Contributions to The Resilience of Peer-to-Peer Video Streaming Against Denial-of-Service Attacks

by

Nguyen Truong Giang

Department of Computer Science
TU Dresden

Ph.D. Dissertation

2016

Thesis summary

Introduction

Due to the growing user demand to watch live events, such as music festivals, football matches, and TV news etc., video streaming over the Internet has become a popular service. Video streaming offers the capability to start playing a video as soon as a small portion of the video is successfully downloaded. Thus, the waiting time is reduced. Statistics show that the traffic from video streaming has increases drastically and this trend is projected to continue in the coming years [3]. To satisfy this demand, several architectures have been proposed for video streaming over the Internet: unicast, IP multicast, Content Delivery Network (CDN) and Peer-to-Peer (P2P) as illustrated in Figure 1.

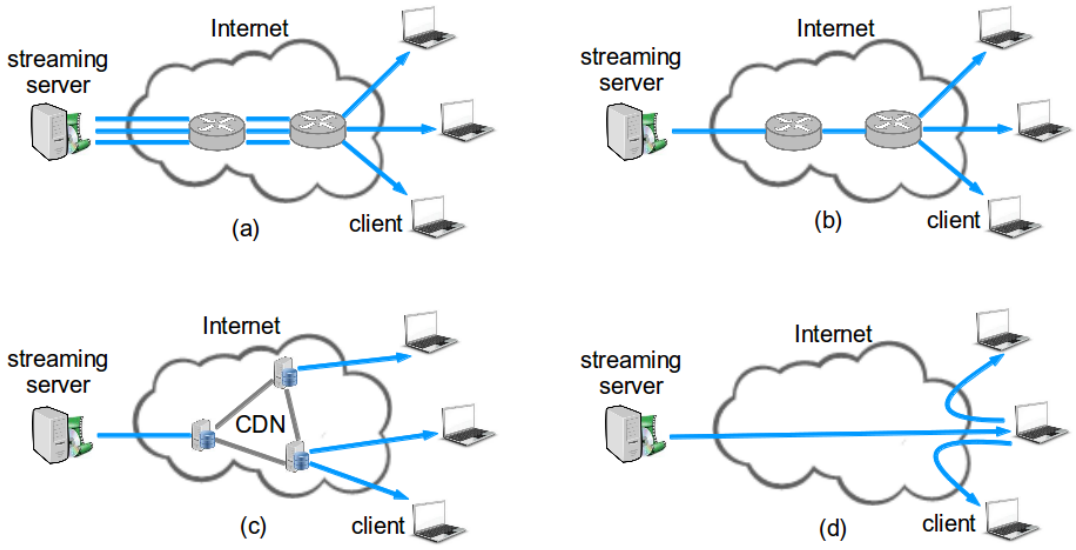

FIGURE 1: Four architectures of video streaming over the Internet: (a) Unicast; (b) IP multicast; (c) Content Delivery Network (CDN); and (d) Peer-to-Peer (P2P).

Video streaming architectures

In a straightforward approach, the unicast architecture [1] shown in Figure 1(a) requires the streaming server to deliver the video directly to its users. To do that, every user has to establish a unicast connection to the server. As long as the server still has enough resources, mainly its upload bandwidth, every newly arriving user can receive the video stream with a low latency. However, the bandwidth demand to the server increases linearly with the number of users. This generates a huge demand not only on the streaming server but also on the Internet's core network. The network has to simultaneously deliver several times of the same stream to users.

To resolve the bandwidth demand to both the streaming server and the core network, the IP multicast architecture [10] is developed. As can be seen from Figure 1(b), the server sends only one stream on the shared paths of the connections between the server and its users. The stream is then replicated at the routers where the paths to different users diverge. Therefore, the video traffic duplication is minimized. Additionally, the delay is also minimized since the video stream is replicated very early at the IP layer without being passed through upper ones. In spite of those huge advantages, IP multicast has two major drawbacks. First, it requires routers to maintain a state for each multicast group, which complicates the management and limits the scaling at the IP layer. Second, the architecture requires modification at the infrastructure level, thus slowing down the deployment [4]. So far, the IP multicast architecture only appears in restricted deployments (e.g., at IPTV backbones) [7].

The Content Delivery Network (CDN) architecture, illustrated in Figure 1(c), addresses the deployment problem of IP multicast by introducing a dedicated network consisting of several replication servers [8]. They are equipped with abundant bandwidth and installed strategically in high traffic regions. Therefore, the video source needs to send the stream to the CDN only once. The stream is then distributed within the CDN to its replication servers and duplicated there to serve nearby users via unicast connections. In addition to a drastic reduction of the bandwidth demand to the streaming server, CDN offers several advantages, such as high capacity and high performance. However, CDN also has several drawbacks, which are high costs and the poor support to live streaming.

The Peer-to-Peer (P2P) architecture, illustrated in Figure 1(d), resolves the high-cost problem of a CDN by requiring users (or peers) to contribute their resource, mainly the upload bandwidth, to disseminate video streams. Participating peers not only receive but also send video streams to others. Since P2P streaming only requires a modest upload bandwidth from the streaming server (or source), it reduces the bandwidth demand on the source. Additionally, the architecture also enables scaling up the system since peers participate with their own resources. The more peers join the system, the more resources the system has. Despite the above advantages,

the P2P architecture poses higher latency than its alternatives since a peer generally receives the stream after it is forwarded via several other peers. The general challenge for P2P streaming is building an overlay interconnecting the source and peers and deciding a method to disseminate video streams so that the overall system maintains a low latency and a low signaling overhead. Since ordinary peers can join and leave the system freely at any time, churn is inevitable. Building an overlay robust to churn is a must to maintain the service without disruptions. Furthermore, the system has to provide attack resistance. To damage the system, a relevant subset of peers can be shut down to disrupt the video dissemination to the rest of the peers. Such sabotage can be performed by an individual hacker simply for fame or by an organization or a government to block the delivery of a certain content. Over the past years, various P2P streaming systems have been proposed. They can be categorized into three classes: push-based, pull-based and hybrid ones as shown in Figure 2.

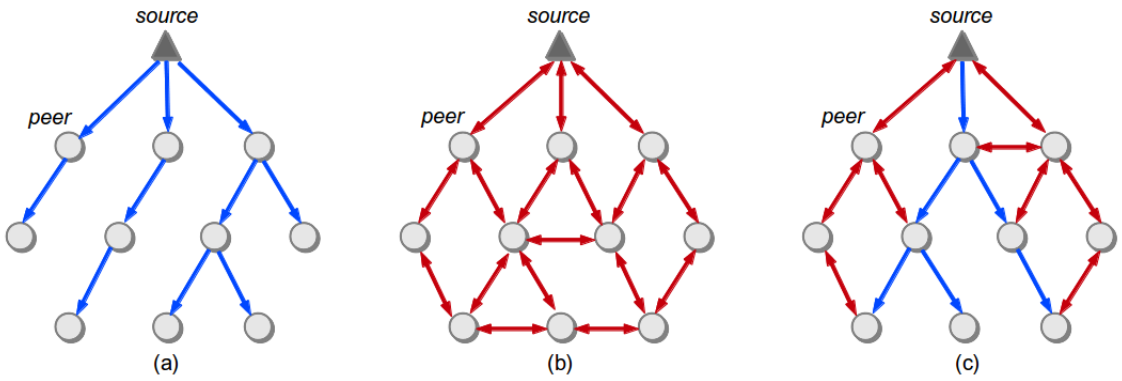

FIGURE 2: Three classes of P2P streaming systems: (a) push-based; (b) pull-based; and (c) hybrid push-pull. The links between nodes depict the overlay connections while the arrowheads indicate the directions of sent video chunks.

P2P video streaming

As illustrated in Figure 2(a), an early push-based system builds a tree rooting at the source and spanning all the peers. Along the tree's branches, the video chunks of the stream are sent (or pushed) from the source and immediately forwarded by intermediate peers until they reach peers at the leaves of the tree. This way, the system achieves a low latency. The single-tree topology, however, is fragile to high churn and especially to attacks on peers close to the source since each peer solely depends on its predecessors. To improve the robustness to churn, later push-based systems maintain multiple-tree topologies providing each peer several direct predecessors to improve the connectivity. Moreover, each tree disseminates video chunks of a partial stream. Even when a predecessor leaves, its successors only lose a partial stream and still receive the rest from their remaining predecessors. Furthermore, some of

the multiple-tree push-based systems are even theoretically resistant to a perfect attacker [2]. However, those superior properties have not been proven in large-scale real-world deployments.

Also tackling the issue of churn, pull-based systems follow a very different approach. A pull-based system strengthens the system's connectivity by constructing a mesh topology interconnecting the source and its peers with one another as depicted in Figure 2(b). Thus, there are several source-to-peer paths for each peer preventing that the departure of a single node leads to the node's isolation and hence its inability to receive the stream. To obtain the video stream, every peer has to explicitly and regularly request video chunks from others. This reactive behavior also helps peers adapt quickly to their neighboring peers' churn. However, the trade-off is that peers and the source have to periodically send buffer maps (BMs), small packets carrying the availability information of peers' video buffers. The combination of frequently exchanging BMs and explicit requests causes high latency and high signaling overhead. Nevertheless, the pull-based class of systems has been adopted widely in many real-world P2P streaming systems [5].

To combine the advantages of both push-based and pull-based systems, a hybrid push-pull (or hybrid) system bootstraps from a pull-based one to leverage its robustness to churn [9]. Subsequently, a subset of peers starts to push video chunks directly to their neighboring peers without requests to reduce the latency and overhead. The process of pushing video chunks begins from the source and gradually to other peers to form a backbone. To strengthen the backbone, it should only consist of stable peers meaning that they already stay sufficiently long in the system. A measurement study shows that such a backbone delivers a majority of video chunks in the system [9]. At the same time, peers in the backbone still keep their existing mesh connections with neighboring peers for two reasons. First, backbone peers can send requested chunks to other peers to reduce the overall latency. Second, when a backbone peer loses the pushed stream from a predecessor due to churn, the peer still can request missing chunks from neighboring peers, which increases the robustness. In other words, the pull-based system is not only a bootstrap but also a Figure 2(c).

Despite the robustness to churn, pull-based systems and their derivatives, hybrid ones, can be vulnerable to Denial-of-Service (DoS) attacks on head nodes, which are the source's partners. Those nodes can be identified by inferring the information in exchanged buffer map. The inference of the overlay structure including head nodes from collected buffer maps has been experimented on PPLive, a pull-based system [6]. It is however unknown whether the inference of head nodes is feasible on pull-based systems in general. However, if an inference attacker can successfully infer the head nodes and subsequently shut them down, the remaining peers are disconnected from the source, thus, disrupting the video stream delivery. Hybrid systems, bootstrapping from a pull-based one are also vulnerable to the inference and therefore are vulnerable to the attacks. Furthermore, the backbone topology of

a hybrid system is a single-tree that is fragile to attacks. Disrupting this backbone can drastically affect not only backbone peers but also the video stream delivery to the rest of the peers. So far, these problems have not been addressed.

In summary, pull-based systems with their proven robustness to churn are potential candidates for P2P video streaming over the Internet.

However, two major problems need to be solved:

(*i*) The resilience of pull-based systems to attacks on head nodes

(*ii*) The resilience of hybrid push-pull system to attacks on the backbone.

Problem description

This thesis needs to improve the resilience of pull-based and subsequently hybrid P2P streaming systems to DoS attacks. For this reason, pull-based systems need to improve their resilience to attacks on head nodes. On top of that, hybrid systems need to build a resilient backbone to counter DoS attacks.

Goals of the study

This thesis sets several goals as follows:

1. A set of metrics needs to be defined to quantify the steady-state performance of P2P streaming systems in benign scenarios and especially to quantify the resilience of the systems under attacks.

2. To thoroughly investigate and fairly compare different classes of P2P streaming systems, a multi-purpose simulation framework needs to be developed.

3. To mitigate the attacks on head nodes, the thesis needs to study the impact of the attacks on pull-based P2P streaming systems and subsequently develop countermeasures against such attacks.

4. To maintain a low latency and a low signaling overhead and at the same time to mitigate the attacks the backbone of a hybrid systems, we needs to develop a countermeasure against such attacks.

Contributions of this thesis

This thesis provides the following four contributions.

A general-purpose simulation framework: The evaluation of different schemes in P2P streaming requires a common framework that allows for a fair comparison. In addition, the framework should be generic and extensible to support the implementation of various P2P streaming systems. Since such a framework has not been available so far, we develop OSSim, our general-purpose simulation framework for P2P video streaming. The framework allows for simulating all classes of P2P streaming systems, namely push-based, pull-based and hybrid. To validate the framework we implement one representative system for each class of P2P streaming systems. The simulation results from our framework match closely with published results. In addition to that, we develop and implement several novel resilience metrics and attacker models. Consequently, we use OSSim as our main tool to evaluate the proposed countermeasures against DoS attacks in P2P streaming systems.

Resilience against DoS attacks on head nodes: Even though pull-based systems are vulnerable to DoS attacks on head nodes. Shutting down those nodes can disrupt the stream delivery from the source to the rest of the peers. We develop the striping scheme to mitigate damages caused by such attacks even if the attacker has a global knowledge. Our approach is to increase the number of partners each peer has, consequently increasing the number of head nodes. Therefore, the system has more chance to maintain connections between the source and the peers under attacks, thus, reducing the negative impacts of the attacks. However, we also need to prevent peers from overloading themselves by chunk requests from their enlarged partner lists. We solve this problem in our striping scheme by enforcing peers to diversify their chunk requests. To do so, we introduce two techniques: First, the video stream is divided into k stripes, which are partial streams. Second, each peer assigns its partners into k groups. Consequently, in each scheduling cycle, every peer has to request chunks of a particular stripe from the respective group of partners only.

Resilience against inference attacks: In pull-based systems the overlay structure, especially head node can be identified by inferring the exchanged buffer maps. Despite mitigating the damages caused by the attacks on head nodes, the striping scheme does not prevent the identification of head nodes. For this reason, we develop SWAP as a countermeasure to the inference attacker that collects buffer maps to infer head nodes. Our idea is to enforce peers to regularly and proactively change their partners. As the result, the attacker is unable to accurately identify head nodes any more. Consequently, we introduce the following three primitives: First, to prepare for the swapping operations, every peer should suggest its replacement candidate. Furthermore, the nominations should be forwarded several times, allowing peers to connect to more diverse new partners. Second, to prevent the nomination message

from infinitely forwarded, we introduce a counter. Third, to be ready for attacks, peers periodically replace a subset of their partners by the respectively nominated ones.

Resilience against DoS attacks on the backbone: After strengthening the resilience of pull-based systems to DoS attacks, we address the vulnerability of hybrid systems. Specifically, to protect hybrid systems from the attacks on the backbone, we develop the Resilient Backbone Construction Scheme (RBCS). The backbone only includes stable peers that stay sufficiently long in the system. The scheme is built from three primitives. First, peers only use their local observation to identify stable peers that stay sufficiently long in the system. Identified stable peers are invited to join the backbone. Using invitations, the system does not reveal peers in the backbone, thus, lowers the risk of attacks on the backbone. Second, the backbone employs the Optimally Stable Topology (OST) to include peers into a multiple-tree overlay. OST has theoretically proven its resilience to DoS attacks, thus, strengthening the backbone against attacks. Third, RBCS introduces an adaptive bandwidth reservation scheme to flexibly coordinate the bandwidth of peers when they push chunks in the backbone and respond to chunk requests at the same time.

Summary

In summary, this thesis addresses the problems of Denial-of-Service (DoS) attacks in P2P video streaming systems. Specifically, the DoS attacks on head nodes of pull-based systems and DoS attacks on the backbone of hybrid systems are addressed. Consequently, three schemes have been developed to solve those problems. First, we develop a striping scheme to counter the DoS attacks on head nodes of pull-based systems. Second, we develop a SWAP scheme to undermine the capability of an attacker to detect head nodes of pull-based systems. Third, our RBCS scheme has been developed to counter DoS attacks on the backbone of a hybrid system. To fairly evaluate and compare the systems, we develop OSSim, a general-purpose simulation framework for P2P video streaming. Extensive simulation studies show that our developed schemes significantly mitigate the damages caused the DoS attacks at the cost of a slight increase in signaling overhead.

Bibliography

[1] Aurrecoechea, C., Campbell, A.T., Hauw, L.: A survey of qos architectures. Multimedia Systems 6(3), 138–151 (May 1998)

[2] Brinkmeier, M., Fischer, M., Grau, S., Schäfer, G., Strufe, T.: Methods for improving resilience in communication networks and p2p overlays. PIK. Praxis der Informationsverarbeitung und Kommunikation 32, 64–78 (2008)

[3] Cisco: Cisco vni global ip traffic forecast, 2014–2019 (May 2015), `http://www.cisco.com`

[4] Diot, C., Levine, B., Lyles, B., Kassem, H., Balensiefen, D.: Deployment issues for the ip multicast service and architecture. IEEE Network 14(1), 78–88 (January 2000)

[5] Gu, Y., Zong, N., Zhang, Y., Piccol, F., Duan, S.: Survey of p2p streaming applications (October 2014)

[6] Hei, X., Liu, Y., Ross, K.: Inferring network-wide quality in p2p live streaming systems. IEEE Journal on Selected Areas in Communications 25(9), 1640–1654 (December 2007)

[7] Minoli, D.: IP Multicast with Applications to IPTV and Mobile DVB-H. Wiley-IEEE Press (2008)

[8] Vakali, A., Pallis, G.: Content delivery networks: status and trends. IEEE Internet Computing 7(6), 68–74 (November 2003)

[9] Wang, F., Xiong, Y., Liu, J.: mtreebone: A collaborative tree-mesh overlay network for multicast video streaming. IEEE Transactions on Parallel and Distributed Systems 21(3), 379–392 (March 2010)

[10] Williamson, B.: Developing IP Multicast Networks. Cisco Press (1999)