Journal of Network and Computer Applications 50 (2015) 92–100

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A reliable multicast transport protocol for information-centric networks

Charilaos Stais, George Xylomenos*, Alexios Voulimeneas

Mobile Multimedia Laboratory, Athens University of Economics and Business, Athens, Greece

ARTICLE INFO

Article history:
Received 15 December 2013
Received in revised form
23 May 2014
Accepted 30 June 2014
Available online 5 August 2014

Keywords:
Information-centric networks
Error recovery
Multicast
Transport layer

ABSTRACT

In the past few years, many researchers have argued that the Internet should transit from its traditional endpoint-centric architecture to an information-centric paradigm. One of the advantages of the information-centric model is that the network can easily aggregate requests for the same content and serve them via multicast. Indeed, most information-centric architectures proposed to date offer native support for multicast, promising a vast improvement in the efficiency of content distribution. However, designing efficient reliable transport protocols for multicast is a largely open issue, due to the problem of feedback implosion towards the sender as group size grows. In this paper we propose RMTPSI, a retransmission-based reliable error control protocol for multicast communication designed specifically for information-centric networks. We compare RMTPSI with existing approaches proposed for IP multicast and evaluate its performance via simulation, showing that our approach leads to more efficient content distribution and error recovery than previous solutions.

1. Introduction

We have recently experienced a research drive targeting new architectures for the future Internet, aiming to improve the efficiency of large-scale content delivery. Many of these efforts are based on native multicast support, which has long been considered as the key for efficient content distribution, but was never widely adopted on the Internet. The design and implementation of the current Internet architecture leans on the traditional telephone network where there are only two parties wishing to communicate. Extending this model to multiparty communication requires considerable engineering effort and costs for network operators. Unfortunately, there is no clear path to multicast adoption aligned with the business models of network operators (Diot et al., 2000), hence IP multicast is prevalent only inside private networks for specific applications, e.g. IPTV over ADSL networks.

Recent research efforts have tried to move the center of attention from *where* the desired information is located to the *information* itself, since in most applications users are only interested in getting the desired content, not in its location. This direction is evident in *Peer-to-Peer* (P2P) file sharing, *Content Delivery Networks* (CDNs) and cloud computing services, which generally operate as an overlay to the existing Internet. A more radical approach is to introduce a clean-slate *Information-Centric Networking* (ICN) architecture, focusing on content rather than the endpoints hosting and consuming it. By designing a suite of network protocols around information itself, these proposals aim to better satisfy the requirements of current and future content distribution applications (Caesar et al., 2006; Koponen et al., 2007; Jacobson et al., 2009; Parisis et al., 2013).

A claimed advantage of the ICN paradigm is that the network can aggregate requests for the same content and serve them via multicast, thus boosting the efficiency of content delivery. While most ICN architectures offer native support for multicast (Xylomenos et al., 2014), they have not yet addressed the issue of designing efficient reliable transport protocols for multicast, even though considerable work has been performed in this area for IP multicast. In general, reliable multicast can be achieved in two ways: sender-driven with acknowledgments as feedback, and receiver-driven with negative acknowledgments. In sender-driven protocols the sender eventually becomes a bottleneck due to acknowledgment implosion as the number of receivers grows (Pingali et al., 1994). Therefore, most reliable multicast protocols are receiver-driven, an approach that we also adopt. Our work in this area is based on the ICN architecture of the FP7 EU project PURSUIT (PURSUIT Project, 2013), referred to as the *Publish/Subscribe Internet* (PSI) architecture. We have previously briefly presented a receiver-driven reliable multicast protocol for the PSI architecture (Stais et al., 2013). In this paper, we present our

---

* Corresponding author at: Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, Patision 76, GR-10434 Athens, Greece. Tel.: +30 210 8203115

E-mail addresses: stais@aueb.gr (C. Stais), xgeorge@aueb.gr (G. Xylomenos), avoulimeneas@gmail.com (A. Voulimeneas).

http://dx.doi.org/10.1016/j.jnca.2014.06.006
1084-8045/

Reliable Multicast Transport for PSI (RMTPSI) in detail, contrast it with *Pragmatic General Multicast* (PGM) (Gemmell et al., 2003), a reliable multicast transport protocol designed for IP multicast, and evaluate our protocol's performance against PGM over the PSI architecture. Our results show that RMTPSI is more efficient than PGM, while requiring the same time to complete a reliable transfer; specifically, RMTPSI requires 2.9–10.2% fewer downstream and 3.5–12.1% fewer total transmissions than PGM.

The target application for RMTPSI is fully reliable multicast delivery, for example, distributing OS patches or antivirus updates over the network. In these applications each recipient must receive *all* data correctly, regardless of how long this may take. These applications, besides being extremely common, offer a natural synchronization between senders and receivers: as updates become available, they are transmitted immediately to all waiting recipients. In contrast, in applications such as media distribution, it is either hard to ensure receiver synchronization (e.g. in video on demand) or mostly reliable delivery is sufficient (e.g. in live video streaming).

The structure of this paper is as follows. In Section 2 we briefly present the PSI architecture and past work on reliable multicast transport, while in Section 3 we present RMTPSI. Section 4 first provides a description of the experimentation environment and then presents the performance results obtained. We conclude and discuss our plans for future work in Section 5.

2. Background and related work

2.1. The PSI architecture

A publish/subscribe architecture consists of three elements: publishers, subscribers, and an event notification service, also known as a Rendez-Vous network, consisting of *Rendez-Vous Points* (RVPs) (Eugster et al., 2003). The publishers are the content owners who offer their content to potential consumers. To announce content availability, publishers advertize it to the responsible RVP by issuing publication messages. The subscribers are the content consumers who express their interest in specific content items by issuing subscription messages. Information indicating the desired content items is included in the publication and subscription messages.

PSI is an instantiation of such a public/subscribe architecture in a networking context: publishers and subscribers are located at network nodes and exchange data via publish and subscribe primitives which are facilitated by a distributed rendezvous function. Data items are identified by a *Scope Identifier* (SId) and a *RVP Identifier* (RId). The SId identifies a collection of content items and is mapped to the RVP responsible for this particular collection, possibly via a *Distributed Hash Table* (Katsaros et al., 2012). The RId identifies a content item within that collection and is determined by the publishing application. The scoping mechanism in PSI is designed to limit access to content, therefore each scope may have different access control rules (Xylomenos et al., 2012).

A subscriber needs to be aware of the SId/RID pair of a desired content item to issue a subscription message for it. When a subscription message arrives at the RVP corresponding to the SId in the subscription, the RVP checks whether the subscriber can access the scope. If so, it determines which publishers can satisfy the subscriber's request and then communicates with the *Topology Manager* (TM) to request a suitable forwarding path from a publisher to the subscriber. The TM, either a service in the same machine or a stand-alone server, maintains network topology information discovered via a link-state routing protocol. The TM can thus calculate a path between the publisher and the subscriber; when multiple subscribers are interested in the same content item, a multicast tree containing all subscribers is calculated.

The path calculated by the TM is described by a Bloom filter, as in LIPSIN (Jokela et al., 2009). Bloom filters are probabilistic representations of sets where each element is encoded as a string of zeroes and ones, calculated via a set of hash functions. A set is represented as the logical OR of all its elements. In PSI, each link is labeled with one such string in each direction. A Bloom filter in the header of each packet includes the labels of all the links that are part of the desired path. When a packet arrives at a router, the router determines to which of its outgoing links (possibly, more than one) it will have to forward the packet, by performing a logical AND between the label of each link and the in-packet Bloom filter. This technique supports native multicast, since the Bloom filter in the packet header may represent an entire multicast tree; the Bloom filter is simply a set of link labels. Link labels *must* be unidirectional, as otherwise packets would loop, hence the encoded paths, whether unicast or multicast, are also unidirectional.

Figure 1 summarizes the above procedures. First, a publisher issues a publication under a certain SId/RId to the corresponding RVP (step 1). A subscriber that is aware of this SId/RId pair subscribes to it (step 2). After the RVP corresponding to the requested SId receives the subscription, it communicates with the TM in order to retrieve a suitable Bloom filter for data dissemination from the publisher to the subscriber (step 3). Once the RVP gets the Bloom filter, it forwards it to the publisher (step 4). Finally, the data are delivered to the subscriber using the Bloom filter (step 5).

2.2. Wide area multicast in PSI

As more elements are added to a Bloom filter, it becomes more likely that it will match elements not added to it; these are *false positive* matches. When Bloom filters are used to encode routes as in PSI, as more links are added to a route it is more likely that random links may match them. If such a link happens to be on the path taken by a packet, the packet will be needlessly transmitted over that link. The extent of this problem depends on how many links are encoded into the set. In Sarela et al. (2011), the authors argue that the number of ones in the Bloom filter should not exceed 40% of the total bits, meaning that with reasonably sized Bloom filters (as they must fit within packet headers) we cannot represent very large groups or very long paths.

To scale this scheme to larger multicast groups, we employ Bloom filter switching at designated *relay points* (RPs) (Tsilopoulos and Xylomenos, 2013). RPs are routers that replace the Bloom filter inside a packet with a new one before forwarding the packet.

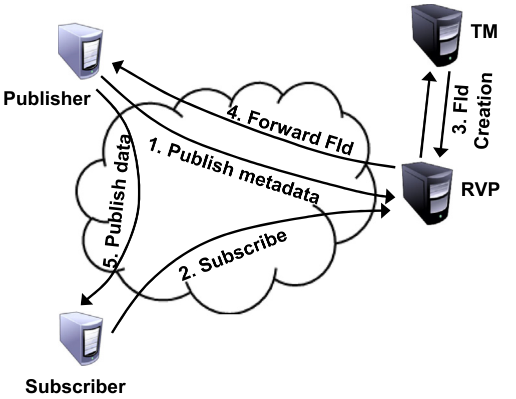

Fig. 1. Communication steps in the PSI architecture.