Attacker Independent Stability Guarantees for Peer-2-Peer-Live-Streaming Topologies

Andreas Brieg, Michael Brinkmeier, Sascha Grau,
Mathias Fischer, Guenter Schaefer

This work was in part supported by the
Deutsche Forschungsgemeinschaft under grant
numbers KU658/10-1 and SCHA1533/1-1.

P2P-Live-Streaming - What & Why?

Goal

Realtime distribution of continously generated multimedia-stream to varying and potentially large set of viewers.

P2P-Live-Streaming - What & Why?

Goal

Realtime distribution of *continously generated* multimedia-stream to *varying* and *potentially large* set of viewers.

P2P-Live-Streaming - What & Why?

Goal

Realtime distribution of *continously generated* multimedia-stream to *varying* and *potentially large* set of viewers.

Key Idea

Incorporate viewers' resources for distribution to overcome problems of classical Client-Server approach:

- restricted bandwidth resources at server
- high hardware costs
- inefficient traffic patterns (all paths lead to server)

Packet distribution in trees

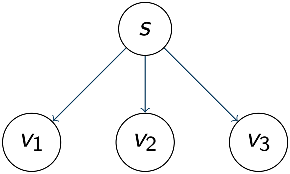
Packets:

- enter the system at source node s

Packet distribution in trees

Packets:

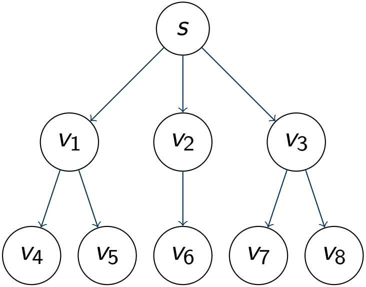
- enter the system at source node s
- distributed to direct neighbors



Packet distribution in trees

Packets:

- enter the system at source node s
- distributed to direct neighbors
- replicated and re-distributed to other nodes

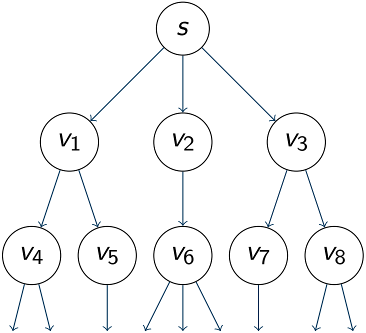


Packet distribution in trees

Packets:

- enter the system at source node s
- distributed to direct neighbors
- replicated and re-distributed to other nodes . . .

Packet distribution over *Spanning Trees*!

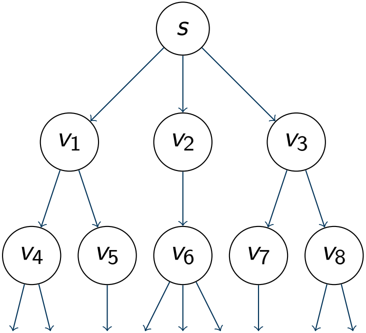


Packet distribution in trees

Packets:

- enter the system at source node s
- distributed to direct neighbors
- replicated and re-distributed to other nodes ...

Packet distribution over *Spanning Trees*!



Service quality of peers in low levels of the tree depends on *cooperation* and *health* of *all* nodes in its path to the source.

Problems of P2P-Live-Streaming systems

But peers...

- constantly join and leave the system
- have small resources
- are vulnerable to attacks and have high failure probability

Problems of P2P-Live-Streaming systems

But peers...

- constantly join and leave the system
- have small resources
- are vulnerable to attacks and have high failure probability

A key idea

Using multiple distribution trees with varying inner nodes decreases dependency on single nodes.

Model of push-based P2P-Streaming systems (1)

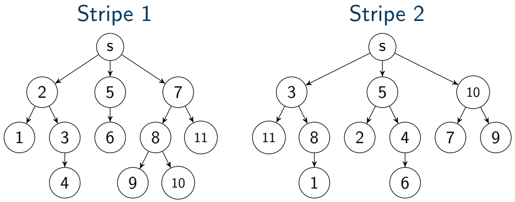
Basic model

- Stream is divided into k substreams called *stripes*
- Participants $V = \{s, v_1, \ldots, v_n\}$ are nodes of a graph G
- Stripe i is distributed using a directed spanning tree T_i over V
- *Streaming Topology* $\mathcal{T} = \{T_1, \ldots, T_k\}$ is set of these k distribution trees

Model of push-based P2P-Streaming systems (2)

More definitions...

- Assumption: source has a maximum degree of $C \cdot k$, for $C \in \mathbb{N}^+$
- Nodes receiving packets directly from s are called *heads* of $\mathcal{T}$
- The *successors* $succ_i(v)$ *of a node* $v \in V$ *in* $T_i \in \mathcal{T}$ are all nodes of the maximal subtree of T_i that is rooted in v



$C = 3$

Heads $H = \{2, 3, 5, 7, 10\}$

$\text{succ}_1(2) = \{1, 2, 3, 4\}$

S. Grau
ITI, TU Ilmenau, Germany

TECHNISCHE UNIVERSITÄT ILMENAU

What do we aim for?

Goal

1. *Identify* the class of all streaming topologies that are *optimally stable* against node failures due to malicious DoS attacks.
2. Provide rules for their *efficient* construction.
3. Design and implement *distributed* topology management mechanisms realizing stable topologies in P2P-streaming systems.

Attackers, damage and stability (1)

Abstract attacker

A map from $\mathcal{T}$ and $x \in \mathbb{N}$ to a set $X \subseteq V \setminus \{s\}$ of x failing peers.

Attackers, damage and stability (1)

Abstract attacker

A map from $\mathcal{T}$ and $x \in \mathbb{N}$ to a set $X \subseteq V \setminus \{s\}$ of x failing peers.

Why exclude source s?

- Source attack would always be optimal.
- Would disregard influence of distribution topology
  $\longrightarrow$ seemingly equal stability of P2P and client-server approach

S. Grau
ITI, TU Ilmenau, Germany

Attackers, damage and stability (1)

Abstract attacker

A map from $\mathcal{T}$ and $x \in \mathbb{N}$ to a set $X \subseteq V \setminus \{s\}$ of x failing peers.

Why exclude source s?

- Source attack would always be optimal.
- Would disregard influence of distribution topology
  $\longrightarrow$ seemingly equal stability of P2P and client-server approach

Damage function $a_{\mathcal{T}}(X)$

The *damage function* $a_{\mathcal{T}} : 2^V \rightarrow \mathbb{R}$ quantifies the damage incured on $\mathcal{T}$ by the failure of nodes.

S. Grau
ITI, TU Ilmenau, Germany

Attackers, damage and stability (2)

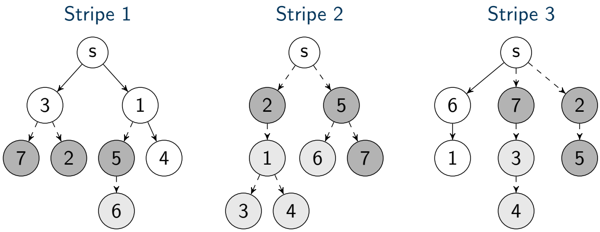
In this work, we chose on the *packet loss* damage function, summing up the number of successors of nodes in X over all stripes.

$$a_{\mathcal{T}}(X) = \sum_{i=1}^{k} \left| \bigcup_{v \in X} \mathrm{succ}_i(v) \right|$$

Attackers, damage and stability (2)

In this work, we chose on the *packet loss* damage function, summing up the number of successors of nodes in X over all stripes.

$$a_{\mathcal{T}}(X) = \sum_{i=1}^{k} \left| \bigcup_{v \in X} \mathrm{succ}_i(v) \right|$$



$a_{T_1}(X) = 4$

$a_{T_2}(X) = 7$

$a_{T_3}(X) = 5$

$a_{\mathcal{T}}(X) = 16$

The complete class of optimally stable streaming topologies (1).

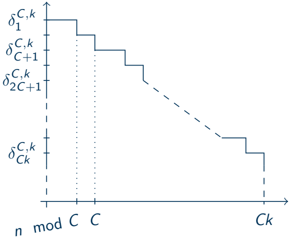
The complete class of optimally stable streaming topologies is characterized by the damage incured by an optimal attacker.

The complete class of optimally stable streaming topologies (1).

The complete class of optimally stable streaming topologies is characterized by the damage incured by an optimal attacker.

For $l = (i \text{ div } C)$ and $h = (i \mod C)$, define

$$\delta_i^{C,k} = \begin{cases} \left\lceil \frac{n}{C} \right\rceil + (k - 2l - 1) & \text{if } h \leq (n \mod C) \\ \left\lfloor \frac{n}{C} \right\rfloor + (k - 2l - 1) & \text{otherwise} \end{cases}$$



The complete class of optimally stable streaming topologies (2).

For $l = (i \text{ div } C)$ and $h = (i \mod C)$, define

$$\delta_i^{C,k} = \begin{cases} \left\lceil \frac{n}{C} \right\rceil + (k - 2l - 1) & \text{if } h \leq (n \mod C) \\ \left\lfloor \frac{n}{C} \right\rfloor + (k - 2l - 1) & \text{otherwise} \end{cases}$$

Optimally stable topologies [1]

A topology $\mathcal{T}$ with parameters C, k, n is optimally stable if and only if $a_{\mathcal{T}}(\mathcal{O}(\mathcal{T}, m)) = \sum_{i=1}^{m} \delta_i^{C,k}$ for $1 \leq m \leq C \cdot k$.

[1] Brinkmeier et. al., "Optimally DoS Resistant P2P Topologies for Live Multimedia Streaming", *IEEE Transactions on Parallel and Distributed Computing*, 2009

The bad news.

Decision problem

Decide whether any given streaming topology $\mathcal{T}$ is optimally stable.

We have shown that this problem is co-NP-complete.
Hence, without P=NP, it is *not solvable in polynomial time*.

S. Grau
ITI, TU Ilmenau, Germany

The bad news.

Decision problem

Decide whether any given streaming topology $\mathcal{T}$ is optimally stable.

We have shown that this problem is co-NP-complete.
Hence, without P=NP, it is *not solvable in polynomial time*.

All is not lost

We can identify a large and easy-to-check subclass of optimally stable topologies!

Necessary properties of stable topologies (1)

Stable topologies must follow a number of necessary rules.

Not-Too-Many-Successors Rule

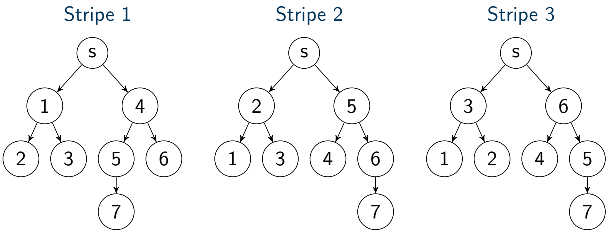
Every peer has at most $\delta_1^{C,k} = \lceil \frac{n}{C} \rceil + (k-1)$ successors.

S. Grau
ITI, TU Ilmenau, Germany

Necessary properties of stable topologies (2)

Head Rules

1. In each stripe each head adjacent to the source has exactly one head from each other stripe as a successor.
2. If $u, v \in V$ are heads and $u \in \mathrm{succ}(v)$, then $|\mathrm{succ}(u)| = |\mathrm{succ}(v)|$.



Necessary properties of stable topologies (3)

Head topology of $\mathcal{T}$

The *head topology* $\mathcal{H}(\mathcal{T})$ of a topology $\mathcal{T}$ is a streaming topology over node set $V_{\mathcal{H}(\mathcal{T})} = \{v \in V \mid v \text{ is head in } \mathcal{T}\} \cup \{s\}$ and in tree T_i, an edge (u, v) exists if $v \in \mathrm{succ}_i(u)$ in $\mathcal{T}$.

Necessary properties of stable topologies (3)

Head topology of $\mathcal{T}$

The *head topology* $\mathcal{H}(\mathcal{T})$ of a topology $\mathcal{T}$ is a streaming topology over node set $V_{\mathcal{H}(\mathcal{T})} = \{v \in V \mid v \text{ is head in } \mathcal{T}\} \cup \{s\}$ and in tree T_i, an edge (u, v) exists if $v \in \text{succ}_i(u)$ in $\mathcal{T}$.

Heads-Are-Optimally-Stable Rule

For topology $\mathcal{T}$ to be stable, $\mathcal{H}(\mathcal{T})$ has to be optimally stable.

Necessary properties of stable topologies (3)

Head topology of $\mathcal{T}$

The *head topology* $\mathcal{H}(\mathcal{T})$ of a topology $\mathcal{T}$ is a streaming topology over node set $V_{\mathcal{H}(\mathcal{T})} = \{v \in V \mid v \text{ is head in } \mathcal{T}\} \cup \{s\}$ and in tree T_i, an edge (u, v) exists if $v \in \text{succ}_i(u)$ in $\mathcal{T}$.

Heads-Are-Optimally-Stable Rule

For topology $\mathcal{T}$ to be stable, $\mathcal{H}(\mathcal{T})$ has to be optimally stable.

- Untrivial requirement.
- Large class of stable head topologies has been identified since paper submission.

A large and easily checkable subclass of optimally stable topologies

Shown requirements are not sufficient to guarantee optimal topology stability.

But: complexity of decision problem traced back to existence of non-heads with head-like successor number.

A large and easily checkable subclass of optimally stable topologies

Shown requirements are not sufficient to guarantee optimal topology stability.

But: complexity of decision problem traced back to existence of non-heads with head-like successor number. **Forbid them!**

A large and easily checkable subclass of optimally stable topologies

Shown requirements are not sufficient to guarantee optimal topology stability.

But: complexity of decision problem traced back to existence of non-heads with head-like successor number. **Forbid them!**

Strictly-Not-Too-Many-Successors Rule

Every head has at most $\delta_1^{C,k} = \lceil \frac{n}{C} \rceil + (k-1)$ successors and every non-head has at most $\delta_{Ck}^{C,k} = \lfloor \frac{n}{C} \rfloor - k - 1$ successors.

A large and easily checkable subclass of optimally stable topologies (2)

Theorem

A streaming topology $\mathcal{T}$ satisfying

- Head Rules 1
- Head Rules 2
- Heads-Are-Optimally-Stable Rule
- Strictly-Not-Too-Many-Successors Rule

is optimally stable.

S. Grau
ITI, TU Ilmenau, Germany

A large and easily checkable subclass of optimally stable topologies (2)

Theorem

A streaming topology $\mathcal{T}$ satisfying

- Head Rules 1
- Head Rules 2
- Heads-Are-Optimally-Stable Rule
- Strictly-Not-Too-Many-Successors Rule

is optimally stable.

- Easy to construct.
- Membership checkable in polynomial time.

Stable topologies in 'the wild'.

Practical topology construction would demand for *distributed construction mechanisms*.

Stable topologies in 'the wild'.

Practical topology construction would demand for *distributed construction mechanisms*.

Current rule set still seems to require central coordination of heads. Options:

- Special treatment $\longrightarrow$ nodes learn about their head status
- Approximation by additional rules

Stable topologies in 'the wild'.

Practical topology construction would demand for *distributed construction mechanisms*.

Current rule set still seems to require central coordination of heads. Options:

- Special treatment $\rightarrow$ nodes learn about their head status
- Approximation by additional rules

Current Implementation

Cost functions based on

- stripe-specific successor numbers of children
- nodes prefer forwarding single stripe: one-stripe-only rule

Simulations [2]: topology properties near optimum

Conclusion & Outlook

Conclusion

- Optimally stable topologies exist.
- General optimally stable topologies are hard to identify.
- Simple rule set defines a large, easy-to-check subclass.

Conclusion & Outlook

Conclusion

- Optimally stable topologies exist.
- General optimally stable topologies are hard to identify.
- Simple rule set defines a large, easy-to-check subclass.

Outlook

- Distributed construction still challenging problem.
- Assuming Multiple Description Coding or Forward Error Correction, more complex damage measures regarding indivual service loss of nodes can be introduced.
  - Hardness of attacker problems already studied in [3]
  - Optimal topologies are topic of ongoing research.

Thank you for your attention!

Do you have questions?

Bibliography

- A. Brieg, "Classification of optimally stable live-streaming topologies (German: Klassifikation optimal stabiler Live-Streaming Topologien)," Master's thesis, TU Ilmenau, 2008.
- M. Brinkmeier, G. Schaefer, and T. Strufe, "Optimally DOS Resistant P2P Topologies for Live Multimedia Streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 6, pp. 831–844, 2009.
- S. Grau, M. Fischer, M. Brinkmeier, and G. Schaefer, "On Complexity and Approximability of Optimal DoS Attacks on Multiple-Tree P2P Streaming Topologies," *submitted to IEEE Transacations on Dependable and Secure Computing*, 2009.