Hybrid Multicasting in Large-Scale Service Networks*

Jingwen Jin
Dept. of Computer Science
University of Illinois at Urbana-Champaign
jjin1@cs.uiuc.edu

Klara Nahrstedt
Dept. of Computer Science
University of Illinois at Urbana-Champaign
klara@cs.uiuc.edu

ABSTRACT

The importance of service composition has been widely recognized in the Internet research community due to its high flexibility in allowing development of customized applications. So far little attention has been paid to composite services' runtime performance-related aspects, which are of great importance to wide-area applications. Service composition in the wide area actually creates a new type of routing problem which we call *QoS service routing*. We study this problem in large networks (e.g., the Web) and provide distributed and scalable routing solutions with various optimization goals. Most importantly, we propose ways to reduce redundancies of data delivery and service execution through explorations of different types of multicast (service multicast and data multicast) in one-to-many application scenarios.

Categories and Subject Descriptors: H.3.5 [Information Systems]: Web-based services

General Terms: Performance, Design

Keywords: QoS, service composition, multicast

1. INTRODUCTION

The *component service* model has been proposed in the Internet for service flexibility and reusability and has triggered many useful applications. Imagine a mobile phone user that wants to retrieve the content of a Web document written in language A and hear it through speech in language B. The original data can flow through a stream of services (or a em service path) to get itself transformed before reaching the destination. Composite services can be also useful in one-to-many application scenarios. Imagine the Web news video distribution application that involves a single sender and multiple receivers, each of which requiring the original video content to be customized according to its own resource conditions. Although it is feasible to have end-to-end service paths individually built, such a unicast delivery model may incur waste of bandwidths (due to redundancies of data delivery) and machine resources (due to redundancies of service execution). We propose to build a single service tree, rather than multiple independent service paths, through which the data should be delivered to save both network bandwidths and machine resources. We term such a group delivery model *service multicast*.

Service unicast has been reasonably addressed in the literature [1, 2, 3]. Some of the existing work, e.g., [3], adopts a global planning approach which, concerning its limited scalability, is not suitable for large networks (e.g., the current Web). Scalable routing falls into two approaches: hierarchical [1] and distributed [2]. The routing approach in this paper falls into the latter category. In the unicast context, a distributed solution based on local heuristics has been described in [2]. The local heuristics alone, however, would only help balance the network and machine loads and potentially optimize the path's overall concave or multiplicative metrics (e.g., the path's bottleneck bandwidth or robustness), but would not pose any constraint on the overall service path length, which is an additive metric that requires global optimizations. As a consequence, service paths computed hop-by-hop by adopting local heuristics tend to be long, and inevitably consume more network resources. We remedy this shortcoming by using the geometric information of the network hosts as guidance to compute more delay-efficient paths. (Details are omitted due to space limitations.) Our major focus would be on the less investigated, more challenging *QoS service multicast routing* problem, whose importance is undubious due to resource constraints. For scalability, we devise a fully distributed approach for service multicast based on the remedied unicast solution. Moreover, we propose to further optimize resource usages by integrating data multicast into service multicast. We call such a combined multicast delivery mode *hybrid multicast*.

We make the following assumptions in this paper. (1) The service routing substrate is built on top of a service discovery system enhanced with the hosts' geometric location information. (2) Service paths/trees are built on top of an unorganized overlay topology. However, another organized mesh topology is maintained for general control messages. For communication efficiency, overlay network nodes are connected into a Delaunay triangulation in which control messages can be routed by using on-line routing methods (e.g., compass routing).

2. PURE SERVICE MULTICAST

To support the dynamic membership feature, we take an incremental approach for service multicast tree building. Unlike the conventional data multicast, where every on-tree node functionally qualifies as a graftable node for all other group members, in service multicast, due to the functionality issues, an on-tree node n only qualifies as a graftable node for a member m (whose service request is r) if n's up-tree service path (the service path from the root to n) is a prefix of r. Construction of our service multicast tree will take the following procedures. Each member joining the multicast group sends its request r towards the source through the organized overlay network topology (Delaunay triangulation) by using compass routing. For each overlay node n_i that is hit by the request, it is verified if n_i is an on-tree node. If it is not, then n_i simply forwards the original request to the next hop towards the source, and if it is, it tries to match r and the locally maintained functional

*This work was supported by NSF grants CCR-9988199 and EIA 99-72884 EQ, and NASA grant NAG2-1406.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

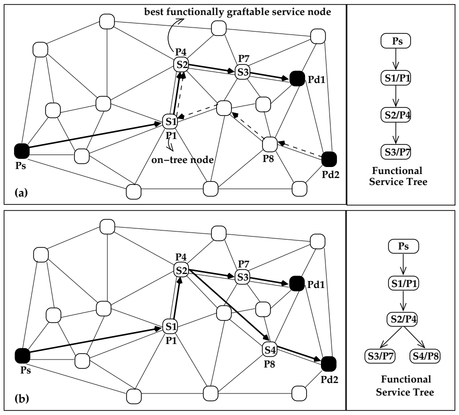

Figure 1: (a) A service request message is sent from the newly joining member p_{d2} towards the source by using compass routing, and the request hit an on-tree node p_1 before it reaches p_s. Since every on-tree node maintains T_f, p_1 found that p_4 is the best graftable node for the current request, thus the request is forwarded to p_4; (b) a service branch is established hop-by-hop from the graftable node p_4 to p_{d2}.

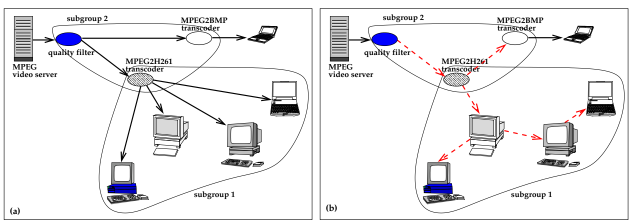

Figure 2: (a) Pure service multicasting; (b) hybrid multicasting (service multicasting + data multicasting).

service tree T_f to identify the best functionally graftable service node n, and forwards the request accordingly. Between n and m, a service branch (for an r's suffix) can be constructed hop-by-hop by using a unicast service routing solution. (Figure 1)

In service multicast, to enable an on-tree node to identify graftable service nodes for others, it needs to keep the functional tree information of the multicast group (due to the service functionality constraints). This implies that whenever the functional aspect of the service tree has been modified, tree state needs to be updated in all current on-tree proxy nodes by broadcasting adequate control messages. It is easy to see that service multicast definitely helps to save proxy machine resources because each service in the functional service tree gets executed only once. It should also reduce network bandwidth usages compared to service unicast, as in most of the cases, we can expect the length of a service branch (satisfying only the suffix of the request) to be shorter than an individually built service path that needs to satisfy the whole request.

3. HYBRID MULTICAST

In pure service multicast, each service branch is directly attached to its best functionally graftable node. However, in such an approach, bandwidth usage may not have been optimized. In Figure 2(a), the proxy offering the MPEG2H261 transcoding service needs to send four separate copies of transformed data to its downstream nodes. This may cause data delivery in those sub-groups to be sub-optimal. First, it may be expensive to do so, because bandwidths need to be separately allocated. Second, after a node's (e.g., the one offering MPEG2H261) outbound network bandwidth usage reaches its limitation, then no new service branches can be created starting from this point. We address these weaknesses by further employing data multicast in the local sub-groups (Figure 2(b)).

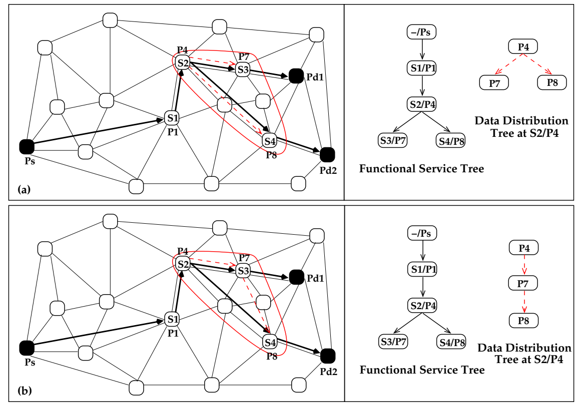

Figure 3: Exploring data multicast in a service multicast scenario: (a) a new service branch's first node, p_8, is initially directly attached to the graftable service node p_4 (p_4 as p_8' parent in the local data distribution tree); (b) p_8 gets parent-switched to p_7 in the data distribution tree.

To realize such a hybrid multicast scenario, the distributed approach requires each on-tree proxy and/or service node to keep two trees: one for the global functional service tree, and the other for local data distribution tree, which we denote as T_f and T_d respectively. The same as in pure service multicast, each on-functional-tree *proxy* will keep an updated T_f, which is the functional service tree of the whole multicast group. In addition to T_f, each on-tree *service node* n also keeps a T_d, whose root is itself, and whose lower-level members are its children in T_f. While the maintenance of T_f is still to enable on-functional-tree nodes to individually search for functionally graftable nodes for other joining requests and is global, T_d is maintained for exploiting benefits of data multicast in subgroups and is local. When a new service branch b gets attached to a graftable node n, initially, n's T_d will have b's first node (say n') attached to itself. However, as n is aware of the geometric locations of its T_d's nodes, it will be able to identify which nodes are closer to n' than itself. If there is any such node, then n will initiate a *parent switching protocol* (details omitted), so that at the end, n' gets attached to a closer parent with sufficient network bandwidth. Note that the parent switching protocol is only for switching parent in the local data distribution tree, it does not affect the global functional service tree. Figure 3 depicts what the global functional service tree and the local data distribution tree would look like in the scenarios. It is clear that with the employment of local data multicast, end-to-end service paths may become longer than in pure service multicast. However, such a performance degradation is justified by saving of network bandwidths.

4. REFERENCES

[1] Jingwen Jin and Klara Nahrstedt. Large-Scale Service Overlay Networking with Distance-Based Clustering. In *ACM/IFIP/USENIX International Middleware Conference (Middleware2003)*, Rio de Janeiro, Brazil, Jun 2003.

[2] Xiaohui Gu, Klara Nahrstedt. A Scalable QoS-Aware Service Aggregation Model for Peer-to-Peer Computing Grids. In *Proc. of High Performance Distributed Computing*, Edinburgh, Scotland, Jul 2002.

[3] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, Quan Z. Sheng. Quality Driven Web Services Composition. In *The Twelfth International World Wide Web Conference*, Budapest, Hungary, May 2003.