Resilient Peer-to-Peer Live-Streaming using Motifs

Lachezar Krumov, Adriana Andreeva and Thorsten Strufe
TU Darmstadt, Germany
[krumov,andreeva,strufe] [at] cs.tu-darmstadt.de

***Abstract*—High robustness against churn and resilience towards adverse behavior are the key requirements for reliable Peer-to-Peer streaming systems. Their highly inter-dependent nature, based on the cooperative service delivery between all peers, necessitates a systematic and structural resilience, which, in their design of self-organization and decentralized control, is very challenging to assure. Reliable services, and hence a structural resilience, still are a vital prerequisite for any commercial deployment of P2P-based live streaming systems or IPTV infrastructures. We propose an entirely distributed, Motif-based topology optimization to this end. Concise comparisons show that it creates topologies almost as resilient as the current state of the art, yet causing significantly less, almost negligible overhead in both computation and messaging, and still offering even better protection of information on the overall system.**

***Index Terms*—Multimedia Communication, Robustness, Distributed Control, Computer Network Reliability**

I. Introduction

Creating resilient topologies, and thus achieving a high robustness of Peer-to-Peer streaming systems, is one of the main challenges when designing a novel approach for this comparably new class of content distribution schemes. Peer-to-peer streaming promises to greatly relieve server load and aid in supplying large audiences with broadband multimedia content [1]. Two different services, video on demand and live streaming, create two fundamentally different problems in this field. While content can be distributed to some or all of the audience in advance in the first [2], [3], it is delivered directly upon creation in the latter, making it much more difficult to guarantee satisfactory provision without deficits. Introducing the cooperative delivery of peer-to-peer systems, in which the streaming packets are relayed between the nodes, further complicates the task. Forwarding peers are not only less performant than dedicated servers, but they additionally exhibit a much less reliable characteristic with high churn of frequently arriving and leaving, or even failing parties. Still, each node relies on the correct service of all preceding nodes on the packet path from the source, since failures and delays are propagated from peer to peer. Moreover, using the system to deliver controversial content, or simply considering any commercial deployment, makes the system a worthwhile target for parties with malicious intent. However, service degradation, or even disruption, be they caused by churn, failure, or DoS attack, is unacceptable, especially when the service is targeted at deployment in commercial scenarios.

Achieving robustness and resilience with peer-to-peer streaming systems hence is a challenging task, since they represent interdependent, large and complex networks, composed of highly heterogeneous nodes with an extremely dynamic behavior. Resilience in these systems consequently has to be achieved by decreasing the interdependence between the nodes while exploiting their resources to increase the scalability. For reasons of complicating attacks by malicious parties, this task is to be achieved with minimum disclosure of the system state.

In general, topology adaptation can either be implemented in a central oracle or by means of cooperative, distributed optimizations. Central instances represent single points of failure and potential bottlenecks, which makes them unfavorable when implementing large scale distributed systems. Distributed adaptation, however, is generally based on broad information on the overall state of the system, which is costly to gather and may be misused by malicious parties. An alternative approach to cooperatively adapting topologies is using metrics based on local information, and by optimizing the local state indirectly approximating overall desirable characteristics

In summary, we see the need to provide a scalable topology adaptation for live peer-to-peer streaming systems that achieves to create topologies of very low interdependency considering only a bare minimum of local information.

In previous work we have proposed a cost-based approach that, considering aggregated information on succeeding nodes, is able to create highly resilient topologies [4]. However, the cost metrics are computationally complex and knowledge about parts of the topologies has to be gathered and may potentially be exploited by malicious parties.

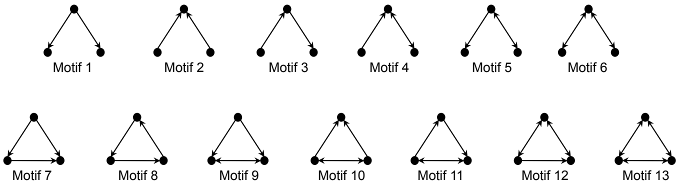

Fig. 1. Network Motifs

Network motifs, however, represent a metric much simpler to calculate, and their character of reproducing quite complex properties of the overall graphs is promising for their application in topology adaptation. They were introduced in 2002 by Uri Alon as a statistical measure for investigating complex networks [5]. The aim of network motifs is to close the gap between local and global knowledge of large networks. They are small k-subgraphs, with k usually being 3 or 4 (Fig. 1).

Motivating examples are most social and self-organizing real world networks, where it is exactly in the local environ-

978-1-4244-7265-9/10/$26.00 © 2010 IEEE

ments where the network participants make their decisions. Another interesting discovery about network motifs is that totally different real world networks fall in the same super families classified by their motif content [6].

According to the investigated network, the motifs are considered directed or undirected. By counting the number of occurrences of different motifs one can look at the local environment surrounding the single nodes. A motif is called *significant* if it is over- or under-presented in comparison with randomized networks with the same number of nodes, edges and an identical degree sequence. Still, the exponential growth of the number of different k-subgraphs with increasing k makes it computationally extremely expensive and even impossible for $k > 4$ within large networks.

Network motifs so far have only been used as a statistical measure [7], to the best of our knowledge. In this paper we propose to harness their ability to reproduce complex characteristics of the network for the purpose of adapting highly robust and resilient overlay live streaming topologies.

In this work, we make the following contributions: (i) we present an innovative approach using network motifs to build local decision rules for the self-optimization of the network; (ii) we improve the resilience of the topology to be close to the resilience of optimal topologies; (iii) evaluating our design in simulations, we illustrate the benefits of our approach compared to the state of the art.

The remainder of this article is organized as follows: We present some background in Section 2, reviewing the state of the art in Section 3. In Section 4 we describe the system design and explain the analytical model. Subsequently we evaluate our approach using different measures in Section 5, comparing it with a reference from the field in Section 6. We conclude in Section 7 and outline directions for future work.

II. Background

It is important to understand the basic principles and the major problems upon live streaming systems. Within such systems, there is one source, which provides the original streaming signal. All other participating peers are subscribing to that signal. The main goal is to build a topology providing each peer with the signal while being resilient to failures and misbehaving participants. To decrease the impact of failures as few dependencies among peers as possible should exist. This has the consequence, that the failure of a single or a subset of peers has the least possible impact on the service of the remaining peers.

In previous work we have defined a class of optimal topologies with proven resilience to DoS attacks. We additionally proposed a fully distributed approach for constructing nearly optimal streaming topologies [4]. We use this approach as a reference throughout this work and adopt its mathematical definition of the problem: Consider s being the source, or the originator of the streaming content, the system can be modeled as an undirected graph $G = (V, E)$ with finite set of N vertices $V = \{v_1, \ldots, v_N\}$, the data source s and a set of edges $E = \{(u, v) : u, v \in V\}$ along which the content is forwarded. The multimedia content can be modeled as a packet stream $S = \{p_1, ..., p_p\}$ of p packets. All p packets can be replicated at each vertex and originate at the data source s. The bitrate of the stream is denoted as R_0. To decrease the dependency between nodes, and hence vulnerability to attacks, the packet stream can alternatively be split into partial streams, with l sequences of k stripes: $S = \{\{p_{1_1}, \ldots, p_{1_k}\}, \ldots, \{p_{l_1}, \ldots, p_{l_k}\}\}$. Each stripe in consequence has an average bitrate of $\frac{R_0}{k}$. The source s has a bandwidth capacity of C times the bitrate R_0. That is, s can deliver the whole bitrate R_0 of the stream simultaneously C times, i.e. it can directly serve $C \cdot k$ stripes. We assume that all participating peers share at least $R_0 + c$ bandwidth in order to participate in the live streaming, and that in consequence they are able to receive the complete stream once and forward $c \cdot k$ stripes. In accordance with the state of the art, we assume c to be 2, and hence each node to be able to receive the stream once and forward the bitrate of the stream at least twice. All joining nodes locate nodes already part of the topology, select them as initial parents and request the stream from them.

An *attacker* is considered to aim at causing the highest possible damage to the system with the least necessary resources. With the benefit of the system being the number of stripes that are successfully and timely received at all nodes $(k \cdot N)$, the damage is measured as the fraction of stripes that are not successfully delivered in the system after the failure or removal of an arbitrary set of nodes. For this purpose the attacker is assumed to exploit knowledge used in the protocol to identify valuable targets: nodes of which a large set of other nodes are dependent. The attacker additionally is assumed to be able to either relocate itself to an identified position and subsequently stop the service, or to be able to launch attacks on identified nodes, thus disconnecting them from the system completely. Since disabling the source will result in immediate loss of the whole signal, the source is assumed to be hidden and cannot be attacked.

III. Related Work

Peer-to-peer streaming systems, commonly also referred to as overlay- or application layer multicast (ALM) [8], are usually divided into the groups of *pull-* and *push-based* systems [9], [10].

In pull-based ALM the video is split into chunks and each peer requests all chunks using some peer-to-peer system (like different distributed hash tables, or, more frequently, BitTorrent-like approaches [11], [12]). The chunks then are delivered to all peers along the reverse path of their issued requests, which, depending on the peer-to-peer method used, may vary quite notably. While naturally being more resilient to churn and attacks, pull-based approaches induce significant delays on the order of minutes to the streaming and thus are not viable for live streaming scenarios [13].

Push-based approaches follow the strategy of creating longer lived distribution topologies over the participating nodes, along which all streaming packets are delivered. They are characterized by significantly lower delay penalties for two main reasons: received packets are forwarded to succeeding nodes in the topology directly on reception; and since no per-packet management messages are needed, the stream can be split into smaller chunks, which then can be delivered in parallel. Since these approaches rely on the created topology, they generally are less resilient to churn and attacks. Node departures as well as node failures in push-based approaches have to be detected first and then encountered by repair and reconstruction of the distribution topology.

The main concept to increase the resilience of push-based ALM, besides introducing different encoding or other forward error correction mechanisms (cmp. [14], [15], [16]), is to decrease the dependency on potentially failing, preceding nodes in the topology [17].

This is achieved using two generally different strategies, feedback based predecessor selection, or pro-active topology optimization. The feedback based predecessor selection leverages on different types of reputation mechanisms in order to increase the dependency on highly reliable and available nodes, and decreasing the dependency on unreliable, inperformant, or potentially failing nodes [18], [19], [20], [21]. These approaches, while leading to quite good results, cause a significant communication or computational overhead and hence are not too attractive, especially for scenarios with very large groups of participating peers. A slightly orthogonal concept is to split the stream into different partial streams (commonly termed *stripes*) and delivering these along multiple paths, which are node and link disjoint at best. Split-Stream [22], DagStream [23], and BCBS [24] all follow this approach. However, they all lead to reasonably high topologies with long paths of dependencies between nodes, and additionally in reality fail to establish node disjoint paths [4]. Another way of decreasing the interdependency between nodes consequently is to create topologies of very short paths, thus generating potentially very fat trees [25]. These approaches are prone to massive damages in case of attacks: with knowledge on the protocol and gathering information on the topology, attackers are able to easily identify highly relevant nodes with large sets of successors, and consequently are given the opportunity to heavily disrupt the system by selecting them as targets. In previous work we consequently have defined a class of topologies, with short, node disjoint paths, that is optimally resilient to attacks. We additionally proposed an approach that creates topologies resembling this class, locally optimizing each nodes' neighborhood, based on local knowledge, only. Our previous approach, however, still gathers information about some properties of the succeeding subtrees, and causes a quite significant computational cost.

IV. System Design

In this section we describe the basic outline of our distributed approach and how it engages network motifs in local decision rules for constructing robust streaming topologies.

A. Network Motifs

In streaming networks, due to security implications, it is dramatically important that the peers know as little about the network topology as possible. Network motifs provide exactly the required non-intrusive way of looking at the environment surrounding a given peer. Therefore, we engage them in a simple decision rule. When the participating peers obey that rule they improve their local environment. More importantly, those multiple local changes lead to drastic improvement of the global properties of the optimized overlay network.

From the problem description it can be derived that the best case, thus the theoretically optimal solution, is to create an optimally balanced spanning tree of minimum height for each of the k stripes (cmp. Fig. 2).

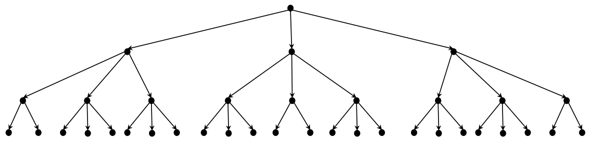

Fig. 2. Example of an optimal topology, $N = 37$, $d = 4$ (cmp. [4])

Our key observation is that the nodes in an optimal topology can be divided in three groups. In the first group are the *internal* nodes. They have the full number of successors: $d = c \cdot k$ and each of the successors has d successors on their turn. The group of internal nodes includes the root s. In the second group of nodes are the *intermediate* nodes. Those are the nodes which all have d successors, but all those successors are leaves. The intermediate nodes are the nodes on the layer before the last one in the topology. The last group of nodes are the *leaves* of the topology, which have no successors.

One observes that in the optimal topology only the motif 1 and motif 3 are represented, see Figures 1 and 2. Furthermore, the three groups of nodes have very distinct local motif signatures, i.e. the type and number of motifs a node is involved in. If one computes the local motif signature for each node in the optimal topology and order them in ascending order with respect to the ratio of motif 1 to motif 3, one gets a clear three range transition of local motif signatures, see Figure 3.

B. Engaging Network Motifs in Topology Optimization

To optimize the streaming topologies we consider the three ranges from Figure 3 to construct local decision rules. One observes that the few internal nodes have a specific motif ratio. Its value can be computed from the maximum number of successors d, representing the bandwidth capacities of the nodes. The number of motif 1 instances an internal node is involved in is equal to all pair combinations of its successors without repetition and is given by $\sum_{i=1}^{d-1} i$. The number of motif 3 instances is given by $\sum_{i=1}^{d} d_i$ because each successor has d successors on its turn and this results in d instances of

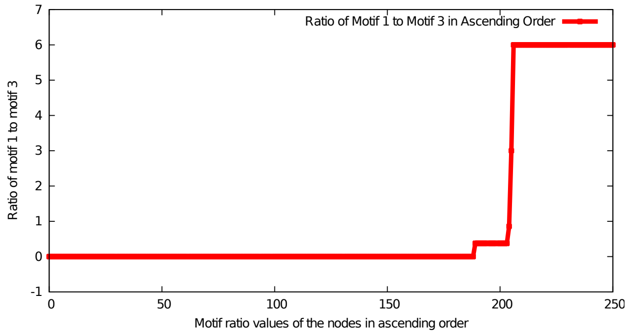

Fig. 3. Motif ratio range transition of an optimal topology

motif 3 per successor. The ratio θ of the motifs 1 and 3 for each internal node, which we call the *threshold*, is given by:

$$\theta = \frac{\sum_{i=1}^{d-1} i}{\sum_{i=1}^{d} d} = \frac{\frac{d(d-1)}{2}}{d^2} = \frac{d-1}{2d} = \frac{1}{2} - \frac{1}{2d} \qquad (1)$$

This threshold θ is the kernel of our approach.

C. Implementation

The source node s is the initial bootstrap node. It estimates the minimum bandwidth d the participating peers should provide depending on their expected configuration. Then, it divides the original signal in k stripes and waits for the rest of the participants to join the network. The value of θ is derived from the estimation of d and provided to the joining peers.

Each node is provided with a *Node Manager*. When a node joins the overlay, it randomly joins at all k stripes as a leaf. The NM monitors that the overall used bandwidth over all k stripe trees is less than the available bandwidth of its node.

The following is a pseudocode outline of the balance operation carried out by each peer p in some predefined time step for each stripe tree T_i:

```
Input: $p, T_i$, $\mathrm{Childs}_{T_i}(p) \leftarrow \{w$ child of $p$ in $T_i\}$
$Pred(p) \leftarrow \{w$ predecessor of $p$ in $T_i\}$;
$M_1 = \frac{|\mathrm{Childs}_{T_i}(p)||\mathrm{Childs}_{T_i}(p)+1|}{2}$;
$M_2 \leftarrow 0$;
$R \leftarrow 1$;
foreach $w \in \mathrm{Childs}_{T_i}(p)$ do
    $\mathrm{NumChilds}_{T_i}(w) \leftarrow \{$# of childs of $w$ in $T_i\}$;
    $M_2 \leftarrow M_2 + \mathrm{NumChilds}_{T_i}(w)$;
end
if $M_2 \neq 0$ then $R \leftarrow \frac{M_1}{M_2}$;
if $R > \theta$ then
    requestSuccessors(pickOne($\mathrm{Childs}_{T_i}(p)$));
else if $R < \theta$ then
    giveUpSuccessors($Pred(p)$);
end
```

Algorithm 1: Topology Control

All nodes only make requests (procedures $requestSucc$ and $giveUpSucc$) to their predecessors/successors and all transfers are made from successor to predecessors. This constraint guarantees that the resulting structures are indeed spanning trees as they are invariant to the transfer operations. Requested nodes have the right to reject the requested transfers, e.g., if the local bandwidth doesn't permit serving another successor. Figure 4 is an example of the two transfers.

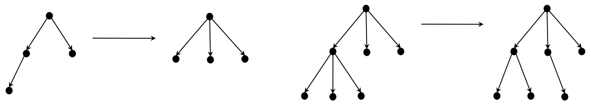

Fig. 4. Successor exchange operations

The only exception to the local decision rule is the root, which in general always denies giving up successors and pushes its bandwidth utilization to the maximum. It requests as many successors as bandwidth is available, and allocates it uniformly distributed over all k stripes.

The algorithm has converged when each of the stripe trees has converged. The algorithm has converged in a stripe tree when all nodes in that tree either have motif ratios equal to the threshold or no more improvements are possible due to local deadlocks.

Our approach improves the topology in each step and does not need to converge explicitly. The trees are functional at any time, it is just their resilience that increases in time.

V. Evaluation

In this section we present the results from extensive evaluations of our approach. They include the management overhead, the structural properties of the produced topologies and most importantly their resilience to attacks.

In all of our experiments we use the following scenario: There is one root node providing the system with the original streaming signal. It decides in how many stripes the signal should be divided and estimates d, the number of maximum successors per node. To enable comparison to our reference system we keep $c = 2$ fixed and divide the source signal into $k = 10$ stripes. Networks with 300 to 3,000 nodes are simulated for all experiments. All results are averaged over 10 repetitions of each experiment and all results are indicated with their standard deviation.

In our resilience simulations we evaluate worst case attacks, causing maximum damage to the topology. Since churn is a subset of that attack and no time measurements were taken, we refrained from packet level simulations and implemented a turn based simulation to generate the topologies.

We investigate our new approach from three different perspectives: management overhead, topological properties and resilience to attacks.

A. Management Overhead

Recall from Section IV that each node is provided with a Node Manager optimizing its motif content within each stripe.

Therefore, the overhead produced by our approach is given by the number of exchange steps per node and per stripe, caused by the Node Managers. That is, how many exchanges each node has to perform on average for the approach to converge. The results are displayed in Figure 5, the error bars are smaller than the data points.

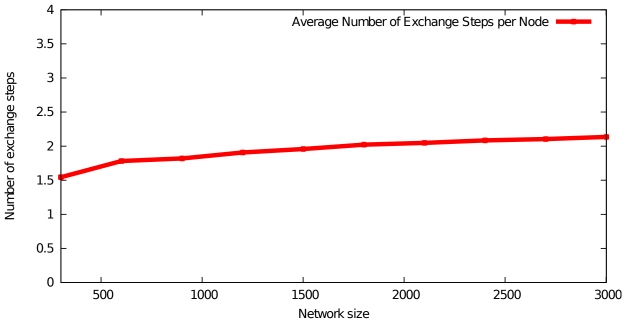

Fig. 5. Number of exchange steps per node per stripe

One observes that the average number of exchange operations per node per stripe is independent of the network size and in total it grows sublinearly in the network size. It increases from 1.5 to just a bit over 2 while the network size grows with one order of magnitude. Even more important is the small number of required exchange operations, which represents a negligible overhead per node.

B. Topological Properties

Now we know that the overhead per node produced by our approach is very small and independent of the network size.

Our next step is to investigate the quality of the produced topologies. For this purpose we use a set of four topological metrics to estimate their graph properties: *ToPo* metric, *tree height*, *Balance* metric and *vertex connectivity*. Additionally, we measure their resilience towards perfect attacks.

The first measure we apply to the generated topologies is a simple topological measure, which we call the *ToPo* metric. Given a tree, it reflects the balance in height of the subtree starting at each node. For each node the ToPo metric is defined as the difference between the longest and shortest branches of the succeeding subtree, starting at that particular node in the whole tree. Figure 6 shows a sample tree with the ToPo metric values of the nodes in the tree.

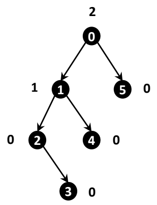

Fig. 6. Sample tree with corresponding ToPo metric values

The ToPo metric value of a tree is defined as the sum of the ToPo metric values of its nodes. In our example $0+0+0+1+2=3$. It follows, that in a tree with N nodes, which, with respect to its height, is perfectly balanced, the ToPo metric is zero. In the worst case where the whole tree is just a list, the ToPo metric value is equal to $\frac{N(N+1)}{2}$.

The ToPo metric is measured for each stripe tree and the value for the whole topology is the average over all its stripes. The results are displayed in Figure 7.

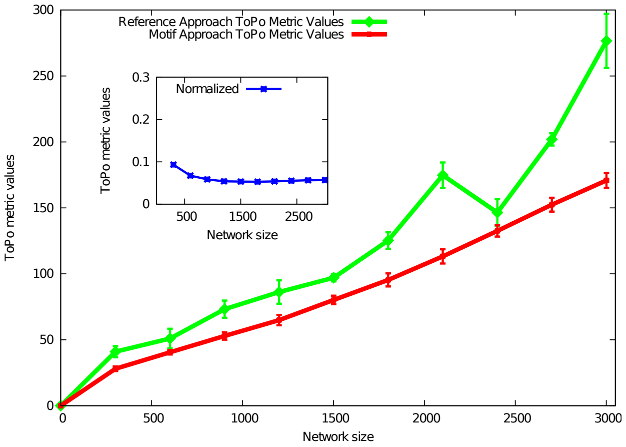

Fig. 7. ToPo metric values of streaming topologies (with std. deviation). Inset: the ToPo metric results of the motif approach normalized with respect to the network size

One observes that the normalized ToPo metric values of the generated topologies are independent of the network size. Furthermore, the motif based approach produces better balanced trees than the reference approach.

Recall from Section II that streaming topologies should have minimum predecessor/successor dependencies. Thus, they should be as flat as possible. Therefore, we also investigate the height of the generated topologies. It is defined as the average height of all different stripe trees within the topology. The results are displayed in Figure 8.

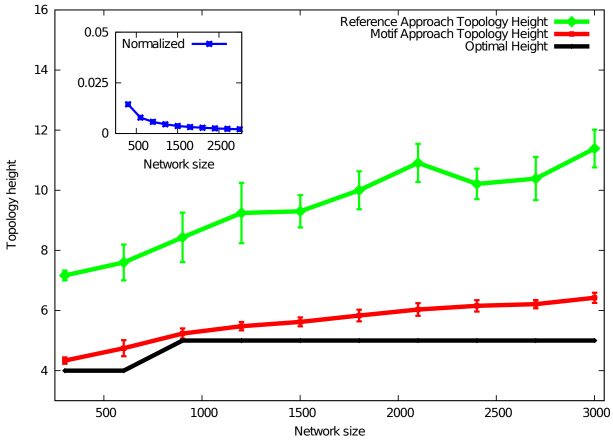

Fig. 8. Height of streaming topologies (with std. deviation). Inset: topology heights of the motif approach normalized with respect to the network size

Once again our normalized results are independent of the network size, confirmed by the inset in Figure 8. They are also clearly flatter than those of the reference approach and very close to optimal. For a perfectly balanced tree with N nodes and maximum allowed successors per node d, the height of the tree is given by $\lceil log_d N \rceil$.

Note that the cost function used in the reference approach is targeted at the direct neighbors of the root node. Therefore, it achieves almost uniform distribution of the nodes among the successors of the root and becomes suboptimal near the leafs in the topology. On the other hand, the motif based approach treats all nodes equivalent. That is why it performs better with respect to global topological properties than the reference approach, but exhibits more skewed distribution of the nodes among the direct successors of the root.

Next, we measure the *Balance*, the distribution of the nodes among the direct neighbors of the root. For a tree with N nodes and $|N_{root}|$ direct neighbors of the root the optimal distribution is $\frac{N-1}{|N_{root}|}$ nodes per subtree starting at each neighbor of the root. The balance metric for a given tree is defined as:

$$B(T) = \sum_{i \in N_{root}} \left| suc(i) - \frac{N-1}{|N_{root}|} \right| \qquad (2)$$

where T is the given tree, N_{root} the set of direct neighbors of the root, n the size of N_{root}, $suc(i)$ the number of nodes in the subtree starting at node i, including i, and N the number of all nodes in T. The balance metric values of both approaches are displayed in Figure 9.

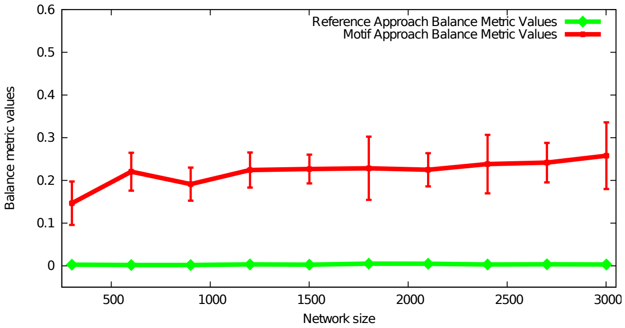

Fig. 9. Balance metric values of streaming topologies normalized with respect to the network size (with std. deviation)

One observes that the balance metric values of our approach are independent of the network size. As it was to be expected, they are not as stable and close to optimal as those of the reference approach. This is because the cost functions in the reference approach punish more severely imbalances closer to the root. The motif approach treats all nodes equally and therefore suffers from imbalances all over the topology and not only at the leaves. However, is due to the fact that our new approach does not rely on any information on the topologies other than the direct neighborhood of each node.

The next measure we apply on our topologies is the *vertex connectivity*. It counts the number of nodes that have to be removed from a given graph, such that it disintegrates into two disjoint parts.

Note that due to the division in 10 stripe trees, the theoretically optimal node connectivity lies at exactly 10 nodes. Figure 10 shows the evaluation results.

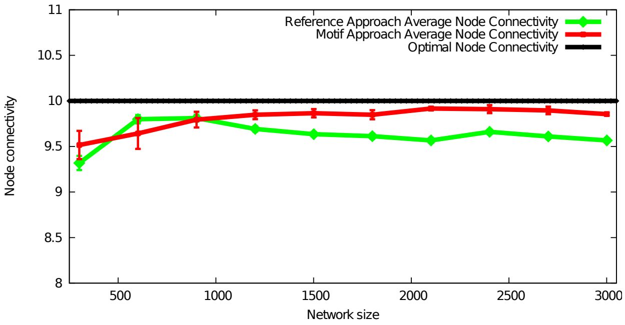

Fig. 10. Average node connectivity of streaming topologies (with std. dev.)

One observes that the node connectivity even increases with the network size. For networks with more than 1000 nodes it is independent of the network size around 9.7. Our approach once again outperforms the reference one.

C. Resilience

So far, the topological measures that reflect the dependencies among the participating peers have been investigated. Finally, we evaluate the resilience of our topologies toward perfect attacks with a metric called *stability* in [4]. The stability of a topology is given by the number of remaining received stripes after removing sets containing the *most important* nodes in the topology. The most important nodes are those whose absence leads to the highest possible damage of the service for the overall system. To determine the sets of the most important node for increasing set-sizes, we use exhaustive enumeration and branch and bound methods. That is a highly unrealistic scenario for streaming networks, but represents the worst case attack and thus an upper bound for possible damages through correlated failures, churn or any conceivable attack.

In previous studies we have shown, that other systems from related work perform roughly comparable to BCBS [24] or that they are even less resilient to perfect attacks [4]. We consequently chose BCBS as a benchmark, of which we can easily generate arbitrary topologies, since it indicates the upper bound of attack resilience of the related work.

We divide the source signal into 10 stripes and allocate a source bandwidth of $C = 2$. It follows that even in the perfect case, where each direct successor of the root in one of the stripe trees is a leaf in all other trees, it is the 20 nodes directly connected to the source that need to be removed to completely disrupt the service. The evaluation results are shown in Fig. 11.

One observes that our approach is almost independent of the system size. It produces similar results for network sizes differing by one order of magnitude. One also observes that there is a clear gap between the new approach and the

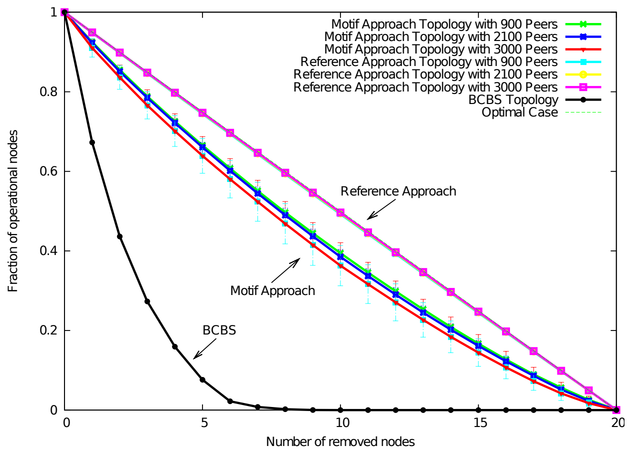

Fig. 11. Stability of streaming topologies (with std. deviation)

reference [4], which achieves almost perfect results. That is no surprise considering the differences with respect to the balance metric, see Figure 9. Still, our results are significantly better than those of comparable streaming topologies [4].

It is clear that our approach is not as optimal as the reference with respect to stability, but it produces very promising results. Furthermore, it has many advantages in other perspectives, which are discussed in the next section.

The bottom line is, that we have tested the streaming topologies generated by our new approach for their topological properties as well as their resilience to attacks. In all of the test cases the achieved results are very close to optimal and independent of the network size.

VI. Method Comparison

In this section we directly compare the motif based approach to our reference [4]. They both share some similar features, but each one of them has its advantages and drawbacks. To point out the most important of them we compare the methods on three different criteria: optimality of the constructed topologies; convergence and complexity; as well as their resilience and vulnerability to attacks by malicious nodes.

A. Topological Properties

We have shown that with respect to two different topological metrics (the ToPo metric and average three height, see Figures 7 and 8) the motif approach produces results close to optimal and even outperforms the reference. However, the complex cost functions used in the reference approach are targeted at the direct neighbors of the source. Therefore, with respect to the Balance metric it outperforms the new approach, see Figure 9. Still, both hardly leave space for improvements and produce good results.

B. Convergence and Complexity

Both approaches we discuss here provide streaming topologies operational throughout the whole simulation.

Our new approach requires only a few exchange operations per node, see Figure 5. The local decision rule is based only on a few simple computations and requires negligible effort from the peer in $O(1)$ computational complexity. The messaging overhead over all stripes per peer is in $O(kd)$, since it requires one value from each neighbor in each stripe. The reference approach causes a computational complexity in $O(d^2)$ and produces the same messaging overhead $O(kd)$. Therefore, the new approach is better suited for devices with little or constrained resources, e.g., most mobile devices.

C. Network Resilience

The resilience against attacks and failures is the outstanding feature of the two approaches we discuss and therefore our main point of comparison.

We first tested the node connectivity of the generated topologies, see Figure 10. That is, the minimal number of nodes that have to be removed, such that the underlying topology breaks down into two separate fragments. In this test case the new approach outperformed the reference approach. However, they both produce close to optimal results, leaving almost no space for improvements.

Subsequently, we have performed perfect attacks on the topologies generated by the two approaches. Both attacks rely on complete network knowledge. The possession of such knowledge is highly unrealistic for distributed systems, but is indeed the ultimate challenge for both approaches. This attack tests the stability of the generated topologies (cmp. Fig. 11). It measures the number of residually received stripes after a perfect attack on the network has been performed. The perfect attack considers global knowledge and is carried out through exhaustive enumeration and branch and bound methods. Being unrealistic, it merely represents an upper bound of possible damage through correlated failure or attacks.

In this test case the reference outperformed the new approach, managing almost perfect, linear decay of the operational nodes. Nevertheless, the stability of the topologies generated by the motif approach is dramatically higher than those of random topologies or topologies generated by other ALM approaches.

The bottom line is, that the reference provides higher resilience to attacks (and failures as a special case). Our new approach, on the other hand, performs very well, too, while causing significantly less computational complexity.

A clear advantage of the new approach is the fact that no knowledge about the topologies is gathered or forwarded, compared to the reference approach that aggregates and forwards information about the underlying topologies. Any miscreant capable of gathering knowledge about the topologies represents a threat, since it might be enabled to infer facts on its current location in the spanning trees of each stripe, and consequently estimate the situation of at least parts of the system. Furthermore, in the new approach the behavior of the root node with respect to its successors is no different than any other node. Thus, even when a malicious node is as high as being a direct successor of the root, there is no way for it to determine that.

VII. Conclusions

This paper studies the resilience of peer-to-peer live streaming topologies. The considered scenario consists of a source peer, which provides the original signal, and further peers that are interested in the stream and provide parts of their available bandwidth to cooperatively distribute it among each other. These systems hence harness the resources at the end-hosts and thus greatly decrease the server load and aid scalability to large audiences.

With each peer relying on the correct service of all preceding peers on the packet path from the source, these systems are prone to experience service disruption due to delays or departing peers. Especially considering a commercial deployment, service degradation or disruption are unacceptable. Taking into account the existence of malicious parties complicates the problem even further due to the cooperative and open nature of these systems.

To this end, we propose a novel approach for constructing streaming topologies that are resilient to node failures and DoS attacks. It employs network motifs, a fairly new statistical metric, originally introduced to analyze networks in biology, and relies on decision rules based on local knowledge of the nodes, only. The approach consequently does not provide participating parties with knowledge on their position within the network nor its overall state. Hence, attackers have no means of inferring the position of other nodes nor their importance, in order to identify valuable targets for attacks.

In extensive evaluation we compare our new approach to another approach from previous work, which has been shown to produce streaming topologies that are almost optimally resilient towards DoS attacks [4]. The comparison includes a series of topological measures as well as their response to perfect attacks. The results indicate that both approaches achieve a comparable performance. The reference approach achieves a slightly better resilience to attacks, yet it relies on gathering some knowledge on the succeeding topologies of each node and is characterized by causing much higher computational complexity. The motif based approach on the other hand achieves better topological properties and is independent of any information on the topology other than the direct neighborhood of each node. Both approaches create topologies that are significantly more resilient to DoS attacks compared to the related work.

Our new approach currently only aims at creating resilient live streaming topologies. However, common objectives in this scenario are to decrease the end-to-end delay and to achieve location awareness to efficiently use the underlying infrastructure network. We are currently in the process of extending the approach to incorporate location information to create network efficient streaming topologies. We are additionally adapting protocol and local decision rules to decrease overlay path lengths and the observed delays. On a broader view we are pursuing the question, whether such simple local decision rules can be applied in other decentralized settings, such as routing and topology adaptation in wireless sensor networks.

Acknowledgment

The authors would particularly like to thank Karsten Weihe for many useful comments and suggestions. This work in parts has been funded by Volkswagen Foundation under grant I/82 717, ETRI, and DFG Research Group 733 (QuaP2P).

References

[1] T. Small, B. Liang, and B. Li, "Scaling laws and tradeoffs in peertopeer live multimedia streaming," 2006.

[2] S. Annapureddy, C. Gkantsidis, and P. Rodriguez, "Providing video-on-demand using peer-to-peer networks," in *IPTV Workshop, WWW*, 2006.

[3] K. Graffi, S. Kaune, K. Pussep, A. Kovacevic, and R. Steinmetz, "Load balancing for multimedia streaming in heterogeneous peer-to-peer systems," in *NOSSDAV*, 2008.

[4] M. Brinkmeier, G. Schäfer, and T. Strufe, "Optimally dos resistant p2p topologies for live multimedia streaming," in *IEEE TPDS*, 2009.

[5] U. Alon, "Network motifs: theory and experimental approaches," in *Nature Reviews Genetics 8*, 2007.

[6] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of designed and evolved networks," in *Science 303*, 2004.

[7] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," in *PNAS 100*, 2003.

[8] Y. H. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," *IEEE JSAC*, vol. 20, no. 8, 2002.

[9] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *IPTPS*, 2005.

[10] D. Carra, R. L. Cigno, and E. W. Biersack, "Graph-based analysis of mesh overlay streaming systems," in *IEEE JSAC vol. 25*, 2007.

[11] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. van Steen, and H. Sips, "TRIBLER: a social-based peer-to-peer system," *Concurrency and Computation: Practice & Experience*, 2008.

[12] D. Carra, G. Neglia, and P. Michiardi, "On the Impact of Greedy Strategies in BitTorrent Networks: The Case of BitTyrant," in *IEEE P2P*, 2008.

[13] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari, "Will IPTV ride the peer-to-peer stream?" *Communications Magazine*, 2007.

[14] D. Nguyen, T. Tran, T. Pham, and V. Le, "Internet Media Streaming Using Network Coding and Path Diversity," in *IEEE Globecom*, 2008.

[15] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan, "Resilient multicast using overlays," in *ACM SIGMETRICS Performance Evaluation Review*, 2003, pp. 102–113.

[16] V. N. Padmanabhan and K. Sripanidkulchai, "The case for cooperative networking," in *IPTPS*, 2002.

[17] S. Grau, M. Fischer, M. Brinkmeier, and G. Schaefer, "On complexity and approximability of optimal dos attacks on multiple-tree p2p streaming topologies," *TDSC*, 2010.

[18] W. Wang, Y. Xiong, Q. Zhang, and S. Jamin, "Ripple-Stream: Safeguarding P2P Streaming Against Dos Attacks," in *IEEE Multimedia and Expo*, 2006.

[19] J. Yang, Y. Li, B. Huang, and J. Ming, "Preventing DDoS Attacks Based on Credit Model for P2P Streaming System," in *Autonomic and Trusted Computing*, 2008.

[20] W. Conner, K. Nahrstedt, and I. Gupta, "Preventing DoS Attacks in Peer-to-Peer Media Streaming Systems," in *Proceedings of SPIE*, 2006.

[21] M. Rossberg, T. Strufe, and G. Schäfer, "Using Recurring Costs for Reputation Management in Peer-to-Peer Streaming Systems," in *3rd IEEE SecureComm*, 2007.

[22] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in cooperative environments," in *19th ACM SOSP*, 2003.

[23] J. Liang and K. Nahrstedt, "DagStream: Locality Aware and Failure Resilient Peer-to-Peer Streaming," in *Multimedia Computing and Networking*, vol. 6071, 2006.

[24] T. Strufe, G. Schäfer, and A. Chang, "BCBS: An Efficient Load Balancing Strategy for Cooperative Overlay Live-Streaming," in *IEEE ICC*, 2006.

[25] S. Birrer, D. Lu, F. Bustamante, Y. Qiao, and P. Dinda, "FatNemo: Building a resilient multi-source multicast fattree," in *9th International Workshop on Web Content Caching and Distribution*, 2004.