Computer Communications 70 (2015) 79–85

Contents lists available at ScienceDirect

Computer Communications

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

isBF: Scalable in-packet bloom filter based multicast

Ilya Nikolaevskiy[a,*], Andrey Lukyanenko[a], Tatiana Polishchuk[c], Valentin Polishchuk[c], Andrei Gurtov[a,b,d]

[a] Aalto University, Finland
[b] Helsinki Institute for Information Technology, Finland
[c] Linköping University, Sweden
[d] ITMO University, Russia

ARTICLE INFO

Article history:
Received 11 November 2014
Revised 16 May 2015
Accepted 20 May 2015
Available online 29 May 2015

Keywords:
In-packet bloom filters
Multicast
Internet
Architecture

ABSTRACT

Bloom filter (BF) based forwarding was proposed recently in several protocol alternatives to IP multicast. Some of these protocols avoid the state in intermediate routers and leave the burden of scalability management to the multicast source and end-hosts. Still, the existing BF-based protocols have scalability limitations and require explicit network management as well as non-trivial functionality from the network components. In this work we address the scalability limitations of the BF-based forwarding protocols by partitioning end-hosts into clusters. We propose several algorithms to do the partitioning so as to decrease the overall traffic in the network. We evaluate our algorithms in a real Internet topology, demonstrating the ability of the proposed design to save up to 70% of traffic volume in the large-scale topology for big groups of subscribers, and up to 30% for small groups.

1. Introduction

Today's Internet contains large volumes of network traffic with one common pattern: almost each piece of the data is interesting for multiple recipients (for example, BitTorrent, YouTube, IPTV traffic, etc). The efficiency is extremely hard to achieve in current networks. A few successful examples of traffic savings for networking are web caches and commercial CDNs [11]. There are even attempts to extend CDN publicly to a peer-to-peer (P2P) environment [6]. The main limitation of CDN approach is commercial-primary deployment unacceptable to private users.

Another method for traffic savings is multicast [4] where the network helps to create a minimal amount of traffic until some point in which the traffic can be cloned to multiple copies, and each copy forwarded to interested recipients. As opposed to CDN architecture, multicast benefits small multicast groups. IP multicast, however, still failed to be widely deployed. There are several reasons for that. First, with the current IP network, routers require state for each subscribing group in each router. As the number of possible subscribers may be more than millions and each subscriber has multiple groups in active use, a middle router can easily become overcrowded with heavy lookups over big tables and memory usage. The latter is a severe problem for core routers. Second, multicast deployment requires provider incentives. Fortunately, there is still positive evidence that hardware manufactures are willing to put most of the multicast protocols into routers [18].

Some researchers suggest clean-slate redesign of the current Internet, where multicast is considered as one of the most important network properties [13]. An elegant solution, firstly proposed in [18] as part of a control plane, was adopted in the form of *in-packet Bloom filters* (*iBF*) as the data plane solution [10]. iBF introduces multicast opportunities without the active router's state; however, iBF still has scalability limitations.

We emphasize that scalability, economical considerations and the complexity of network components create barriers for wide multicast deployment, independently of the architecture. In this work we extend iBF based forwarding for networks with global flat forwarding fabric deployed around the Internet. We propose new mechanisms for scalability and management by introducing a novel *in-packet scalable Bloom filters* (*isBF*) protocol. We attempt to avoid network component complexity by utilizing existing IP fields for our purposes.

The contributions of this work are twofold. First, we design a number of new algorithms based on flat BF labeling, which are able to scale to the whole Internet. Next, we evaluate our algorithms in a real Internet topology (CAIDA data set).

The rest of the paper is organized as follows. In Section 2 we describe the related work and Bloom filters basics. We propose four

---

[*] Corresponding author. Tel.: +358505758927.
E-mail addresses: ilya.nikolaevskiy@aalto.fi (I. Nikolaevskiy), andrey.lukyanenko@aalto.fi (A. Lukyanenko), tatiana.polishchuk@hiit.fi (T. Polishchuk), valentin.polishchuk@helsinki.fi (V. Polishchuk), gurtov@hiit.fi (A. Gurtov).

http://dx.doi.org/10.1016/j.comcom.2015.05.002
0140-3664/

Table 1
Comparison of multicast alternatives.

	Forwarding	Structure	Scalability in the number of groups	Scalability in the number of subscribers	Additional network elements	Traffic economy
IP multicast (ideal)	IP	Flat	No	Yes	No	Perfect
LIPSIN	iBF	Hierarchical	Yes	n/a	Yes (RVZ)	n/a
isBF	iBF+IP	Flat	Yes	Yes	No	Average

novel algorithms to achieve scalability in Section 3. Section 4 presents evaluation of the proposed algorithms. In Section 5 we discuss economic incentives to utilize multicast. Section 6 concludes the paper.

2. Background and related work

Over two decades ago, Deering and Cheriton [3,4] proposed IP multicast as an efficient solution for data dissemination from a single source to multiple receivers. It was deployed experimentally [16], but never adopted in any significant way by service providers. The failure of multicast [5] to achieve the wide spread adoption can be explained by several technical and economic factors, including complexity of the multicast network management and uncertainty in how to appropriately charge for the service in the case when sources and receivers belong to different domains. For many years multicast was implemented only locally within the service providers' networks supporting IPTV[21] and conferencing applications, and also in enterprise networks [12], where the aforementioned issues of management, billing and inter-provider dependencies are mitigated.

Due to its unquestionable capability to significantly reduce network load, multicast remains the most studied research problem in computer networking [17]. According to work [10] the main challenge in efficient *information centric network* (*ICN*) design is how to build a multicast infrastructure that could scale to the general Internet and tolerate its failure modes while achieving both low latency and efficient use of resources. In topic-based ICNs, the number of topics is large, while each topic may have only a few receivers [15]. IP multicast and application level multicast have scalability and efficiency limitations under such conditions. In IP multicast, the amount of routing state is proportional to the number of multicast groups. To address this issue, several multicast proposals [8,10,18,23] implemented the idea of using *Bloom filters* (*BF*) [2] in the packet headers. This way the intermediate routers are purged from the burden of keeping states.

The authors of LIPSIN [10] proposed the multicast fabric, which uses the iBF directly for the forwarding decisions, removing the need for IP-addresses and proposing *Link IDs* (*LIDs*) as a generic indirection primitive. To implement full scale routing in the Internet, authors propose to use two levels of forwarding fabrics - intra-domain and inter-domain. Each AS should have a dedicated node which works as a gateway between intra-domain and inter-domain routing. Those nodes are called rendezvous (RVZ).

The iBF based forwarding fabric was utilized in several works afterwards. Some works are focused on datacenter networks [19]. But they are not extendable to internet-size topologies. Another work [14] utilizes Bloom filters on switch interfaces to encode all the multicast groups on the interface. However, this approach is also limited in scalability. Jokela et al. [9] utilized iBF to reduce multicast state, however this work considers only small number of clients in each multicast group and is not scalable to Internet-size networks.

Tapolcai et al. [22] propose same stateless design as we propose in our work. They utilize multi-stage iBF to encode several levels of data-dissemination tree. However, this solution requires variable size headers in packets. For internet-wide multicast that solution requires headers as long as 2KB. That greatly limits it's applicability for a wide-scale multicast.

Heszberger et al. [7] also suggest to utilize iBF for multicast addressing. To limit size of Bloom filters they propose novel adaptive length Bloom filters. However, their solution does not scale up to whole Internet either.

The packet header with iBFs creates false positives, which give rise to forwarding anomalies. Successful methods for their reduction are discussed by Sarela et al. [20], although they do not solve scalability problem in terms of the number of recipients.

2.1. Comparison to our approach

In Table 1 we compare our isBF protocol with two alternatives. While IP multicast and LIPSIN use stateful routers in the middle, they obviously achieve the most traffic reduction in the network. For LIPSIN this is due to the additional hardware components which should be deployed widely. However, when the number of RVZ servers increases (proportionally to the number of active ASes), the solution stops being scalable in terms of the number of subscribers. Consequently, some additional measures will be needed to cover all recipients with one BF, either a third layer of hierarchy (which will increase the per-packet overhead) or some smart splitting methods. The benefit of our approach is that we use algorithms on flat labels with minimal per-packet overhead, and our protocol can be easily extended with in-network RVZ servers (treated as clients). It is possible to add RVZ servers anywhere in the network for our protocol. RVZ server will be a gateway for all clients in the corresponding part of the network. RVZ in turn will work as a single client for any multicast server or other RVZ. With such RVZ servers our protocol comes closer to the ideal multicast performance. This is not possible for LIPSIN.

2.2. Bloom filter basics

A Bloom filter (BF) [1] for encoding a set X is a probabilistic data structure for membership queries in X. The filter consists of k independent hash functions $h_1, \ldots, h_k$ and a binary array of length m. Each function maps any element x that may belong to X to an integer chosen uniformly at random from $1 \ldots m$. To add an element x to the set, it is needed to get k array positions using the hash functions, and set bits in the array at these positions to 1. To test whether x is in the set, feed x to each of the k functions to get k array positions. If any of the bits at these positions is 0, the element is definitely not in the set; if all are 1, then either the element is in the set, or the bits have by chance been set to 1 during insertion of other elements, resulting in a *false positive*.

An iBF encodes all links which packet should traverse to reach all clients. The network routers simply forward the packet into all the local links which are encoded in the iBF. Obviously, if we set all bits in iBF (m bits set), then such a packet will be broadcasted to the whole network. For practical and security reasons it is advisable to keep the maximum *fill factor* (the ratio of bits set to m) in BFs below 50% [20].

3. Scalability of iBF

In large multicast networks the following issue arises: any reasonably large set of far situated destinations produces a data delivery tree in which the LIDs are encoded in an iBF almost completely filled with 1*s*. Such a BF exceeds the fill factor threshold and will encode all the links in the network causing undesirable packet flooding. One