1 Introduction

What TV is extremely good at—and realize that this is "all it does"—is discerning what large numbers of people think they want, and supplying it.

David Foster Wallace

Nowadays, we are observing the convergence of classical content distribution forms, like TV and the Internet, towards Internet Protocol Television (IPTV) systems. The resulting systems distribute a large number of different streams, including a significant number of real-time streams, to an audience of assumably very heterogeneous clients. Streams are split into several substreams (e.g., by MDC coding [Goy01]), so-called stripes, and are distributed independently of each other. Thus, users can subscribe to arbitrary subsets of the provided streams and may subscribe to several of them in parallel, e.g., multi-view. Furthermore, per stream a user can decide to subscribe to all stripes or only to a subset of them, whereas the best quality is provided by a subscription to all stripes of the stream. Subscribing to a subset only causes a quality degradation of the respective stream, but allocates less bandwidth. Additional stripes may provide extra content, like additional audio tracks or extra media objects to be placed in the video (e.g., a sign language translator instead of the newscaster). This results in arbitrarily structured user demands.

1.1 Problem Statement

In order to distribute such IPTV content, following a client-server approach is not advisable, e.g., taking into account that Cisco predicts [Cis10] IPTV content to hold 91% of the global data traffic by 2015. Network multicast with source-based, shortest path trees is the most efficient form of content distribution and is approximated by IP Multicast. However, IP Multicast still is not deployed on a large-scale and has drawbacks regarding security and scalability [DLL+00].

Hence, Application Layer Multicast (ALM) as an efficient and scalable form of content distribution has been proposed for distributing data in large-scale IPTV systems [CRSZ02]. ALM incorporates the participants' resources into the distribution process so that each user who receives a stream forwards it as well. ALM systems are usually classified into pull-based, push-based and hybrid systems [FD07]. As this thesis focuses on IPTV that serves as a replacement for conventional TV, mainly push-based ALM is considered as it is the most promising approach for live-streaming.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

However, ALM systems need to incorporate potentially unreliable or even malicious end-systems into the distribution process, opening up numerous opportunities for potential Denial of Service (DoS) attacks. Furthermore, as Peer-to-Peer (P2P) networks, ALM overlays are established on top of an underlying infrastructure network. The failure of a link or router in the underlay (e.g., as a result of direct attacks) may affect multiple overlay links concurrently [BFGS09b], so that high overlay damage can result.

Most current ALM systems are highly vulnerable to external attacks on end-nodes and provide nearly no resistance to malicious nodes, which can easily gather topology knowledge to prepare subsequent attacks. Thus, the main focus of this thesis is on increasing the attack resilience of ALM overlays against external attacks on end-nodes that are guided by participating malicious nodes and on increasing the resilience against attacks on underlay components at the same time. Maintaining efficient topologies to provide low end-user delay and to decrease the overall traffic load in core networks is the subject of this thesis as well. Therefore, current research efforts in the direction of a future Internet have been considered, like virtual routers, e.g., [ZA06, WKB+08], and pushing additional functionality to core networks, e.g., [Jan02, ZBF+08].

To sum up, this thesis attempts to answer the following questions:

- *By which mechanisms and to what extent is it possible to limit the damage from external attacks by providing graceful degradation properties?*
- *Which topology construction mechanisms and countermeasures limit the effectiveness of internal attackers?*
- *How can underlay-aware topologies be constructed in a distributed manner, so that the overlay dependency on single underlay components is limited and thus the resilience against attacks on underlay components increases?*
- *Which benefits in terms of core network traffic savings is provided by virtual routers assisting in IPTV stream distribution and what are appropriate methods of adapting these virtual routers to a changing user compound?*

The next section summarizes the answers to the questions provided by the thesis.

1.2 Contributions of this Thesis

This thesis was established in the context of two DFG projects, namely “DosResistantStreaming” and “DosResistantIPTV”. These projects were conducted together with the Automata and Formal Languages group at TU Ilmenau. The author of this thesis worked on developing heuristics and performing prototypic evaluation of protocols, whereas the Automata and Formal Languages Group worked more on the algorithmic aspects, like finding optimal solutions and determining the difficulty of the arising problems. Both groups worked closely together on the analysis of the problems as well as on the design of suitable heuristics and protocol mechanisms throughout these two projects, and thus results have been published jointly.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

The contributions of this thesis are described in the paragraphs given below.

Overlay Resilience against Attacks on End-Nodes Most of the current ALM systems are highly vulnerable to attacks on end-nodes. Brinkmeier et al. [BSS09] have introduced optimal stable push-based ALM topologies for a single stream scenario. They provide graceful degradation in the presence of attacks on the overlay and provide minimum upper bounds for the achievable damage to them. Moreover, in [BSS09] a small subclass of optimal stable topologies has been identified that can be constructed efficiently with global knowledge. Furthermore, a distributed construction heuristic in [Str07] has approximated these topologies.

In the context of this thesis, a prototype of the system in [Str07, BSS09] has been implemented and the results of the simulative evaluation from [Str07, BSS09] have been reproduced. Thus, it has been shown that optimal stable single-stream topologies can be approximated in real-world scenarios as well. In addition, in joint publications with the AFS group, the known subclass of optimal stable topologies could be extended significantly and their important properties could be identified [BBG+09, GFBS11, GFS11].

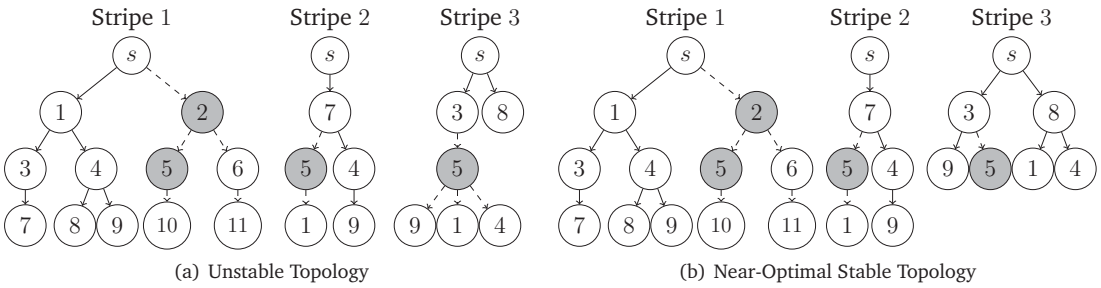

Figure 1.1: (a) An unstable topology with three stripes and (b) a near-optimal stable one with the same parameters. Attacking nodes $\{2, 5\}$ leads to a damage of eleven disrupted source-peer paths for the unstable and eight for the near-optimal stable topology.

Unfortunately, the results from [Str07, BSS09] cannot be applied directly to IPTV and a multi-stream scenario as optimal stability is based on the presumption that all users subscribe to all stripes of a single stream. Thus, due to the arbitrarily structured user demands of IPTV systems, the problem of finding optimal stable topologies becomes more complex. For this reason, this thesis addresses the problem of establishing sufficiently stable IPTV topologies against attacks on end-nodes in a distributed manner. Therefore, the concept of optimal stability [BSS09] has been transfered to an IPTV scenario [FGKS11] with multiple-sources and in which not every user subscribes to all stripes. Hence, the strict concept of optimal stability has been relaxed to near-optimal stability and a heuristic for the efficient distributed construction of corresponding topologies has been found [FGKS11]. Figure 1.1(a) shows an example of an unstable and Figure 1.1(b) a near-optimal stable topology. Besides an increased resilience against attacks, near-optimal stable topologies are also less vulnerable to random failures, which are more common in ALM streaming.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Manipulation-Resistant Topology Construction ALM systems that rely on end-users in content distribution are vulnerable to internal attacks by malicious nodes. The analysis of the current state of the art in the context of this work revealed that most ALM approaches are highly susceptible by internal attackers as they do not restrict them in their topology knowledge. Thus, another major goal of this thesis is the development of manipulation-resistant topology construction mechanisms that restrict the knowledge of internal attackers and the damage of subsequent DoS attacks on disclosed streaming nodes. These mechanisms have been partially published in [BFGS09a]. Furthermore, they have been evaluated against a sophisticated DoS attacker that deploys malicious nodes in the topology to disclose relevant targets for DoS attacks from the outside. Extensive simulations indicate that the proposed construction mechanisms and novel countermeasures limit the damage of such an attacker to one of an instantly attacking adversary that does not presume long-term cooperation of malicious nodes.

Overlay Resilience against Underlay Attacks After establishing resilient topologies against attacks on end-nodes in a manipulation-resistant manner, the arising topologies are still vulnerable to attacks on components in the transport network (routers and links). Most ALM approaches incorporate only delay measurements between member nodes, e.g., [Str07, BSS09]. However, this does not properly reflect topological network properties. Only Topology-Aware Grouping (TAG) [KF05] attempts to include the topology of the underlying transport network by path measurements gained via traceroutes. Nevertheless, these topologies do not exploit the bandwidth capacities of end-users and so induce larger delays on the end-user side. Furthermore, as TAG focuses on efficiency only, its overlay topologies provide nearly no resistance to attacks on overlay nodes and on underlay components.

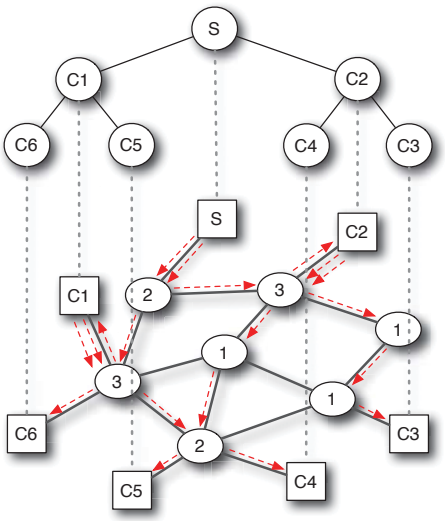

Figure 1.2: Embedding of an ALM overlay to an underlying infrastructure network. Numbers in oval routers indicate the number of traversing overlay links. Dashed arrows indicate overlay links traversing underlay links.

For this reason, this thesis focuses on minimizing the overlay dependency on single underlay components in order to limit the overlay damage coming from attacks on

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

underlay components. An example is given in Figure 1.2, which shows the underlay-aware embedding of such a streaming topology rooted at source s and spanning members $c_1, \cdots, c_6$. Thus, a topology construction on the basis of complete underlay path information has been developed [FDGS11] which fulfills this property. In addition to an increased resilience against underlay attacks, this approach also induces an efficiency gain. It reduces the traffic load in core networks considerably compared to a conventional ALM approach, e.g., an approach that includes only delay information between nodes to the overlay construction.

Increasing the Efficiency of Content Distribution via Virtual Routers Inspired by the current research towards a future Internet and from peer-assisted ALM streaming approaches, this thesis attempts to increase the efficiency of the content distribution by making use of large server bandwidth combined with forwarding by end-users. For example the approach given in [Jan02] attempts to reproduce IP Multicast functionality at application layer by allowing end-hosts to assign forwarding tasks to routers within the network. However, this introduces new forms of DoS attacks and results in significant signalling load. Nevertheless, the idea of supporting content distribution from network routers appears promising, and so it has been adopted for this thesis. Hence, routers that take over a more active role in forwarding stream data assist the IPTV system from within the network. These ALM routers have been introduced in [FKGS11]. They are flexible virtual components that are placed on top of core network routers and run the same ALM algorithm as end-users to redistribute content. They are capable of adapting to a changing user compound by relocating themselves between neighboring routers based on a strictly local metric and without external intervention. Simulation results indicate that they significantly reduce the traffic load in core networks in dependence on their contributed distribution capacity.

Summary This thesis addresses several problems. First, topologies which are sufficiently stable against attacks on end-nodes are established. Second, the construction of these topologies is modified so that they are established in a manipulation-resistant manner even in the presence of malicious nodes. Third, the ALM approach is designed to limit the overlay dependency on single underlay components and so decreases the damage resulting from failures in the transport network or from attacks on underlay components. Lastly, in order to decrease the traffic load in core networks, the IPTV system is supported from within the network by flexibly positionable virtual routers.

For the evaluation of the developed approaches, a comprehensive simulation framework has been developed that allows for a distributed topology construction as well as for sophisticated attack mechanisms to test these topologies in a dynamic environment. Furthermore, several attacker models and metrics have been developed to evaluate the different aspects of the proposed system. For example, this includes a sophisticated attacker model for a realistic DoS attacker that is supported by malicious nodes participating in the system. In addition, novel metrics for the evaluation of overlay resilience and the underlay dependency of the established overlay networks are introduced. A combined simulative evaluation of all proposed single mechanisms indicate the suitability of the proposed solutions in an IPTV scenario.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

1.3 Structure of this Thesis

The rest of this thesis is structured as follows: Chapter 2 contains an overview about multimedia content distribution and techniques for multimedia coding. In addition, this chapter presents a classification of P2P systems and summarizes general attacks on these systems as well as possible countermeasures. Furthermore, the simulation tool used for the evaluation of the developed mechanisms is described.

Chapter 3 summarizes the requirements for a P2P-based IPTV system. On that basis, the current state of the art in content distribution to large user populations is analyzed. This includes IP Multicast, Content Distribution Networks (CDNs) and several ALM approaches. Furthermore, a closer look is taken at topology-aware P2P overlays with respect to the given requirements.

As a consequence of the drawbacks identified in the state of the art, Chapter 4 introduces a self-designed approach, which is called Attack-Resilient and Efficient ALM-based IPTV (AREA IPTV). Based on a formal model, several mechanisms are developed to construct overlay topologies that are resilient to attacks on end-nodes and underlay components.

Furthermore, manipulation-resistant topology construction mechanisms are presented. They enable a construction of ALM topologies in the presence of malicious nodes and by incorporating mobile users and nodes that are diverse in their bandwidth equipment. Moreover, methods to increase the efficiency of the established distribution topologies are covered as well.

Chapter 5 contains a detailed discussion of AREA IPTV with respect to the requirements from Chapter 3 and summarizes research questions and novel metrics for their evaluation. The remainder of the chapter evaluates the different single mechanisms of AREA IPTV both separately and in combination. A summary of the obtained results is given at the end of the chapter.

Chapter 6 concludes this thesis with a short summary and an outlook for future work in the field of P2P-based IPTV.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

2 Background

Who controls the past controls the future. Who controls the present controls the past.

George Orwell (1984)

This chapter provides essential background information for this thesis. Section 2.1 sketches the basics of multimedia content distribution and provides a classification of multimedia content distribution. Section 2.2 classifies P2P systems and summarizes possible attacks and countermeasures against them. In Section 2.3 the foundations of Discrete Event Simulation (DES) are sketched and OMNeT++ as the simulation tool used for the evaluation of this thesis is described.

2.1 Multimedia Content Distribution

The following explains the challenges in multimedia streaming and provides a classification. Afterwards, an overview of multimedia coding techniques is given and popular approaches are described, compared to each other, and checked for their application to multimedia streaming and in particular to P2P live-streaming.

2.1.1 Challenges and Classification of Multimedia Streaming

A stream is multimedia content that is received continuously and that can be played back immediately without storing the whole content locally in advance. However, the Internet has not been designed with continuous and time-based traffic in mind. Thus, streaming multimedia data via the Internet faces several challenges.

First, the transmission is usually *time critical* with graduations depending on the specific application. The playback of multimedia data requires a continuous stream of packets with no or only limited capabilities of re-requesting data in case of *transmission failures* and thus lost packets. Every packet that arrives after its planned playback time is worthless and can be dropped. Thus, the Transmission Control Protocol (TCP) is not applicable for the transport of multimedia data, especially not with strict timing constraints and a small buffer. Hence, for the transport of time-critical multimedia data, mostly the User Datagram Protocol (UDP) and additionally the Real-Time Transport Protocol (RTP) are used, which has been specifically designed for streaming data.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Furthermore, since network conditions may change and subsequent packets may take different routes through the network, the differences in the variance of the end-to-end delay of streaming packets, also known as *jitter*, can be great. Packets may even overtake each other and arrive in the wrong order. For this reason, a reordering of packets is required at the receiver's side.

Multimedia Content Distribution can be classified into three different classes. First, *Live-streaming* is the form with the strictest requirements regarding the time criticalness and the least chance that lost packets can be re-requested in time. Live-events are encoded on the fly and have to be distributed with the least possible delay. Therefore, all clients are loosely in synchronization and thus receive the same content nearly simultaneously. A related class is *Near-Video-on-Demand*, which summarizes streams that start periodically in predefined time intervals. However, since the content to be distributed is usually not new, their timing constraints are less strict than for live-streams. The third and probably most popular class is *Video on Demand (VoD)*. Clients joining the stream usually start obtaining the stream from the beginning. For this reason, all the clients can have different playback times. Timing constraints are less strict than for live-streaming and near-video-on-demand, so that clients can buffer large parts of the stream in advance. As a result, packet loss can be tolerated and most lost packets can be easily re-requested prior to the playback deadline. Another difference of VoD from live- and near-video-on-demand streams is that users can influence the playback of the stream. They can pause the playback or can move backwards or forwards in the stream.

2.1.2 Multimedia Coding Techniques

The encoding of multimedia data is a large field of research. For this reason, the following gives only a basic overview of the most popular coding techniques for transmissions across data networks.

Forward Error Correction (FEC)

Forward Error Correction (FEC) coding attempts to reduce the error rate when transmitting or storing data. Therefore, redundancy is applied that enables to detect a limited number of errors and allows to correct them to a certain extent. FEC approaches are usually classified into *block* and *convolutional* codes [Bha83]. As the name implies, block codes operate on fixed sized blocks of bits or symbols and can be decoded in polynomial time to their final block length. For this, usually per symbol a hard decision is made about its value, which is in contrast to the soft decisions enabled in convolutional codes. Reed-Solomon codes are the most popular block codes so far since they can be found on the Compact Disc (CD), the Digital Versatile Disc (DVD), or on hard disks.

Convolutional codes operate on bit or symbol streams of arbitrary length. The actual decoding in most convolutional approaches is based on the Viterbi algorithm [Vit67], which is a maximum likelihood decoder. Thus, instead of applying hard decisions per symbol, soft decisions can be made. Its complexity increases exponentially with

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

the constraint length of the code, but the larger this constraint length, the better the code and the larger the coding gains. The drawback of Viterbi decoding is its poor performance in bursty channels. This can be partially handled via the interleaving of data, but results in increased delay that may not be appropriate for all possible applications.

Novel forms of FEC coding techniques are Low-Density Parity-check Codes (LDPCs) that were originally invented in 1963 [Gal63], but had been forgotten for some time. LDPCs appeared again with the advent of turbo codes [BGT93] that were originally published in 1993. Both coding techniques allow approaching the theoretical limits of the channel capacity given a specific noise level. LDPCs are block codes, whereas turbo codes in the meantime appear in different forms like convolutional, block, or hybrid codes as a combination of convolutional and block coding techniques.

Since convolutional codes require data interleaving techniques for bursty channels and thus enlarge the delay, block codes should be preferred over convolutional ones. Hence, especially LDPCs seems to be a promising coding technique for multimedia data.

Scalable Video Coding (SVC)

Scalable Video Coding (SVC) is an extension to the *H.264/MPEG-4 AVC* video compression standard. An overview about the basic concepts of SVC is given by Schwarz et al. [SMW07]. A SVC stream consists of one or more subset bitstreams. Such a subset stream is gained by dropping packets from the larger original stream so that the bandwidth for the subset stream can be reduced.

SVC supports four different scalability modes:

- *Temporal Scalability* applies to the frame rate of the stream. Data is arranged in a way that allows dropping complete frames/pictures from the original bitstream. Starting from the base layer, the first k layers belong together and can be decoded independently from all layers above. Thus, removing all layers greater than k still allows decoding the stream and this for all k in the range from one to the highest layer number of the respective stream.
- *Spatial Scalability* is related to the screen size that has to be supported. Data of lower resolutions can be directly used to predict data of higher resolutions so that the bitrates for coding higher resolutions can be reduced.
- *Quality Scalability* maintains a single spatial resolution but at different qualities. Data of lower qualities is used to predict data of higher qualities.
- *Combined Scalability* represents a combination of the three aforementioned SVC scalability modes.

However, loosing a layer of a stream makes all subsequent layers useless so that the loss of the base layer results in the loss of the complete stream. For this reason, several enhancements to SVC were proposed that make use of either Multiple Description Coding (MDC) [vdSR01] or FEC [CH03] methods.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Multiple Description Coding (MDC)

Multiple Description Coding (MDC) [Goy01] divides the stream into multiple substreams, so-called stripes or descriptions, as is the case with SVC as well. However, the difference is that each stripe is independent of the other ones. Thus, packet loss in one stripe has no influence on other stripes of the respective stream. MDC comes in various forms and is still an active field of research.

In the simplest form, data is split at the source and each set is compressed independently in order to create one description per set. For example, two descriptions can be generated by separating a stream of data into odd and even packets. However, this MDC form relies entirely on the redundancy that is present in the source. This redundancy can be correlation or memory.

In *Progressive Source Coding*, only a certain fraction of the preferably most important data is repeated across different descriptions. The rest of the data is simply split between the different descriptions without applying any redundancy. Thus, this approach is also called *unequal error protection* since not all bits are protected equally.

MD Quantization is another method that allows assigning a relative importance to central descriptions and each side description.

During *MD Correlation Transformations*, the statistical dependencies between transform coefficients in between descriptions are increased. As a result, on the basis of the received transmissions, lost descriptions can be more effectively estimated.

A technique with the same purpose is the *MD Coding with Frames*, which is similar to a block channel code and allows for a better estimation of lost descriptions.

Additionally, MDC techniques can be found in combination with SVC, e.g., [CW03]. For example, data is encoded once for low-bandwidth clients and once for high-bandwidth clients and both codes share the same base layer.

The main advantage of MDC compared to FEC lies in smaller block sizes and thus a smaller transmission delay. Furthermore, MDC can easily be applied to multimedia streaming and can be found in various ALM streaming systems, e.g., [CDK+03, BSS09, XLKZ07].

Summary

Regarding the resilience to packet loss, FEC coding seems to be the best alternative for multimedia streaming. However, FEC introduces additional redundancy and thus enlarges the data to be transmitted and consequently the delay as well. For this reason, FEC is not directly applicable to a live-streaming context.

SVC divides the stream into several subset streams so that different forms of scalability can be achieved, ranging from scaling to higher resolutions (and screens) to varying frame rates. However, when a subset bitstream is lost all subsequent layers are useless since every layer depends on all previous layers. Thus, this mechanism may not be suitable for P2P multimedia streaming in which the probability of the loss of subset bitstreams is high.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

Similar to SVC, MDC divides a stream into several substreams, the difference being that each substream is independent of the other. As a result, packet loss in one stripe does not affect other stripes, and so, depending on the specific encoding of the stream, packet loss can be tolerated. Hence, MDC seems to be the best alternative for multimedia streaming. Furthermore, it is possible to combine FEC and MDC methods. Transferring this to P2P-based streaming enables the successors of a failed node to reconstruct the lost stripe from the other stripes they receive so that packet loss can be prevented to a certain extent.

2.2 P2P Systems

Peer-to-Peer (P2P) systems gained a lot of attention in the past. Popular applications like Napster, Gnutella, BitTorrent, PPLive are founded on the P2P principle. The following presents a classification of P2P approaches. Furthermore, attacks on P2P systems are summarized with particular emphasis on attacks on P2P-based multimedia streaming and potential countermeasures are given.

2.2.1 Classification of P2P systems

P2P systems are usually deployed on the application layer as an additional overlay on top of a common underlying infrastructure network. Thus, these networks are independent of the structure of the underlay. A peer can be connected with any other peer. The main principles are self-organization and decentralization so that a Single-Point-of-Failure (SPoF) is prevented. All peers are equal and are both client and server at the same time so that they can consume services as well as offer them. Therefore, each peer provides its resources to the system, like computational power, bandwidth, or memory.

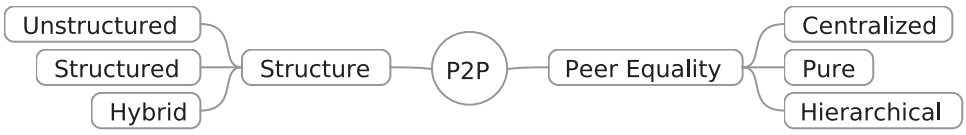

Figure 2.1: Classification of P2P approaches.

P2P overlays can be roughly classified into unstructured, structured and hybrid approaches [LCP+05] as given in Figure 2.1 on the left. In unstructured P2P systems, peers are loosely coupled among each other. For this reason, the overlay structure itself does not carry any information for finding services or content. Thus, search operations have to be carried out by flooding the network or by using gossiping approaches. For this reason, unstructured approaches are suited well for overlays in which content is highly replicated and perform poor when rare content needs to be found. A popular unstructured P2P system is the *Gnutella* network.

In contrast, structured P2P overlays contain (as the name implies) an inherent structure regarding the connections among member nodes. Hence, this structure can be used to deploy content or information at strategic positions in the overlay so that it

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

can be found efficiently afterwards. Most structured P2P approaches make use of a Distributed Hash Table (DHT) data structure like in *Chord* [SMLN+03] or *Pastry* [RD01]. Overlay nodes are assigned an ID and the overlay is structured accordingly. Data is mapped to keys so that each data key can be mapped to the ID of a designated peer. However, complex queries can be achieved only with high effort. Moreover, the node to which a key of a data object is mapped has to store either the data object itself, a copy of the object, or at least the node has to maintain a pointer to the data object. The most popular structured system that is based upon a DHT is the BitTorrent file-sharing system. Most recent structured approaches rely on *Skip Graphs* [JRS+09], e.g., *SkipNet* [HJS03]. Instead of organizing peers and data according to hashes as in DHTs, they are organized by their string names. With it, SkipNet is able to maintain locality properties in addition to the usual DHT functionality.

Moreover, hybrid designs exist that combine unstructured techniques, like flooding, to find popular and highly replicated items with structured techniques, like DHTs, to locate rare content.

Furthermore, P2P overlays can be classified into *centralized, pure and hierarchical P2P* approaches as shown in Figure 2.1 on the right. In the earliest form, centralized approaches emerged. In such systems, centralized components take over active tasks within the P2P overlay, which can comprise bootstrapping tasks and content or peer indexing functions. For example the *Napster* system, as one of the first popular file-sharing systems, utilized central indexing servers that tracked the available content at participating members. Moreover, these indexing servers took over a broker function to bring clients together. Then the content itself was transferred directly from peer to peer. Pure P2P systems completely relinquish centralized components so that the whole overlay consists of equal peers only. Hybrid, or also called hierarchical P2P approaches, combine centralized and pure P2P by selecting peers that take over a more active role in the overlay. In some approaches, these peers are also called super-nodes, e.g., in ChordNet [SP11].

2.2.2 Attacks on P2P Systems

P2P systems usually have an in-built resilience to peer failures, as they need to rely on potentially unreliable end-systems. Hence, single peers are also more vulnerable to attacks than dedicated servers in a client-server-based system, for example. However, due to their inherent failure tolerance, attacks on the service availability in P2P systems are much harder to conduct than in client-server-based systems.

The following section offers a classification of attacks on P2P systems. On this basis, specific threats to P2P-based multimedia streaming are discussed in more detail and thereupon possible countermeasures are summarized. This section provides an overview and classification of attacks on P2P streaming.

DoS Attacks

Denial of Service (DoS) attacks are usually classified into *resource destruction* and *resource exhaustion* attacks [Mil92]. Most direct attack forms can be mapped to this

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

categorization. Resource destruction attacks are carried out with the goal of destroying a service completely. For example, they can be carried out by actively exploiting weaknesses in the targets' client software or by physical attacks (e.g., routers, cables, servers) to switch them off. Additionally, the deviation from a compulsory protocol operating sequence can be seen as a resource destruction attack as well as this can lead to a crash of the attacked system. However, this largely depends on the operating system, software, and the safeguarding of systems, and so it has only minor relevance to this work. Resource exhaustion attacks attempt to deplete the resources of a target system, be it memory, bandwidth, or CPU time. Hence, the attacker allocates as many resources as possible in the course of an attack so that legitimate requests cannot be processed anymore. Especially bandwidth-depleting DoS attacks have gained much attention over the last years.

Attacker Scope

There are two basic attacker types in a P2P system, the *external* and the *internal* one. An *external attacker* can observe and attack the system from the outside only. Thus, it may issue attacks on participating nodes without being sanctioned by the system. In general, it is assumed that an external attacker can take down any peer node by issuing a DoS attack against it or triggering some other effect to remove the node from the system. The main limitation of an external attacker is that it has to gather information by eavesdropping on communication between nodes, which can be restricted by cryptographic protocols.

Unless an external attacker can gather a lot of information about the system, an *internal* attacker may be more powerful, especially if combined with an external component. These *sleeper attacks*, with additional capabilities, cover a wide range of possible attack scenarios and many attack forms can be attributed to them.

The attacker controls a set of sleepers, or *agents*, which participate in the system and may support each other, e.g., by recommending other agents for more valuable tasks or assigning each other high reputation values (when a reputation system is used). In addition, they may communicate with a central authority guiding the attack. The agents join the system in the same way ordinary nodes do and start participating. As a result, they can easily use all procedures the system provides.

One important parameter of sleeper attacks is the *degree of cooperation* between agents. The agents may be completely independent of each other, preventing coordinated attacks – except when a global synchronized clock is used. Or they may communicate via an external authority allowing coordinated attacks and the simultaneous collection of information from all agents.

Preliminary Attacks

In the context of P2P networks, a number of preliminary attacks are known, which are carried out in advance of subsequent direct attacks. These attacks are employed to either identify valuable attack targets, to attract traffic (in order to inspect it or

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

to forward it selectively), or simply to determine a good moment to launch a direct attack, e.g., a Distributed Denial of Service (DDoS).

The most popular preliminary method is the *Sybil* attack [Dou02], in which a single physical node gains a multitude of diverse logical identities in a P2P network. This provides an attacker with the ability to easily deploy several malicious nodes (agents) in a P2P overlay, i.e., to gain more knowledge about other participating nodes. Thus, it can use this knowledge to prepare subsequent attacks, like a DoS attack, on some of the nodes disclosed to the adversary. It also enables the attacker to have the complete knowledge of every agent at all times.

The *eclipse attack* [SCDR04] can be executed subsequent to a Sybil attack, as it requires a huge amount of controlled identities in a P2P network. With it, the attacker attempts to enclose a victim node by only its controlled nodes in order to launch subsequent attacks more effectively, e.g., to selectively forward information to the victim.

Attackers within the system can attempt to lie about their own performance and thus deviate from the original operating sequence to gain an advantage in the respective P2P protocol. For example, a P2P system that incorporates a network coordinate system (e.g., Vivaldi [DCKM04]) for considering the delay of overlay connections in the overlay construction would allow attracting additional traffic by announcing spoofed distances to other peers in the overlay.

Attacks on Multimedia Streaming

Direct attacks on ALM-based multimedia streaming require knowing some of the participating nodes or even the source in advance. A direct attack on the source in single-source ALM would disrupt the streaming immediately and represents the worst-case attack on such a system. For this reason, it is assumed that the source is non-attackable and only attacks on member nodes are carried out.

In order to identify targets to attack, agents need to infiltrate the ALM system first. These agents may be guided by an external component that executes direct DoS attacks on nodes in the streaming overlay. Figure 2.2 shows a rough classification summarizing the options of an (internal) attacker in ALM systems, which will be explained in detail in the following.

In addition to simply cooperating with the streaming system, agents may pursue their task of reaching relevant positions more *actively*. For example, in surprisingly many streaming systems (see also Section 3.4), peers can promote themselves to more important positions and are often even rewarded for taking on more responsibility with the goal of inhibiting free riders.

A second way for agents to gain importance is to lie about their performance, making them seemingly better than other peers [JCY+06]. This misinformation may lead to a 'false promotion' since other nodes assume them to be the best possible choice for an important position in the P2P network. Besides, they may try to attract peers to increase their number of successors and thus their own relevance.

Third, they can become aggressive in advance by issuing attacks on other peers in their surroundings, provoking an at least local reconstruction of the topology, which

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

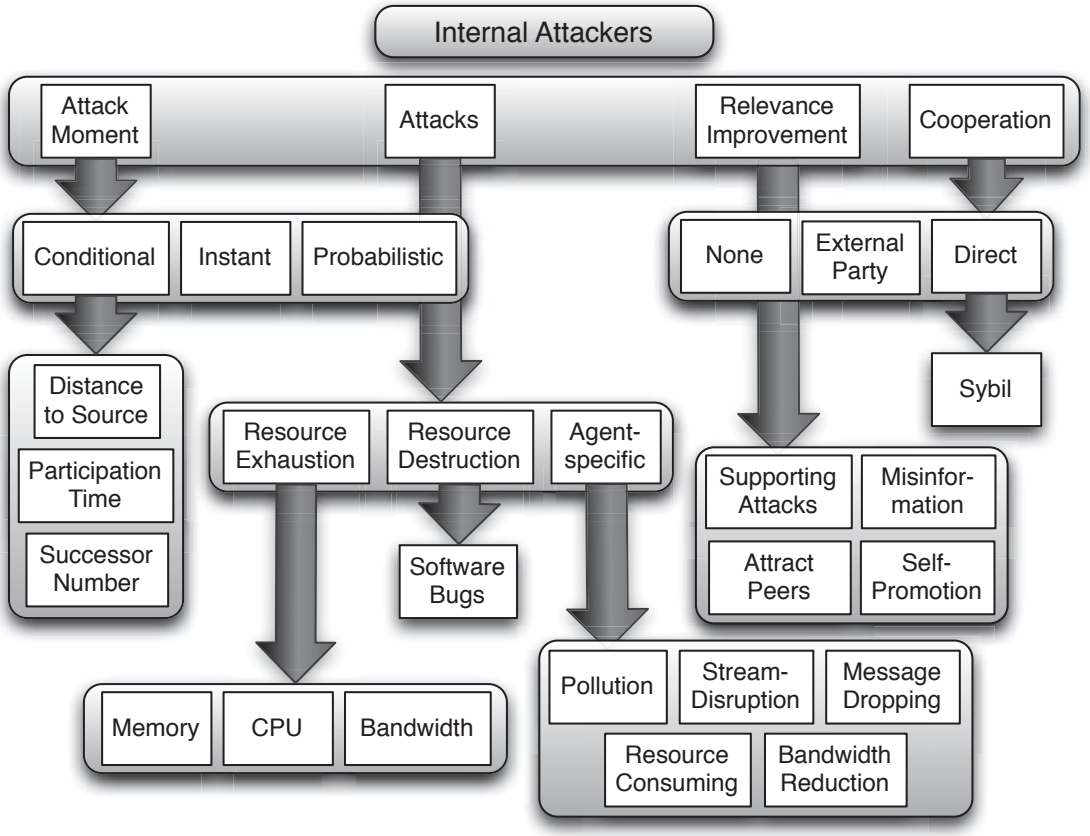

Figure 2.2: Classification of internal attackers in an ALM system.

provides them with the chance of reaching a better topological position. These supporting attacks might be conducted by an external assistant, which prevents the discovery of the agents and possible sanctions, like their exclusion.

Another way of gaining importance is by carrying out an eclipse attack [SCDR04], i.e., by making sure that the attacker's agents are predecessors of the target nodes on all possible distribution paths. This allows subsequent coordinated attacks on target nodes, like disrupting them from streaming completely.

In a completely *passive* sleeper attack, agents simply wait until a certain condition is met before triggering an attack. This *attack moment* may be chosen in many ways. Either agents attempt to reach certain relevant positions in the topology or they just collect information and try to identify important nodes, or even the possibly concealed source itself. The second aim is basically equivalent to the first since usually a node that reaches a specific position can identify the source. Hence, it can be assumed that an attack is triggered as soon as the positions of the agents satisfy specific conditions.

The decision whether the current positions of agents are important or not mainly depends on the signaling procedure of the particular ALM system. If the system provides its participants with information on the number of nodes depending on their service (i.e., their number of successors) or even the IDs of these nodes, the attacker can calculate the exact damage it can cause at any given time. Hence, it may trigger the attack as soon as a given objective is reached. In other situations, the position of a node may correlate with the time spent in the system. In these cases, the attacker

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.

may simply wait for a specific time and then has a chance that his agents may cause a given damage.

If the attacker decides or assumes that its agents have reached good positions, it can trigger a direct attack, which can either target at resource destruction or resource exhaustion. One way to disrupt the service is a coordinated DoS attack on nodes that the agents previously have identified. In addition, agents in a streaming system enable special forms of DoS attacks. This might be a simple disruption caused by all agents leaving the system at the same time. Optionally, the agents may not leave the system but reduce their bandwidth so that the Quality of Service (QoS) for their successors decreases significantly. A related form is the message dropping attack [XZ06], in which participating agents drop certain stream packets.

Furthermore, agents can provoke damage by a *resource-consuming attack* [JCY+06] that induces a bandwidth depletion at targets by connecting a large number of agents to the same node so that it is blocked from forwarding the stream to regular participants. Alternatively, agents may start to *pollute* the system by inserting corrupted data into the stream. Successors may detect the pollution (e.g., by verifying cryptographic signatures on the stream) and try to reconnect to the system, possibly leading to a massive reconstruction and a decrease in QoS. This *Pollution attack* [DHRS07] has the same effect as agent failures or attacks on specific nodes.

2.2.3 Countermeasures against Attacks on P2P Streaming

Countermeasures can be divided into proactive and reactive ones. Reactive approaches trigger an action after a failure has occurred, whereas proactive approaches attempt to take action before a failure occurs. For this reason, proactive approaches are usually faster in error recovery but require more effort than reactive ones.

Proactive Approaches

The pull-based P2P streaming approach SecureStream [HvR06] includes proactive countermeasures against internal attackers. For that, a strict overlay structure is enforced, thus limiting the view and communication range of nodes so that eclipse and high level DoS attacks (e.g., sending a flood of requests) are prevented. Based on a public-private key-pair for every node, the system employs a simple reputation mechanism that enables a central party to exclude free-riding nodes and nodes that over-request packets from others. However, SecureStream does not prevent or limit the impact of bandwidth-depleting DoS attacks even though it enforces a strict overlay structure. Furthermore, agents that denounce well-behaving nodes could exploit the reputation mechanism.

[BSS09, BBG+09] introduces the concept of optimally stable topologies for push-based streaming, which enforces a certain structure of the distribution trees and balances the relevance of participating nodes. It is shown that, among all such topologies, the identified subclass minimizes the maximum possible attack damage. Thus, this concept does not limit the agents' abilities to take part in the topology construction, but it limits the impact of attacks.

Dieses Werk ist copyrightgeschützt und darf in keiner Form vervielfältigt werden noch an Dritte weitergegeben werden.
Es gilt nur für den persönlichen Gebrauch.