Multi-Destination Routing and the Design of Peer-to-Peer Overlays

John Buford
Panasonic Princeton Lab
Princeton, NJ, USA
buford@research.panasonic.com

Alan Brown
University of Stirling
Stirling, Scotland, FK9 4LA
abr@cs.stir.ac.uk

Mario Kolberg
University of Stirling
Stirling, Scotland, FK9 4LA
mko@cs.stir.ac.uk

Abstract

We propose the integration of peer-to-peer network overlays with underlay networking in which multi-destination multicast routing is available. Network overlay operations are parallelized by using multi-destination multicast messages in the underlying network in place of same-source unicast messages. This mechanism is generally applicable to structured overlays including one-hop, multi-hop, and variable-hop, and unstructured overlays. The main result is significant message reduction, which varies according to the overlay algorithm.

1. Introduction

We are interested in improving the performance of peer-to-peer overlays by mapping overlay messaging to native multicast paths for overlay operations that are inherently parallel. Here we describe and analyze the applicability of multi-destination multicast routing to several different categories of overlays. Separately [1] we have investigated the impact of using multi-destination multicast routing in the underlay to support overlay operations in the EpiChord [2] 1-hop overlay, and have shown through simulation that multi-destination addressing in EpiChord achieves about 30% message reduction for both edge and internals links. We have also investigated [3] several improvements to the EDRA [4] one-hop overlay maintenance algorithm which include the use of multi-destination multicast routing.

This paper contains the following contributions:

- We formulate criteria for determining whether overlay messages can be parallelized using multicast. These criteria are maximum group size, number of groups, the time to create a new multicast group, and group formation rate.
- We show how multi-destination routing can be used in several categories of overlays for various overlay operations including DHT operations, overlay maintenance, replication, and measurement.
- We estimate the message savings based on the Chuang-Sirbu [5] formulation of multicast efficiency.

The remainder of this paper is organized as follows. We formulate the criteria for evaluating the use of multicast messaging in the next section, followed by a review of related work in Section 3. Section 4 describes and analyzes multi-destination routing in several overlay categories and message operations. Section 5 summarizes the analysis and Section 6 concludes the paper.

2. Problem statement

2.1 Criteria for multicast messaging

The typical use of an overlay is to provide widely available end-to-end network services that would be difficult to deploy in the network, or to share computing resources among a large set of users. Overlay messaging includes operations such as join/leave, bootstrap, routing table exchanges, DHT lookup and insert, and probes.

Let P be the set of peers in the overlay during some interval T, where $|P| = n$. Let M be the set of overlay protocol message types for the overlay, $M_t \in M$ is one of the above message types, and m_j is a message instance of a given type M_t, with j as a unique identifier for each message instance in interval T. Define F_i as the set of all combinations of P of sizes i = 2, 3, … n.

We use an outgoing message queue at each peer to identify temporal locality needed for using a multicast message in place of unicast messages. Each peer p has a queue Q which has pending messages m_j to send. After adding a message m_j to Q, the peer examines Q and may combine a set u of messages in Q to create a new multicast message m_c to group g_k where m_c contains the contents of the individual u messages, $p \notin g_k$, $|g_k| = |u|$, $g_k \in F_{|u|}$, and k is a unique group identifier. The peer may flush one or more messages from Q, combine other unicast and multicast messages in the queue, or wait for further messages. The peer acts to maintain the maximum queueing delay of any message below a threshhold d_q. Assume peers agree on the rules for combining and extracting unicast messages to and from multicast messages. Assume further that the decision process by which messages are combined considers that the benefit of multicast for network efficiency is proportional to the amount of overlap of the content of the combined unicast messages.

The number of multicast groups used in the overlay in the interval T is then $N_G = |G|$ where $G = \{g_i : \forall i\}$. The maximum group size is $|g_{max}|$ such that $\forall\ g_k : |g_k| \leq$

$|g_{max}|$ and $\exists\ g_k : |g_k| = |g_{max}|$. The rate of group formation is $r = N_G/T$ and the frequency of group use $f(g_k) = |m_c|/T$, the number of multicast messages m_c to the group g_k in interval T.

Multicast routing offers efficiency and concurrency to overlay designers. It may improve response time for operations in which parallelism locates a shorter path more quickly. Reliability may be effected, since a lost multicast packet effects multiple operations.

For multicast to be practical it is necessary that:

1. The scalability of the multicast algorithm for number of groups meets the scalability requirements of the overlay. If C is the capacity of the network to support simultaneous multicast group state for this overlay, then $N_G \leq C$. Likewise, if v is the maximum group size, then $|g_{max}| < v$.
2. The overlay's rate r of group formation and group membership change be attainable by the multicast mechanism. The time to create a new multicast group $t_c < d_q$.

Preferably group join in the multicast protocols leverages the overlay formation methods.

2.2 Host-group multicast

The prevailing host-group multicast protocols including PIM-DM, DVRMP, and CBT create a group address per multicast tree, and each router stores state for each active group address. The state in the router grows with the number of simultaneous multicast groups. There is delay to create a group, and the network may have a limited number of group addresses.

For a large overlay it is impractical for each node to have a group address for each set of other nodes it sends multicast messages to. Suppose N = 1M, T = 60min, and each peer conservatively uses 5 groups for those $m \in M$ it parallelizes, so $N_G = 5\ N$. Worsening the problem is that the peer session time is as low as 60 minutes in some overlays, meaning that group state is replaced relatively frequently.

Host group multicast is designed for relatively small numbers of very large sets of recipients. So host group multicast is not a good choice for use in parallelizing network overlay operations where there are many simultaneous small groups of peers involved in a message, and the groups are short-lived.

2.3 Multi-destination multicast

The concept of multi-destination routing was proposed in the early years of multicast protocol design [6], but as Ammar observes [7], subsequent protocol design focused on enabling large multicast groups. However in the past several years, there has been recognition of multi-destination routing as a complementary multicast technology that has advantages for applications which feature large numbers of small groups.

In multi-destination multicast routing, instead of two or more unicast messages sent to separate destinations, a single message is sent containing the list of the destinations and the message content from the original messages. Multicast-enabled routers route the message until a split point is reached (according to unicast routing decisions). At each such point, duplicate messages containing the subset of destinations for each forwarding path are created and routed. This continues until a message contains only a single address in which case it is converted to a unicast message and is routed to its destination.

If only a subset of all routers are multicast-enabled, these routers forward multicast packets to other multicast routers using tunnels through unicast routers. Alternately, the network contains hosts which implement the multi-destination multicast routing. Overlay nodes sending a multi-destination multicast message send the message to one such host. The host routes the multi-destination messages to other hosts according to the network routing rules. At each such host, duplicate messages containing the subset of destinations for each forwarding path are created and routed. This continues until a message contains only a single address in which case it is converted to a unicast message and is routed to its destination.

Multi-destination multicast routing does not require state in routers. Thus there is no router state constraint on N_G. However there is additional processing overhead at each router for the forwarding algorithm to process the list of addresses. Whereas host-group multicast router has forwarding state for each group address, for a multi-destination packet with N destinations, there is O(N) work at each router to process the list of addresses and make a forwarding decision for each destination. Packet duplication work for multi-destination routing is similar to that of host-group multicast.

Multi-destination multicast imposes a maximum group size v. Practical values for v appear to be not more than 50. Since peers in the overlay maintain routing tables or addresses of other peers, there is no group join overhead when peers are directly reachable in the overlay. Thus the time to create a new multicast group t_c is not a factor.

Recently an experimental IP protocol for multi-destination multicast called explicit multicast (XCAST) protocol has been specified [8] and several XCAST testbeds have been deployed. He and Ammar [9] analyze the performance of XCAST combined with host-group multicasting.

2.4 Selection and integration

A peer joins an overlay and through a combination of configuration and discovery determines whether a multicast mechanism is available in the underlay. Depending on how many different message types M are used in the overlay and the inherent parallelism of the associated operations, the overlay itself can issue those messages as multicast messages. In addition, as described in Section 1, outgoing messages can be

queued and messages containing overlapping content can be combined.

3. Related work

Oh-ishi et al. have considered the use of Protocol Independent Multicast (PIM) [10] in sparse mode (PIM-SM) and source specific mode (PIM-SSM) [11] to reduce message traffic in peer-to-peer systems. Their analysis focuses on using multicast routes between peers in different ISP networks.

He and Ammar [9] analyze the performance of XCAST combined with host-group multicasting, where XCAST is used for small groups and host-group multicasting is used for large groups. For XCAST sessions they use a dynamic tunneling mechanism between routers corresponding to XCAST branch points in a given session. Since most routers in a multicast path are non-branching, the XCAST routing processing in each router is significantly reduced. However this mechanism is session-oriented and would not be useful for parallelizing overlay operations which use short-lived groups.

The quantitative benefits of multicasting have been formulated in the Chuang-Sirbu [5] scaling law which shows that the ratio of the number of edges in the multicast tree versus the average unicast path length equals $m^{0.8}$ (where m is the multicast group size). The per link reduction in message traffic from multicast is then $(1\text{-}m^{-0.2})$. The exponent $\varepsilon = -0.2$ represents the multicast efficiency. Further evaluation of Chuang-Sirbu has been done in [12] which derives another similar expression and confirms it with respect to various networks, and [13] which finds some shortcomings of Chuang-Sirbu with respect to large groups and provides a revised formulation. Chalmers and Almeroth [14] using actual multicast data sets on the Internet and synthesized multicast trees find a slightly lower multicast efficiency ε in the range $-0.34 < \varepsilon < -0.30$. These results are based on actual multicast infrastructure in place at the time of the data collection which may constrain multicast branching points more so than in synthetic topologies. Further, their analysis includes multicast trees extended to the end points, which produces increased savings from Chuang-Sirbu if there is clustering of end points at subnets.

In the case of parallelized overlay operations discussed here, the size of the multicast group is within the Chuang-Sirbu formulation but is frequently at the lower end of the formulation, $m < 20$. In this range, the multicast efficiency exponent derived by Chalmers and Almeroth in [14] is also applicable. For brevity, in this paper we refer to multicast savings $1\text{-}m^{\varepsilon}$ with $-0.34 < \varepsilon < -0.20$ as the Chuang-Sirbu law. So 5-way multicast would provide a 28% to 42% savings compared to unicast according to this rule.

4. Examples

4.1 Kademlia – multi-hop overlay

Kademlia [13] is a multi-hop overlay that by virtue of its symmetric distance metric (the XOR function) is able to issue parallel requests for its routing table maintenance, lookups and puts.

During a node lookup, a peer computes the XOR distance to the node, looks in the corresponding k-bucket to select the α-closest nodes that it knows of already, and transmits parallel requests to these peers. Responses return closer nodes. Kademlia iteratively sends additional parallel requests to the α-closest nodes until it has received responses from the k-closest nodes it has seen. A typical value of α is 3. Figure 1 shows a node lookup for a node in the 110 k-bucket. For a 160-bit address space there will be up to 160 buckets.

Node lookup is used by other Kademlia operations including DHT store, DHT refresh, and DHT lookup. A Kademlia peer does at least k/α iterations for a node lookup in a given bucket. For $k = 20$ and $\alpha = 3$, that is 3-way queries to seven multicast groups. With 160 buckets each peer would need at least 160 groups to do queries across its address space. If the multicast queries were α-way, Chuang-Sirbu estimates a 18% savings, and if the queries were k-way, k=20, Chuang-Sirbu estimates a 45% to 64% savings from multicasting Kademlia requests, although responses would be unicasted.

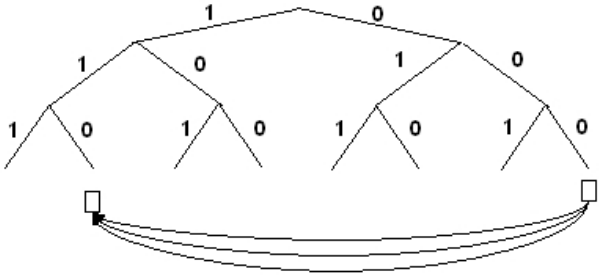

Figure 1 Kademlia node lookup using α=3 to nodes in k-bucket 110, i.e., nodes whose distance is in the range $[2^5..2^6)$

4.2 Meridian – measurement overlay

Meridian [16] is a measurement overlay in which relative distance from other nodes in the overlay is used for solving overlay lookups like closest node discovery and central leader election. Each peer organizes its adjacent nodes into a set of concentric rings, each ring contains $k = O(\log N)$ primary entries and l secondary entries. In a simulation of N=2500 nodes, k=16, and the number of rings $i^* = 9$.

Meridian uses a gossip protocol to propagate membership changes in the overlay. During a gossip period, a message is sent to a randomly selected node in each of its rings. The message contains one node randomly selected from each of its rings. Unicast gossip messages can be multicast to i^* destinations using a

single i*-way message. For i* = 9, the savings is 36% to 53% over unicast messaging.

4.3 EpiChord – O(1)-hop overlay

In EpiChord [2], peers approach 1-hop performance on DHT operations compared to the O(log N) hop performance of multi-hop overlays, at the cost of the increased routing table updates and storage.

To improve the success of lookups, EpiChord uses iterative *p-way* requests directed to peers nearest to the node. During periods of high churn, a peer maintains at least 2 active entries in each slice of its routing table. When the number of entries in a slice falls below 2, the peer issues parallel lookup messages to ids in the slice. Responses to these lookups are used to add entries to that slice in the routing table.

If parallel unicast lookup messages and slice refresh messages are replaced with a single multi-destination packet [1], this can reduce the number of lookup messages by up to 32% for edge links and 31% for internal links over 5-way unicast. Alternately, for a given message load, a higher routing table accuracy can be obtained. Note that p-way EpiChord results in parallel message traffic that is on average less than p-way due to invalid routing table entries, re-transmissions, and negative acknowledgements [1].

4.4 Accordion – variable hop overlay

Accordion [17] is a variable hop overlay, in which a peer limits its routing table update message level based on its available bandwidth. During periods of low bandwidth, routing table accuracy can approach that of multi-hop overlays while for higher bandwidth, routing table accuracy reaches one-hop.

Unlike Kademlia and EpiChord, Accordion uses *recursive* parallel lookups so as to maintain fresh routing table entries in its neighborhood of the overlay and reduce the probability of timeout. The peer requesting the lookup selects destinations based on the key and also gaps in its routing table. Responses to forwarded lookups contain entries for these routing table gaps. Note that recursive parallel lookups create more load on the target peer compared to iterative parallel lookups, since the target node receives p messages for each request.

Excess bandwidth is used for parallel exploratory lookups to obtain routing table entries for the largest scaled gaps in the peer's routing table. The degree of parallelism is dynamically adjusted based on level of lookup traffic and bandwidth budget, up to a maximum configuration such as 6-way.

Replacing Accordion p-way forwarded and exploratory lookups with multi-destination lookups will reduce edge traffic by (p-1)/2p; e.g., p=5 means 40% reduction on the edge. For a fixed bandwidth budget, this means that a peer can increase its exploration rate by factor of 2.5, substantially improving routing table accuracy. Alternately, a peer can operate at the same level of routing table accuracy (and number of hops per lookup) for a lower bandwidth budget.

4.5 EDRA maintenance algorithm

D1HT [4] is a one-hop overlay that defines the overlay maintenance algorithm EDRA (Event Detection and Reporting Algorithm), where an event is any join/leave action. EDRA propagates all events throughout the system in logarithmic time. Each join/leave event is forwarded to $\log_2(x)$ successor peers at relative positions $\log_2(0)$ through $\log_2(n)$ (Figure 2).

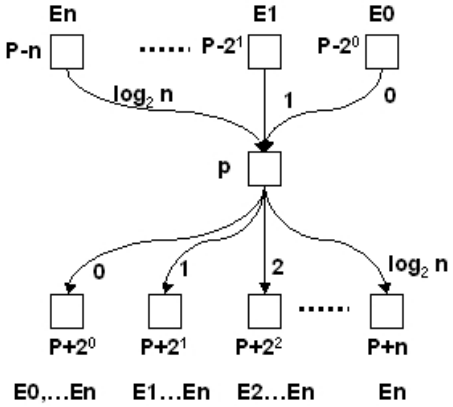

Figure 2 EDRA [4] event propagation as a multicast tree

Following the notation of [4], Θ is the interval at which a peer propagates events to its successors in the ring, and $\rho = \lceil \log_2 n \rceil$ is the maximum number of messages a peer sends in the interval. Propagated events are those directly received as well as those received from predecessors since the last event message. Each message has a time to live (TTL) and is acknowledged. If there are no events to report, only messages with TTL=0 are sent.

During any interval Θ, a peer sends at most $\rho = \lceil \log_2 n \rceil$ messages containing its current events. Each message contains the same set of events but different TTL in the range [0.. ρ). We replace the ρ unicast messages with a ρ-way multi-destination packet containing the set of events and a list of [peer,TTL] pairs. Each peer receiving the message extracts its TTL from the list.

At size n=10^6, Chuang-Sirbu [5] estimate gives up to 64% message reduction savings (ρ = 20). At size n=10^3, Chuang-Sirbu estimate gives up to 54% savings (ρ = 10). Figure 2 shows EDRA event propagation as a multicast tree.

4.6 Replication and load-balancing

Beehive [18] is a replication mechanism for prefix-based multi-hop overlays such as Kademlia and Pastry. Assuming object popularity follows Zipf distribution, Beehive uses object access statistics to proactively push objects to sufficient levels in the overlay to meet the required number of hops per query.

Parallel messaging in Beehive occurs in two areas. First, object access statistics are aggregated at each object's home node and propagated to nodes along the

access path. Second, each peer locally determines for objects that it currently stores, whether the access level for each object requires increased replication and will push the object to other peers that precede it in the prefix-based routing path.

In both cases, the potential parallelism is determined by the prefix size used in the overlay routing mechanism. For base b in a m-bit address space, the tree has up to b-way branching factor with at most m/b hops from root to leaf nodes. However, few nodes will reach b-way branching due to sparseness of the address space, limiting the benefits of multi-destination routing to parallelize Beehive. However it is possible that peers could push replicas and aggregated statistics to several levels to achieve greater message parallelism.

4.7 Overlay multicasting

Several P2P overlays support multicasting. These are referred to as implicit ALM, and include Scribe/Pastry, Bayeux/Tapestry, and NICE. ALM trees suffer from constraints on the in-degree and out-degree of nodes which are using unicast links to connect to parent and children nodes. This increases path length in the tree.

If the network contains hosts which implement the multi-destination multicast routing, overlay nodes sending a multi-destination multicast message send the message to one such host. The host routes the multi-destination messages to other hosts according to the network routing rules. At each such host, duplicate messages containing the subset of destinations for each forwarding path are created and routed. This continues until a message contains only a single address in which case it is converted to a unicast message and is routed to its destination.

Further this integration with ALM and multi-destination routing means that arbitrarily large groups can be created. For example, suppose we limit multi-destination packets to 50 destinations and each node is constrained to say C number of connections. Nevertheless we can form overlay trees of millions of nodes where each node connects to at most C*50 out-going nodes. Each node receiving a single incoming packet forwards it using the set of addresses which is corresponding to its adjacencies.

5. Conclusion

We have shown that parallelizing a variety of overlay routing algorithms using multi-destination multicasting instead of parallel unicast messages results in significantly reduced message traffic on both edge and internal links. In structured overlays, this message reduction occurs for a variety of operations such as joins, routing table maintenance, and application lookups. In general, latency behavior and operational semantics are retained.

We defined criteria for determining whether an overlay message can be parallelized using multicast. These criteria are maximum group size, number of groups, the time to create a new multicast group, group formation rate.

6. References

[1] J. Buford, A. Brown, M. Kolberg. Parallelizing Peer-to-Peer Overlay with Multi-Destination Routing. IEEE Consumer Communications and Networking Conference (CCNC 2007). To appear.

[2] B. Leong, B. Liskov, and E. D. Demaine. EpiChord: Parallelizing the Chord Lookup Algorithm with Reactive Routing State Management. *Computer Communications*, Elsevier Science, Vol. 29, pp. 1243-1259.

[3] J. Buford, A. Brown, M. Kolberg. Evaluation of the EDRA One-Hop Overlay Maintenance Technique. Submitted.

[4] L. Monnerat and C. Amorim. D1HT: A Distributed One Hop Hash Table. In *Proc of the 20th IEEE Intl Parallel & Distributed Processing Symposium (IPDPS)*, April 2006.

[5] J. Chuang and M. Sirbu Pricing multicast communications: A cost-based approach. In *Proc INET'98*

[6] L. Aguilar, Datagram Routing for Internet Multicasting, Sigcomm 84, March 1984.

[7] M. Ammar. Why Johnny Can't Multicast: Lessons about the Evolution of the Internet. Keynote - NOSSDAV 03.

[8] R. Boivie, et al, Explicit Multicast (Xcast) Basic Specification, draft-ooms-xcast-basic-spec-09.txt, Work in Progress. Dec. 2005.

[9] Q. He, M. Ammar. Dynamic Host-Group/Multi-Destination Routing for Multicast Sessions. *J. of Telecommunication Systems*, vol. 28, pp. 409-433, 2005.

[10] T. Oh-ishi, K. Sakai, H. Matsumura, A. Kurokawa, Architecture for a Peer-to-peer Network with IP Multicasting, *18th Intern. Conf. on Advanced Information Networking and Applications (AINA'04)* Vol. 2, 2004.

[11] T. Oh-ishi, K. Sakai, K. Kikuma, and A. Kurokawa. Study of the Relationship between Peer-to-Peer Systems and IP Multicasting. *IEEE Communications Magazine*. Jan. 2003.

[12] G. Phillips, S. Shenker, and H. Tangmunarunkit. Scaling of multicast trees: Comments on the Chuang–Sirbu scaling law. *ACM SIGCOMM'99*.

[13] P. Van Mieghem, G. Hooghiemstra, and R. van der Hofstad. 2001. On the efficiency of multicast. IEEE/ACM Trans. Netw. 9, 6 (Dec. 2001), 719-732.

[14] R. Chalmers and K. Almeroth, Modeling the Branching Characteristics and Efficiency Gains of Global Multicast Trees, *IEEE Infocom*, Anchorage, AK, USA, April 2001.

[15] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Proc of IPTPS02*, Cambridge, USA, March 2002.

[16] B. Wong, A. Slivkins and E. G. Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In Proc. of *SIGCOMM Conference*, Aug 2005.

[17] J. Li, J. Stribling, R. Morris, and M. F. Kaashoek, *Bandwidth-efficient Management of DHT Routing Tables*, NSDI 2005.

[18] V. Ramasubramanian and E. Sirer. Beehive: O(1) Lookup Performance for Power-Law Query Distributions in Peer-to-Peer Overlays. In Proc. of *Networked System Design and Implementation (NSDI)*, March 2004.