Distributed Secure Monitoring in Large Scale Overlay Networks

Andreas Disterhöft
Technology of Social Networks, University of Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
Email: distehoeft@cs.uni-duesseldorf.de

Kalman Graffi
Technology of Social Networks, University of Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
Email: graffi@cs.uni-duesseldorf.de

Abstract—Controlling the Quality of Service of peer-to-peer networks may open doors to new quality-centric overlays and application fields, such as peer-to-peer-based online social networks, telemedicine or robot coordination. To reach this aim, a sophisticated, timely and broad view on the current status of the network, the nodes and the content in the network is needed, in order to come to optimized decisions in the various protocols operating in the peer-to-peer system. The main aim of our research topic is to research monitoring mechanisms which provide sophisticated information on the network’s status, the nodes and the content. Our mechanisms are evaluated within large-scale simulations as well as in peer-to-peer applications tested with real users. For the latter, we developed a Web-based peer-to-peer framework which does not require the users to install any software installation to join. It supports a simple secure online social network and is capable to support large-scale networks. This research statement contains aims and challenges of our work.

I. INTRODUCTION

Nowadays peer-to-peer overlay networks are well understood and offer dedicated benefits in contrast to a client/server architecture. These benefits are due to its decentralized nature and mainly disallow censorship, they are not deflectable and in some cases they are boosting the efficiency of the system by using the peer’s resources. The downside of the decentralized nature is the lack of an adjustment unit which keeps track of the overall network performance. Preconfigured settings cannot adapt to the quality needs and current network properties, as being unaware of them. In order to provide a certain quality of service level, e.g. through continuous adjusting of the local peer parameters, such as the routing table size, information about the network is needed. In general, information about the network enables to initiate correcting actions, information about nodes enables to select ideal nodes based on their capacities and information on data elements enables to adjust storage mechanisms to better fit the needs of the users.

Our work focuses on ways to effectively monitor p2p networks to provide future quality-centric overlays with a wide set of information, which can be used to optimally select parameter settings and strategies. Resulting mechanism also will be applied and evaluated within a web-based p2p framework previously proposed by the authors. This p2p framework is called WebP2P and is briefly presented in Section III. The next Section II summarily presents our aims for our research topic and its difficulties we have to challenge.

II. AIMS AND CHALLENGES

In this chapter we briefly present our aims on this research topic. We enumerate five subtopics, where each refers to a separate section in the wide area of p2p monitoring. In the first part we research robust and precise system monitoring methods which overcome current p2p overlay restrictions and combine existing monitoring systems. In the second part we detect and provide counter-measurements for malicious nodes in monitoring systems. System measurement methods with incomplete participation, where only a subset of peers in a p2p network is joining the monitoring system, are our aim in the third part. In part four we improve node capacity indexing and searching in overlay networks to provide a capacity search in multiple dimensions. The last aim is to supply document statistics and ranking mechanisms in overlay networks.

A. Robust and precise system monitoring methods

Current monitoring solutions are mainly either gossip-based or tree-based. SkyEye.KOM [1] for example constructs a tree to collect and distribute aggregated information. To achieve this, a tree overlay is built on top of the existing p2p overlay, where each node calculates its position and determines its role in this tree. This approach requires restrictive assumptions about the underlying p2p overlay making it vulnerable against churn and thus lowering the monitoring precision.

We research a self-adaptive and self-organizing tree-based monitoring system on top of a self-organizing protocol like Care-Chord [2]. A self-stabilizing solution will reach a desired network state from any initial connected network state. With this in mind, churn will be no longer a problem for the monitoring structure. The challenge here is to find a self-organizing tree structure to use it in our monitoring system.

Another approach we follow is to combine tree-based and gossip-based monitoring solutions. In gossip-based monitoring solutions, peers communicate with current neighbors to share their locally aggregated data in order to approximate the global view. By merging gossip-based and tree-based monitoring systems, we may profit of the stable but inaccurate gossip and the churn-vulnerable but accurate tree-based monitoring system. To achieve this, we synchronize the protocols to overcome the epoch-based calculations in the gossip-based solutions and capably introduce gossip connections in the tree to provide stability to our tree-based system.

B. Secure and manipulation-resistant measurement methods

To detect manipulations in tree-based monitoring systems, first we evaluate the how malicious nodes can manipulate data and how honest nodes can discover attacks. If malicious behavior has been detected, we provide a mechanism to reconstruct
data. Therefore we construct reconstruction data in a proactive way by utilizing our knowledge about the deterministic tree construction and using parallel, redundant aggregation paths. For the verification, we check data passed from nodes we are not sure whether they are malicious by consulting their source (e.g. a grand-child). With a scoring system we rate nodes, which forward data, in a scale and upon reaching a certain level we may decrease the verification rate for this node. If a node detects a potential malicious node by verifying its data, it may start to reconstruct this broken data set. The reconstruction may be invoked by contacting the forwarding node’s predecessor. As this approach does not scale a challenge is to evaluate proper data reconstruction.

C. System measurement methods with incomplete participation

We evaluate how to monitor a network with an incomplete participating node set. Considering that only a subset of peers might be expected to participate voluntarily in the monitoring of the p2p network, we evaluate the options for this limited (active) peer set. A first approach will be to discover active nodes, which are willing to participate in the monitoring. These nodes may build up an overlay maintaining passive nodes, which are probed in order to get required information. Afterwards we plan to use the monitoring system, which we mentioned in Section II-A, to start the monitoring. A challenge we may encounter are interdependencies between the various overlays, as we may use three overlays for this solution.

D. Indexing and retrieval of node capacity information

Within this subtopic aim at peer-capacity-based search instead of a document-based search in the p2p network. The challenge is to index dynamic content, here the capacity of peers, which changes frequently in contrast to static documents and also nodes may leave. Existing solutions like Adaptive Chord [3] need one overlay for each capacity, which may produce a huge overhead when using multiple capacities. Our solutions will use either a multi-dimensional space with fixed capacity classes or a heap-based index structure to realize a tree-based structure. In the first approach, we aim to use for every capacity a dedicated dimension and assign the nodes to coordinates in this multi-dimensional space based on their current capacities. In the second approach, we use a heap-structure to sort nodes by their various capacities and to find through a bottom-up search suitable peers with desired capacities. However making nodes searchable in a p2p overlay will provide new use cases for p2p networks like distributed computation or make p2p overlays more efficient by using node’s capacities for its structure.

E. Distributed measurement of document statistics in overlays

In the last part of our research project we refer to the creation of document-based statistics and rankings. First, we aim to support document statistics for every document in the p2p overlay. These statistics have to be publicly available, so that every peer is able to retrieve document-based statistics like popularity rank, related traffic or access count. To achieve this we plan to evaluate two different approaches: memory- and publish/subscribe-based.

The memory-based approach stores document statistics under a well-defined place, where nodes can access and modify it. On the other side, the publish/subscribe-based approach will use a multicast tree per document, where users interested on particular document statistics can invoke or receive statistic updates. In addition to these document statistics we aim to support rankings in the elicitation of document statistics. So it will be possible to identify the document, which caused the k-most traffic or which is k-most popular. The architecture of this solution is challenging as statistics completeness, low load and timeliness are hard to achieve all together. One severe challenge is defined by churn as a node is responsible for a document and its right position in the list: wild fluctuations may be possible. We aim to implement and maintain an efficient monitoring structure for the ranking and the statistics of the documents.

III. PORTING TO A REAL-WORLD SCENARIO

All previously described mechanisms will be implemented in a p2p system simulator, such as PeerfactSim.KOM [4]. It allows to evaluate the monitoring mechanisms on various overlay networks with large-scale networks. In addition, to see these mechanisms in a real-world scenario we plan to implement selected mechanisms within our WebP2P framework and to test it with real users.

In its current state our WebP2P framework builds up a secured web-based Chord network via WebRTC using browser instances. We use OpenChord, which we have heavily modified and extended to support p2p-based networks with a Public Key infrastructure, creating the opportunity to communicate secured and verified, different DHT entries to enable low level access control and a modified join procedure to support a login. Our software’s main advantage is the use of a p2p network without any requirements for installation, as participants only need an updated browser. On top of this framework we implemented an online social application providing a contact list management, contact search and secured text-based and audio / video chat. Further p2p mechanisms and applications can be added easily. The graphical user interface uses up-to-date HTML5 for presentation.

In future, our application may include a social computation, where user may pool their storage and computation power e.g. for social games. For this Section II-D’s mechanism may help choosing a contact, strong enough to handle the requested calculation. Another use case may be document statistics and ranking in potential shared file systems, where users may use Section II-E’s mechanism to access rankings about e.g. the most popular and most viewed document.

REFERENCES


