Internet Engineering Task Force
INTERNET DRAFT
draft-crowcroft-apex-multicast-01.txt
October 2 2001

Jon Crowcroft
UCL
Ken Carlberg
UCL

MAPEX:
Application Level Multicast Architectural Requirements for APEX

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

This is a memo that presents an approach to do application layer multicast. Our intention is to present a design that produces a tree distribution structure within the existing design structure of APEX and BEEP. It is not our intention to present a design that replaces multicast at the network layer. Where the need exists (and in scenarios where it would seem symbiotic), we would find it constructive to integrate the two technologies and perspective (i.e., application later and network layer multicast).

Assumptions

We are using APEX on top of BEEP on top of TCP, as the basic "hop".

The general initial service requirement is for a one-to-many eventually leading to a higher level many-to-many service. The many-to-many service, if needed, may be constructed by configuring/associating a set of one-to-many APEX.

Considerations must include:

- Group identifiers and management
  - (allocation/revokation/distribution)
- Group dynamics
  - long or short lived
    - (we think long lived)
  - high or low rate change of membership
    - (we think potentially high rate, but aggregatable within domains and thus low rate with respect to APEX)
- Topology
  - likely "hop" count
    - hierarchy of hops into clusters?
    - managed or self organised clusters?
  - sparse or dense
  - tree depth/breadth
  - metrics (delay, throughput, fanout)
- Extensibility/Flexibility
  - Reprogrammable
- Transparency
  - Aware of lower level topology?
    - (we think no)
  - Aware of lower level IP multicast capability?
    - (not usually -- or at least not dependant on it)
  - Aware of lower level considerations like AS boundaries?
    - not yet - could possibly be brought in indirectly
  - Aware of lower level considerations like NAT
    - (not directly)

Introduction

APEX [1] provides a very high level abstraction for a "datagram" service. The purpose of this document is to capture the architectural requirements for extending this service to provide "multicast". Since this is effectively an application level service, we make no assumpotions at all about the network layer (i.e. whether

or not IP multicast is available). This is separated from any consideration here by at least 2 levels -- BEEP and the transport layer [2]. If IP multicast was present, then BEEP might use a one-to-many reliable multicast transport, such as PGM [4], in place of TCP for "local" communication between neighbouring APEX services [3] instead of a set of TCP connections. This is for later exploration.

We assume that initially we are building up from one-to-one, to one-to-many, and eventually to many-to-many -- the latter two representing a multicast type of distribution model.

There are various goals for a multicast topology construction and maintenance protocol.

Scaling servers
Simplification of Application implementation.
Reduction in "network/link" traffic

The order here is significant. Since we are operating way about the network level, we do not really have (much) knowledge of the network topology or link performance, so the main goal here is to provide a "natural" interface for group application programmers, and to provide something to allow massive scalability of group communication by effectively combining this with a load balancing scheme. e.g. a group of a million APEX end points cannot be serviced simultaneously by 1 APEX end point, but 1000 APEX relays can distribute messages easily to many more if organised correctly.

Group Identification and its management

communication usually requires a group name which has, usually in IP level multicast systems (except in Express/SSM [5]), been devoid of topological significance. We have no need to retain that idea. In our model, we will explicitly attach a group name to a domain - this makes the allocation task (basically collision avoidance) simple.

An APEX end point entity is enhanced to add a group name allocation facility - group@domain (to be discussed further). This is then distributed over a "well known" apex session channel. To all APEX relays, end points indicate their interest in the channel by "joining" the channel by sending a message to the multicast service in the endpoint's administrative domain; e.g., apex=multicast@example.com, which would act as the entity acting as the nearest relay. Note that the apex core requires that apex services be co-resident with the relays.

Topology Construction

Normally, to date, IP multicast has been built by reversing paths taken from the Unicast routing table. DVMRP [6] uses its own distance vector (RIP basically:-) protocol to build the forward table and then uses a flood and prune, data driven approach to building the tree. PIM DM [7] is similar, except that it relies on the underlying unicast routing, which we don't have in APEX. But in any case, neither scale on an interdomain basis.

MOSPF [8] adds membership reports to link state messages - this does not look to silly here given the likely size of the APEX relay network. We'll look at this in more detail in a moment. CBT [9] and PIM-bidir [10] are aimed at many-to-many applications with a low average delay per given source, but both rely on managers knowing where to place the core or RP. This is mainly concerned with link traffic optimisation aspect of ip multicast, and is of low importance in APEX. PIM SSM provides a source based channel and looks like a very nice approach for APEX, except that it requires the underlying SPF calculation that was done for unicast, to provide the shortest path from receiver to sender as the way to graft a path from the existing tree rooted at a sender, downstream towards the receiver. In other words, with respect to APEX, it is using an upstream perspective to instantiate a downstream tree. Further, it is bound to the (typical) single metric unicast routing available at the network layer. Since APEX is application level oriented, its routing perspective has the potential to be more versatile because of the smaller scale of nodes participating.

So what we need to do here is look at:

a) how we learn the topology of APEX servers (an OSPF like "link-state" protocol may be appropriate, and one that combines group identifier distribution with state flooding might be quite neat).

b) what metrics we use then to buid a "shortest" path

There are three pieces to this, and we probably want to keep this programmable (i.e. extensible) but provide some default behaviour.

Neighbourhoods

An APEX relay knows about some other relays - lets call this a neighbourhood. We can "flood" neighbourhood information to all APEX servers. This gives everyone an APEX Routing Information Base.

Top Down

External configuation gives local and neighbourhood, and global scope constraints - e.g. we need preferences (a la MX) for delay, throughput and fanout - APEX relay fanout is locally controlled.

Delay is an end user preferences -- throughput is measured by a collection of APEX servers and is therefore somewhere between. Fanout, and promoting a high degree towards stub domains or APEX end points, would be considered a network (or source APEX end point) preference with respect to attempting to minimize overall state in the network.

Different APEX users may desire a path to a group that minimises delay AND maximiss throughput - this is NP hard, but there are approximations. For now, we are thinking in terms of a single APEX-relay-hop metric for this part of the scheme. This can be built on later by discovery means.

Bottom Up

YAM like [11,12] -- an APEX end user receivers could graft by doing a YAM like one-to-many join to a list of APEX relays (potentially in more than one neighbourhood) - each APEX relay looks at the join parameters and decides to respond/bind depending on the match, plus its own (e.g. fanout) constraints.

We would consider this approach an optimization effort by downstream endpoints subsequent to an initial top down construction of the tree. Regardless of whether a YAM like algorithm is used, our intention would be to segment optimization of the tree in order to allow different algorithms to be developed, as well as to retain the simplicity of the initial top down construction

The implication of such a design is that we don't have to flood several metrics, and the corresponding changes in their condition. We only flood the relatively stable one of hop count for the top down construction, and then 'discover' on-demand the condition of other metrics (if so desired).

Summary, Conclusions, and Comments

We need some APEX topology and group management protocol elements

1) join/leave messages, with metrics
2) group distribution/revokation
3) "link" state advertisement/flood
   - top down on hops, bottom up on other metrics

We need some tree building code: basically Dijkstra (or incremental Dijkstra if you prefer), plus RPF code. Our preference is to retain a measure of simplicity in the initial tree construction that allows us to take advantage of the alternative paths available from a link-state algorithm.

We may want to allow for native ip multicast in stub domains. The natural question that arises is: what is the protection against loops between native multicast at the network level and application level multicast. This may have additional implications with respect to scoping -- possibly setting TTL scopes for intra-domain distribution, and yet assigning a set of multicast addresses for inter-domain APEX distribution.

Another issue that probably needs to be addressed is the issue of NATs. It may be a case of using NAT boxes as APEX gateways between native (TTL scoped) multicast at a source/destination domain, and TCP unicast distribution of APEX.

Questions to Consider:

Is MAPEX aware of lower level topology?

Right now, we'd say no. just aware of the server topology

Is Mapex aware of lower level considerations like AS boundaries?

At the moment, No. we'd say that APEX is naturally "aware" of the domain boundaries, but not the more abstract (or aggregated) level of AS boundaries.

Other than a combination of link-state advertisements and YAM, are there other application level-like approaches to consider?

Yes, Yallcast [15] and Application Level Active Networking (ALAN) [16,17].

Acknowledgements

We would like to thanks Marshall Rose for some clarifications of an earlier draft.

References

[1] M. Rose, D. Crocker, G. Klyne, "The APEX Presence Service", Work in Progress, Internet-Draft, draft-mrose-apex-core-02.txt, 2/20/2001

[2] M. Rose, "The Blocks eXtensible eXchange Protocol Framework",

Work in Progress, Internet-draft, draft-mrose-bxxp-framework-01.txt, 6/16/2000

[3] M. Rose, "Mapping the BXXP Framework onto TCP", Work in Progress, Internet Draft, draft-mrose-bxxp-tcpmapping-01.txt, 7/13/2000

[4] T. Speakman, et al, "PGM Reliable Transport Protocol", Work in Progress, Internet-Draft, draft-speakman-pgm-spec-06.txt, 2/13/2001

[5] H. Holbrook, B. Cain, "Source-Specific Multicast", Work in Progress, Internet-Draft, draft-holbrook-ssm-arch-01.txt, 11/24/2001

[6] T. Pusateri, "Distance Vector Multicast Routing Protocol", Work in Progress, Internet-Draft, draft-ietf-idmr-dvmrp-v3-10.txt, 9/2000

[7] Deering, et. al, "Protocol Independent Multicast Version 2 Dense Mode Specification, Work In Progress, Internet Draft, draft-ietf-pim-v2-dm-01.txt, November 1998

[8] J. Moy, "Multicast Extensions to OSPF", Request For Comments 1584, IETF, March, 1994

[9] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture", Request For Comments 2201, IETF, Sep. 1997

[10] M. Handley, I. Kouvelas, L. Vicisano, "Bi-directional Protocol Independent Multicast (BIDIR-PIM)", Work in Progress, Internet-Draft, draft-ietf-pim-bidir-01.txt, 11/23/2000

[11] K. Carlberg, J. Crowcroft, "Building Shared Trees Using a One-to-Many Joining Mechanism", ACM Computer Communications Review, Jan. 1997

[12] M. Faloutsos, et. al., "QoSMIC: Quality of Service sensitive Multicast Internet Protocol (QoSMIC), Proceedings of ACM SIGCOMM'98, Sept. 1998.

[13] D. Frigioni, et. al., "Fully Dynamic Algorithms for Maintaining Shortest Paths Trees", Journal of Algorithms, vol. 34, (2000), pp. 351-381.

[14] S. Cicerone, et. al.,, "A Fully Dynamic Algorithm for Distributed Shortest Paths", Proceedings of the Latin American Theoretical INformatics (LATIN2000). Lecture Notes in Computer Science, 1776, pp. 247-256.

[15] Yallcast, Presentation by P. Francis at on Reliable Multicast workshop, May, 1999

[16] Application level multicast CMU:-Narada/RDP, http://www.cs.cmu.edu/~hzhang/multicast/other/endsystem-index.html (see ref to degree bounded k-spanner, constraint based tree formation)

[17] M. Fry, A. Ghosh, "Application Level Active Networking", Fourth International Workshop on High Performance Protocol Architectures (HIPPARCH '98), June 1998.

Other reference to consider...

Application level multicast CMU:-Narada/RDP, http://www.cs.cmu.edu/~hzhang/multicast/other/endsystem-index.html (see esp. ref to degree bounded k-spanner, constraint based tree formation!)