An Efficient Multicasting Architecture for Context-Aware Messaging Services in the Future Internet

Hojin Lee, Bokgyun Jeon, Soyoung Park, Taekyoung Kwon, and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {lumiere, jeonbg, sypark}@mmlab.snu.ac.kr and {tkkwon, yhchoi}@snu.ac.kr

***Abstract*— Due to large control overhead and latency, traditional multicast and unicast mechanisms are not suitable for context-aware messaging services where multiple nodes subject to a particular context receive several appropriate messages from a particular server. We propose novel multicast architecture for context-aware messaging services. By the aid of additional infrastructure our mechanism builds a multicast tree in a top-down manner from a source to multiple receivers without clients' JOIN operations while the first data message is delivered. Simulation results show that our mechanism has significantly lower overhead and latency than traditional unicast and multicast mechanisms.**

***Keywords*—Context, multicast, architecture, Xcast, future Internet**

I. Introduction

In this paper, we propose an efficient delivery architecture for context-aware messaging services where multiple nodes subject to a particular context receive several appropriate messages (e.g., the recommendation of the new path changed due to car traffic congestion, and notification of speed reduction due to car accidents in front) from a context-aware messaging server (CAMS). We assume the context is sensed by nodes themselves or sensors that have Internet connectivity.

The message delivery mechanisms are classified into pull- and push-based methods [3]. In the pull-based method, clients or nodes explicitly request messages for a particular context to the CAMS whereas messages are sent out to clients without the explicit solicitation in the push-based method. The pull-based method is conventional in the current Internet, e.g., HTTP, Music on Demand and, Video on Demand. In the pull-based method, nodes detecting a particular context request the corresponding messages from the CAMS individually if there is no multicast mechanism.

To deliver messages efficiently to multiple nodes, IP multicasting can be used. Since the number of the destination in the IP header is fixed as one, a specific IP address (i.e., a class D address) represents multiple nodes by a single multicast group address. To map multiple nodes to a single multicast group address, each node performs JOIN operations. Through the JOIN operations, a multicast tree is also built bottom-up (from the last-hop router to the first-hop router[1]). However, multicasting initiated by nodes incurs more latency and control overhead due to additional procedures to acquire the multicast group address and to join the multicast tree.

To provide context-aware services efficiently and effectively, additional infrastructure is encouraged as shown in [1]. Our proposed scheme exploits the infrastructure (Context Server abbreviated by CS) in a push-based manner in terms of clients. In our proposed scheme, clients do not explicitly request messages by the aid of CS. Therefore, latency is lower than existing schemes. Furthermore, our scheme uses explicit multicast (Xcast) [4] where multiple destinations are explicitly encoded in a packet header. Owing to Xcast and the aid of CS, individual nodes' JOIN operations are not required, which leads to less control overhead than IP multicasting. To realize our scheme, we need a flexible network layer header structure, which is believed to be feasible in the future Internet.

The rest of this paper is organized as follows, In Section II we explain our proposed mechanism and the performance evaluation is presented in Section III. Finally, the paper is concluded with future work in Section IV.

II. Proposed Mechanism

We assume that the CS maps between the context and the last-hop routers of the target nodes thanks to mobility management technologies in the future Internet. When the context occurs, the CS sends a ReportContext message to the CAMS. The ReportContext message includes the context information and the addresses of last-hop routers.

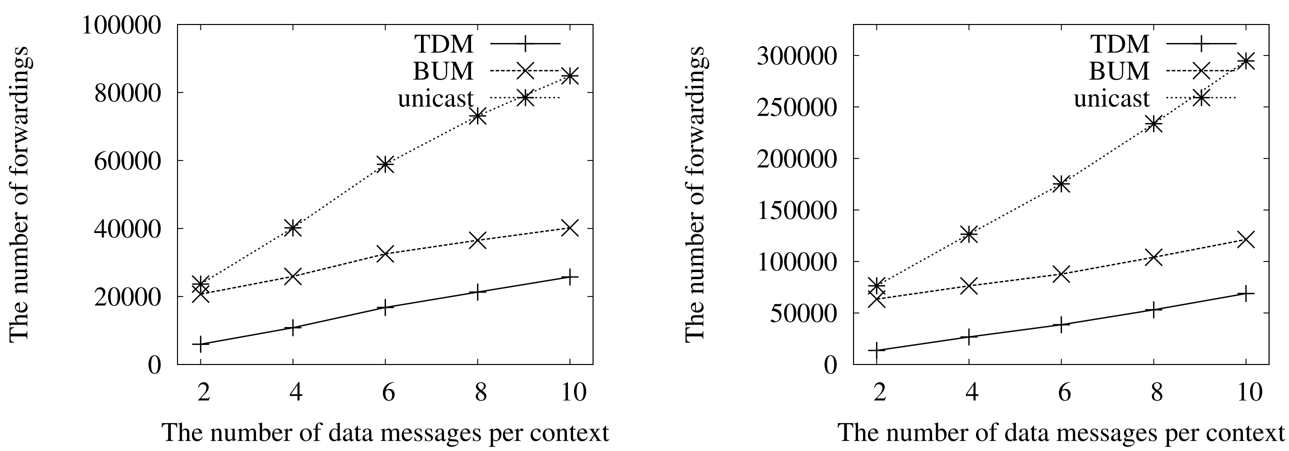
Our proposed scheme constructs a multicast tree in a top-down manner because a multicast source (i.e., the CAMS) knows the last-hop routers of all the destinations. We assume that the network layer header in the future Internet can have multiple destination addresses. (A similar idea is proposed in

[0]This work was supported in part by the IT R&D program of MIC/IITA [2007-F-038-01, Fundamental Technologies for the Future Internet]. The ICT at Seoul National University provides research facilities for this study.

[1]A last-hop router is the closet one to a receiver, and a first-hop router to a multicast source.

TABLE I
SIMULATION SETUP

Parameter	Value
Genral setup	
Simulation time	300 s
Simulation area	10 km $\times$ 10 km
Topology tool (routers)	GT-ITM [5]
# of routers	100
Link characteristics between routers	
Bandwidth	100 Mbps
Propagation delay	10 ms
# of subnetworks	50
# of nodes in a subnetwork	uniform distribution in [10, 20]
Link characteristics between a router and a node	
Shared medium (e.g., wireless)	
Bandwidth	10 Mbps
Propagation delay	1 us
Transmission range	1 km
Context characteristics	
# of CAMS	10
Context interval per CAMS	exponential distribution with average 10 s
Context radius	2 km
# of messages per context	2, 4, 6, 8, 10



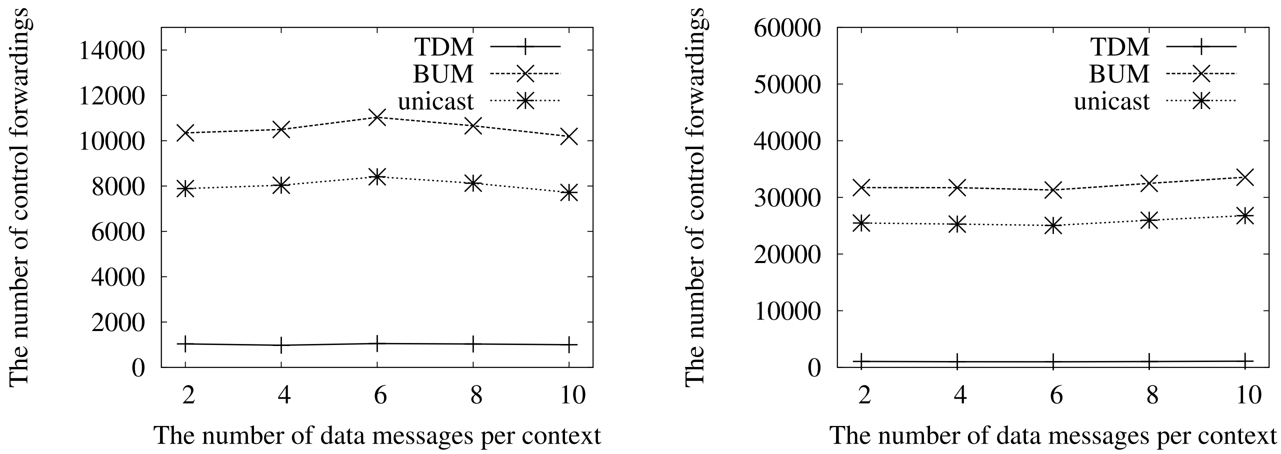
(a) Subscriber ratio 0.1 (b) Subscriber ratio 0.3

Fig. 1. The number of forwardings with respect to the number of data messages per context

the form of a shim header in [2].) When the CS sends the first data message, Xcast [4] is used; all the addresses of all the last-hop routers and a multicast group address are written in the destination field. A multicast group address is assigned to each multicast CAMS flow by the CAMS. The pair of the multicast group address and the CAMS address is unique globally.

A router receiving the first data message finds the outgoing interfaces, through which the message is forwarded, by looking up a unicast routing table with last-hop router addresses in the destination field. Then, the router groups the addresses of last-hop routers by outgoing interfaces and for each interface, the data message is forwarded only with the last-hop routers pertaining to the outgoing interface. Also, the multicast tree information of the CAMS flow is maintained in a soft state manner. From then on, the following data messages do not need to include the addresses of the last-hop routers due to the multicast tree information. Overall, owing to our top-down manner, a multicast tree is built while the first data message is forwarded, which reduces latency and control packets.

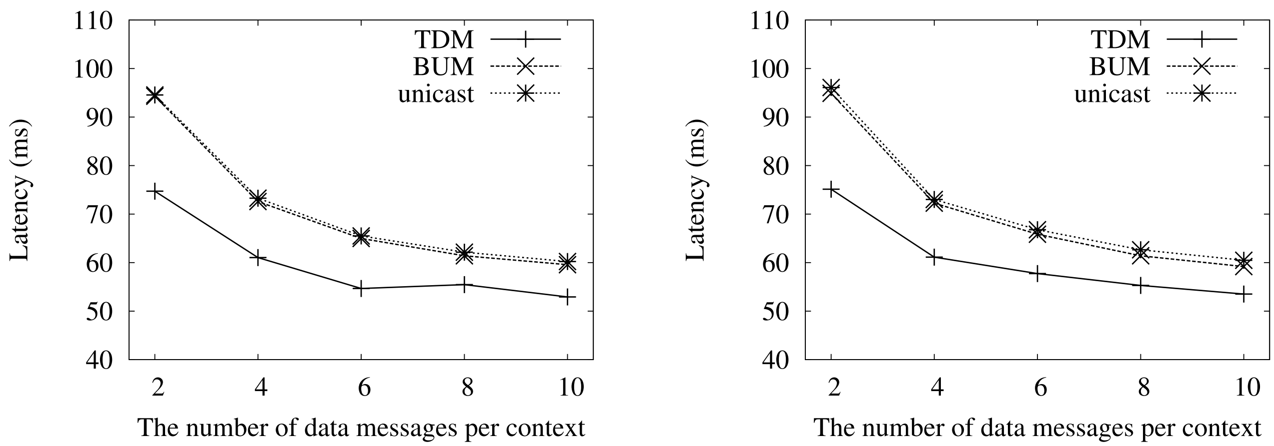
Note that the proposed scheme uses the addresses of last-hop routers instead of the individual target nodes. We believe in many scenarios multiple nodes may belong to the same subnet. Therefore, the number of destinations in the first data message header is reduced significantly. On receipt of each data message, the last-hop router broadcasts it throughout the subnet. Depending on link layer technologies, more efficient multicasting is feasible, which is out of scope in this paper.



(a) Subscriber ratio 0.1

(b) Subscriber ratio 0.3

Fig. 2. The number of control packet forwardings with respect to the number of data messages per context



(a) Subscriber ratio 0.1

(b) Subscriber ratio 0.3

Fig. 3. Latency with respect to the number of data messages per context

III. Performance Evaluation

A. Simulation Setup

In order to evaluate the proposed scheme, we compare it with a unicast-based scheme and a traditional IP multicast-based scheme using computer simulation. We deploy 100 routers in the 10 km $\times$ 10 km simulation area with GT-ITM [5] which models a topology of large internetworks. We set the bandwidth between routers to 100 Mbps and propagation delays to 10 ms. There are 50 subnetworks. The number of clients (hosts) in each subnetwork is uniformly distributed between 10 to 20. We assume shared medium between a router and a client, e.g., wireless medium. As wireless access is more and more widely deployed due to its convenience, this assumption is reasonable in the future Internet environment. Since the bandwidth of wireless access medium is typically lower than wired core medium, we set the bandwidth between a router and a node set to 10 Mbps. The propagation delay is set to one us and the transmission radius is one km.

In our context-aware services, we generate 10 type of contexts. The interval of each context is exponentially distributed with the average 10 seconds. There are 10 CAMS corresponding to each type of contexts. When a context occurs, a part of nodes around the context become subscribers (i.e., a multicast group). The radius of a context is set to two km in our simulation. We varied the subscriber ratio from 0.1 to 0.3. When a context occurs and it is notified to a corresponding CAMS, the CAMS generate several data messages. The detailed parameter is presented in Table I.

We use three metrics to evaluate our scheme: 1) the total number of forwardings, 2) the number of control packet forwardings,and 3) End-to-end latency.

B. Simulation Results

TDM, abbreviation of Top-Down Multicast, represents our proposed mechanism, and BUM, abbreviation of Bottom-Up Multicast, does traditional multicast mechanism using a source-specific tree such as PIM-SSM [6] in Fig. 1, Fig. 2, and Fig. 3.

1) The total number of forwardings: The total number of forwardings of data and control messages is shown in Fig. 1. When the number of data messages per context is low, the number of forwardings in BUM is similar to that in unicast despite multicasting. This is because the number of control message forwardings is very large in BUM. That is, the constant cost of traditional multicast is very large and the cost is compensated when the multicast flow is long-lived and large size. Therefore, traditional multicast is not suitable for short-lived or a few message flows. However, the number of forwardings in TDM is much smaller than unicast even when the number of data messages per context is low. This is because there is no extra control messages to build a multicast tree, which is constructed by delivering the first data message.

2) The number of control packet forwardings: Fig. 2 shows the number of control packet forwardings. As mentioned in Section III-B.1, The number of control packet forwarding in BUM is larger than one in TDM and unicast. In uniast, when a context occurs, each node interested in the context individually requests the corresponding messages to the CAMS and then the CAMS starts to send out messages to the nodes. In BUM, each node also individually requests to the CAMS. Then, the CAMS informs the nodes of the multicast group address piggybacked with first data message. On receipt of the multicast group address, the nodes additionally performs JOIN operation bottom-up and then following data messages are delivered on the constructed multicast tree. Due to the additional JOIN operation, the control packets of BUM is larger than that of unicast. On the other hand, when a context occurs, CS notifies to the appropriate CAMS in terms of the interested nodes and then the CAMS sends out messages using explicit multicast. Overall, the control overhead of TDM is minimal among the three schemes.

3) End-to-end latency: The latency is plotted in Fig. 3. In TDM, interesting clients do not request the appropriate messages individually. Instead, the CS notify the occurring of the context and the relevant last-hop routers to the CAMS. On the other hand, in BUM and unicast, all the interesting clients explicitly solicit the CAMS. Furthermore, in TDM the first message is also delivered in multicast whereas in BUM it is delivered in unicast piggybacked in a control message including a multicast group address (after receiving the message, the client can perform JOIN operation). Therefore, the latency in TDM is smallest. Note that there is no background traffic in our simulation. If the network is congested, multicast-based schemes (i.e., TDM and BUM) have better results than unicast due to efficiency of multicast.

IV. Conclusion and Future Work

In this paper, we proposed an efficient multicasting architecture for context-aware messaging services where multiple nodes subject to a particular context receive several appropriate messages from a particular server. In the proposed architecture, a multicast tree is built without explicit signaling from the clients; instead, we need to deploy a context server. Simulation results show that our mechanism has significantly lower overhead and latency than traditional unicast and multicast mechanisms. In future work, we plan to extend the architecture to support node (client) mobility.

References

[1] Marten J. van Sinderen, Aart T. van halteren, Maarten Wegdam, Hendrik B. Meeuwissen, and E. Henk Eertink, "Supporting Context-Aware Mobile Applications: An Infrastructure Approach," *IEEE Communications Magazine*, vol. 44, no. 9, September 2006.

[2] Sylvia Ratnasamy, Andrey Ermolinskiy and Scott Shenker, "Revisiting IP Multicast," in *Proc. ACM SIGCOMM '06*, 2006.

[3] Demet Aksoy, and Mason Sin-Fai Ieung, "Pull vs Push: A Quantitative Comparison for Data Broadcast," in *Proc. IEEE GLOBECOM '04*, 2004.

[4] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, "Explicit Multicast (Xcast) Basic Specification," *IETF Internet Draft*, January 2003.

[5] "GT-ITM: Geogia Tech Internetwork Topology Models," http://www.cc.gatech.edu/projects/gtitm/, Online link.

[6] H. Holbrook and B. Cain, "Source-Specific Multicast for IP," *IETF Proposed Standard RFC*, RFC 4607, August 2006.