be easily integrated with SIP (a very flexible control plane protocol). When an application decides to use CLM forwarding it does not affect its interface to the SIP agent: it can use the same SIP messages as it would use when it opted for meshed unicast forwarding.

4.2. CLM as an interdomain multicast routing protocol

4.2.1. Simple Multicast

Simple Multicast provides a tunnel mechanism. This tunnel is used either to cross non Simple Multicast domains or to limit the multicast forwarding state in parts of the network. When the tunnel is used for the latter purpose, the bandwidth saving of multicast is lost in these parts of the network, i.e. parallel tunnels for the same group can exist on a specific link (Figure 2). Assume ERi are Simple Multicast Exit Routers and BRi are Backbone Routers in which one wants to avoid multicast forwarding state. S is a sender to the group and Ri are receivers. For this topology Simple Multicast will construct a first tunnel from ER1 to ER2 and a second tunnel from ER1 to ER3, yielding duplicate traffic on the link ER1-BR1.

```
                 +--------BR2--------ER2------R1
                 |
                 |
S------ER1------BR1-------BR3--------ER3------R2
```

Figure 2

If the backbone routers are CLM routers the duplicate traffic can be avoided without building multicast state. The interworking is easy since ER1 knows the endpoints of both tunnels: it can send the multicast packet in CLM mode by putting ER2 and ER3 as destinations in the packet, thereby creating a point-to-multipoint tunnel.

4.2.2. PIM-SM

With MSDP [FARI] the Rendezvous Points (RPs) of different PIM-SM domains discover which RPs have sources for a certain group in their domain. Similarly, a Multicast Receiver Distribution Protocol (MRDP) can be created. MRDP allows RPs to discover which RPs have receivers for a certain group in their domain. If MRDP is running and an RP has a source for a group, it can send the data in a CLM mode to the RPs which have receivers for this group.

4.3. Summary

The working modes of CLM and other mp2mp communication mechanisms are summarized in Table 1.

	control plane		data plane
	session creation	member state	forwarding
IRC	servers	servers	servers (UC)
multicast	servers	network	network (MC)
SIP/CLM	host/server	hosts	network (CLM)
SM/CLM	/	exit router	network (CLM)
MRDP/CLM	/	RP	network (CLM)

Table 1

5. Premature Cloning

A router can decide to prematurely clone a CLM packet, i.e. duplicate a CLM packet before its actual branching point. This early cloning facilitates a less complex forwarding in all downstream routers.

An operator's border router can e.g. prematurely clone the CLM packets that would normally branch in the operator's domain. Or an edge router can prematurely clone CLM packets to multiple unicast packets (when CLM was only used to bridge the local loop).

6. Caching

The increased forwarding complexity is the main problem of CLM: packet headers have to be processed at wirespeed. It was already mentioned that in major parts of the tree the normal unicast forwarding can be applied (from the moment there is only one destination address left in the packet).

A method for further enhancing the forwarding is building a cache for the highest bandwidth sources. If the source supports caching it will put a non-zero number in the cache field of the IP option or in the CLM header: the Receiver Update Notification (RUN). Each time the set of destinations changes the source will indicate this to the downstream routers by changing the RUN. The cache will contain entries for (S, RUN). If the router receives a high bandwidth flow with a new (S, RUN) it will create a new cache entry. Unused cache entries will time out.

A possible optimization is that the source increments the RUN value by 1 if the list of receivers changes. In that way the router can immediately remove the entry (S, RUN) and replace it by an (S, RUN+1) entry. When the RUN value reaches its maximum value the clearing of

the cache entry still relies on a the expiration of a timer.

Note that a source can send to multiple sessions. In this case the source has to partition its RUN space amongst these sessions.

Maintaining a cache may seem contradictory to the main characteristic of CLM (avoid state in the router), but it is important to note that each router can independently decide to create a cache or not. The router itself can make the tradeoff between memory and processing time.

There are two ways of keeping a cache:

1) Each (S, RUN) entry has a list of next hops with the associated outgoing IP option or CLM header (or list of destinations).

2) Only (S, RUN) which have one next-hop (still multiple destinations) are cached. These packets do not have to change their IP option or CLM header, so the latter information should not be kept in the cache.

Both ways of applying a cache can be used in parallel.

7. Address list encoding

In this section the Hierarchical Address List Encoding method (HALE) is described which possibly allows to reduce the packet overhead or to increase the forwarding capacity. Given a list of IP addresses, the method iteratively separates the common prefixes of the list of IP addresses.

To illustrate HALE, consider the multicast tree in Figure 3. The source S1 wants to deliver the same information to destinations D1 (ABCD), D2 (ABCE) and D3 (AFGH). For simplicity we will assume in this example that the separation of common bits is only executed on byte boundaries. The addresses of D1 and D2 have ABC in common, so it can be written as the compound address ABC{D,E}. The address of D3 and the compound address ABC{D,E} have A in common, yielding a new compound address A{BC{D,E},FGH}.

```
                                            D1
                                           /
                                          /ABCD
        A{BC{D,E},FGH}        ABC{D,E}   /  ABCE
    S1-------------------R1------------R2----------D2
                          \
                           \AFGH
                            \      AFGH
                            R3---------------D3


    ABCD |
          >  ABC{D,E} |
    ABCE |            |
                       > A{BC{D,E},FGH}
    AFGH              |
```

Figure 3

HALE, combined with the aggregation in the routing tables, allows to save on lookup cycles.

More details on HALE and how the forwarding benefits from this encoding can be found in Appendix A.

8. IP Protocol extensions

8.1. IPv4

In IPv4 two approaches can be followed to include the list of destination addresses. Either one adds a new IPv4 option or one adds an extra header between the network and the transport layer.

Since the IPv4 option field is limited to 40 bytes (of which 4 bytes are used for the option type field) only 10 (9 in the option and 1 in the destination field of the IP header) destination addresses can be included. If the number of receivers is larger than 10 multiple packets can be sent. Although for a different purpose, [AGUI] and [GRAF] already proposed an IPv4 option to carry multiple destinations. We propose the new IP option depicted in Figure 4.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 0 0|    TYPE  |     LENGTH     |D| ENC |          RUN          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              (Encoded) List of DS Bytes and Addresses         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4

TYPE = number for this IPv4 option

LENGTH = total length of the option in bytes

D = if this bit is set the packet will contain a (encoded) DS-byte for each destination.

ENC (3 bits) = encoding method used on the list of DS bytes and destination addresses. Following encoding methods are defined:

0 = no encoding
1 = hierarchical address list encoding (see section 7)

RUN (12 bits) = Receiver Update Notification (see section 6). If the sender does not support the caching mechanism, it has to set RUN to zero.

Figure 5. shows the 'List of DS Bytes and Addresses' in case the D-bit is set and ENC = 0. Remember that the first destination and DS byte are placed in the normal IP header.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     destination address 2                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              . . .                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     destination address N                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  DS byte 2     |  DS byte 3     |      ...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                |  DS byte N     |           padding           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5

The 'List of DS Bytes and Addresses' encoding in case the D-bit is

cleared and ENC = 1 is explained in Appendix A.3.

Instead of using the IP option field one could add an extra header between the network and the transport layer. This header is depicted in Figure 6. The IP header will carry the protocol number PROTO_CLM.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|VERSION|UNUSED |    PROT ID    |            LENGTH             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          CHECKSUM             |D| ENC |          RUN          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              (Encoded) List of DS Bytes and Addresses         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6

VERSION = CLM version number

PROT ID = specifies the protocol on the transport layer

LENGTH = length in bytes of the CLM header.

CHECKSUM = it is not clear yet whether a checksum is needed (ffs). If only one bit is wrong it can still be useful to forward the packet to N-1 correct destinations and 1 incorrect destination. On the other hand when the header info is used to install a cache (see section 6) one can not allow that packets are permanently forwarded wrongly. One approach could be to provide a checksum per destination.

The other fields were described earlier.

8.2. IPv6

Since IPv6 allows more flexibility in adding options (e.g. no limitation on the size), there is no limitation in terms of encoding on the number of destinations that can be put in a packet. However, since a packet can only be sent after all the destinations have been processed, packets with a lot of addresses will experience a lot of delay and can delay other packets as well if no preemptive scheduler is used.

For IPv6 the overhead will be larger since the addresses are longer. Note however that IPv6 probably allows an encoding with better compression because of the geographical/provider distribution of

addresses. Moreover if CLM is used as an interdomain multicast mechanism only the locator part of the address needs to be considered.

9. Security Considerations

Since a packet contains every destination, it is much more easy to restrict multicast sessions to certain receivers. It also allows ISPs to check, when a packet enters their network, how much resources this packet packet will consume in their network.

In the end-to-end mode it is also easy to restrict the number of senders or to restrict/monitor the bandwidth of a sender.

10. Acknowledgements

The authors would like to thank Emmanuel Desmet, Hans De Neve and Miroslav Vrana for the fruitful discussions and/or their thorough revision of this document.

A Hierarchical Address List Encoding (HALE)

A.1. Compound addresses

A list of addresses can be compressed into a compound address by writing common prefixes only once, followed by a list of their suffixes.

We used following syntax to represent compound addresses:

```
<compound_address> ::= prefix "{" <suffix_list> "}"
<prefix>           ::= [<address_fragment>]
<suffix_list>      ::= [<suffix_list> ","] <suffix>
<suffix>           ::= <compound_address> | <address_fragment>
<address_fragment> ::= [<address_fragment>] <symbol>
<symbol>           ::= [A-Z] in our example, one bit in practice
                       (can also be a nibble or octet)
```

Let's take the following list of addresses as an example:

ABCD, ABCE, AFGH

The addresses all have a common prefix A, so we can write A once followed by the list of suffixes BCD, BCE and FGH. i.e.

```
A { BCD, BCE, FGH }
```

Two of these suffixes share a common prefix BC, so we obtain:

```
A { BC { D, E }, FGH }
```

A.2. Processing of a compound address

Following pseudo code parses a compound address and issues a minimum amount of lookup() calls to a routing table engine. We assume the lookup takes a symbol (e.g. one bit) as argument and returns a pointer to it's internal (e.g. tree) structure which can be passed to a next lookup. The parser maintains a stack of such indices. We assume that a meta symbol also denotes the number of non-meta symbols that follow (see encoding in A.3.).

```
index=routing_table_root;     /* routing table entry point */
skip=0;                       /* flag used to skip over irrelevant
                               * parts of the compound address */
getch(meta);                  /* get first meta symbol. Only the
                               * length is used */
while(1) {
  len=LEN(meta);              /* number of non-meta symbols that
                               * follow */
  for (i=0; i<len; i++) {     /* parse len symbols as
                               * address_fragment */
    getch(symbol);            /* read next symbol from the compound
                               * address */
    if (!skip) {
      lookup(symbol, &index); /* search in the routing table, using
                               * symbol, starting from index, storing
                               * the resulting location back in index
                               */
      if (IS_LEAF(index)) {   /* the next-hop is determined */
        forward(index);       /* forward packet to this nexthop */
        skip=stack_pointer;   /* skip to end of this suffix */
      }
    }
  }
  getch(meta);                /* get next meta symbol */
  if (IS_OPEN_BRACKET(meta))
    push(index);              /* push index on the stack */
  if (skip==stack_pointer)
    skip=0;                   /* we've skipped the list_element */
  if (IS_CLOSE_BRACKET(meta))
    pop(index);               /* pop index from the stack */
  if (IS_COMMA(meta) && skip)
```

```
       index=stack_pointer;      /* take the index on top of the stack */
   }
```

A.3. Compound address encoding example

Below is a possible encoding scheme for a compound address which limits the amount of overhead.

CLM header:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |VERSION|UNUSED |    PROT ID    |           LENGTH              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          CHECKSUM             |D|ENC=1|          RUN          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    NUM META   |           meta symbols  ...                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 meta symbols + padding                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                address_fragments, not byte aligned            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NUM META = number of meta symbols

Meta symbol:

```
      0 1 2 3 4 5 6 7
    +-+-+-+-+-+-+-+-+
    |X|sym| length  |
    +-+-+-+-+-+-+-+-+
```

```
   X       = unused
   sym     = meta symbol
             00 = open bracket
             01 = close bracket
             10 = comma
             11 = close bracket followed by comma

   length  = number of bits of the address_fragment (symbol) following this
             meta symbol.
```

References

[AGUI] L. Aguilar, "Datagram Routing for Internet Multicasting", Sigcomm84, March 1984.

[DEER] S. Deering, "Multicast Routing in a datagram internetwork", PhD thesis, December 1991.

[FARI] D. Farinacci, "Multicast Source Discovery Protocol", draft-farinacci-msdp-00.txt, June 1998.

[GRAF] C. Graff, "IPv4 Option for Sender Directed Multi-Destination Delivery", RFC1770, March 1995.

[HAND] M. Handley, H. Schulzrinne, E. Schooler, J, Rosenberg, "SIP: Session Initiation Protocol", RFC2543, March 1999.

[HOLB] H. Holbrook, "Express Multicast: Making Multicast Economically Viable".

[OIKA] J. Oikarinen, D. Reed, "Internet Relay Chat Protocol", RFC1459, May 1993.

[PERL] R. Perlman, "Simple Multicast: A design for Simple, Low-overhead Multicast", draft-perlman-simple-multicast-02.txt, February 1999.

Authors Addresses

Dirk Ooms
Alcatel Corporate Research Center
Fr. Wellesplein 1, 2018 Antwerpen, Belgium.
Phone : 32-3-240-4732
Fax : 32-3-240-9932
E-mail: Dirk.Ooms@alcatel.be

Wim Livens
Alcatel Corporate Research Center
Fr. Wellesplein 1, 2018 Antwerpen, Belgium.
Phone : 32-3-240-7570
E-mail: Wim.Livens@alcatel.be