

# An Overlay for Ubiquitous Streaming over Internet

Chai Kiat Yeo<sup>1</sup>, Bu Sung Lee<sup>1</sup>, and Meng Hwa Er<sup>2</sup>

<sup>1</sup> School of Computer Engineering

<sup>2</sup> School of Electrical & Electronics Engineering

Nanyang Avenue, S639798, Singapore

{Asckyeo, Ebslee, Emher}@ntu.edu.sg

**Abstract.** Conventional distribution of real-time multimedia data uses multicasting or a series of relays and tunnels for unicast networks. The former is a capability not popularly enabled by a lot of networks while the static relays cannot readily adapt to changing network conditions and are potential bottlenecks in a heavily accessed system. This paper proposes a dynamic overlay framework for streaming multimedia data over heterogeneous networks. The overlay comprises a self-improving tree which is built from client relays on the fly and a lightweight server to manage the tree. The overlay provides a better QoS than conventional relays as it monitors the network and re-configures the tree to adapt to changing environments. Clients can switch parents for better QoS. The robustness of the tree is improved by using a spiral mechanism and failure of the lightweight server will not impact the data distribution functionality of the existing tree.

## 1 Introduction

The IP multicast [1] has been a highly efficient delivery mechanism for best-effort, large-scale, multi-point delivery of real-time multimedia data. However, Internet Service Providers and organisations deliberately disable multicast traffic to protect their networks against unwanted traffic. With the increasing popularity of multicast and broadband applications, the only way then for intranet clients and multicast-disabled networks to access multicast sessions is through a combination of tunnelling and a network of static relays. [2] and [3] are examples of such applications.

[2] proposes a hierarchical configuration of reflectors to act as unicast-multicast bridges. It uses a clustered-based approach by the manual placement of distributed servers at bottlenecks in the network to balance the load. The problem with this approach is the inability of the system to respond to rapid changes in the network and the potential of these servers becoming bottlenecks themselves.

[3] proposes a centralised framework for developing collaborative applications using a lightweight application level multicast tunnelling called mTunnel [4]. A centralised server is used to view, manage and effect all tunnelled sessions with specific gateways employed to unify unicast-multicast clients. Its drawback is the potential bottleneck in host processing capability and network resources.

## 2 Framework Overview and Design

Fig. 1 shows the architecture and the operation of the framework. It comprises the Directory Server (DS), the Web Server (WS) and the overlay tree of client nodes. The overlay tree is responsible for the distribution of data streams while DS is only responsible for the management functions. Hence the load of DS is vastly reduced compared to [3]. WS provides the GUI for sources to advertise their sessions. A separate overlay is built for different sources.

### 2.1 Overlay Construction and Operation

A source can either be unicasting or multicasting. The former will have to advertise its session by contacting WS (Step 1) while the latter will be automatically discovered by WS via the Session Directory Service (sdr) (Step 1) [5]. The overlay is built using DS as a point of contact [6]. The tree-only approach is much less complex than the tree-mesh approach adopted in [7] and [8]. Note that should DS fail, data distribution will still function normally except that new clients cannot join the tree until DS recovers.

Fig. 2 shows an example of a 4-level overlay tree, rooted at the source. Level 1 clients are multicast-enabled clients (C1 and C2 linked by dotted lines to the source) and proxies (Proxy 1) set up by the framework. The proxies act as relays for unicasting sources as well as a parent for the first unicast client joining the tree. It also doubles up as a static relay in the event of severe client failures. Clients from Level 2 onwards are simply members who join the group over time.

A new member selects the session to join from WS (Step 2 of Fig. 1) and issues a join request to DS (Step 3). DS will search its database and returns a list of potential parents (Step 4) using an algorithm which is similar to Prim's [9], commonly used to derive the minimum spanning tree in multicast routing. The clients are categorised into four groups, i.e. 1 to 4 based on Round Trip Time (RTT) between the client and the source. The categories are derived from data provided by [10, 11]:

Cat 1 $RTT < 100$ ms	Cat 2 $100$ ms $< RTT < 200$ ms
Cat 3 $200$ ms $< RTT < 400$ ms	Cat4 $> 400$ ms

Cat 1 clients are always chosen to be the parent for a client to ensure that the chosen parent is closest to the originating source. Note that unlike Prim's algorithm, the process does not necessarily mean that the chosen parent is closest to the client. However, the proposed framework is self-improving such that the clients converge towards the closest parent, ultimately reverting to Prim's algorithm again. DS will return a list of 5 parents (where available) in ascending order of categories with a maximum of 3 Cat 1 clients, 1 Cat 2 client and 1 Cat 3 client. The latter two clients are selected randomly. The client will then establish connections with the given Cat 1 potential parents (Step 5) and connect to the parent with the best QoS i.e. closest to it as per Prim's algorithm and update the DS (Step 6).



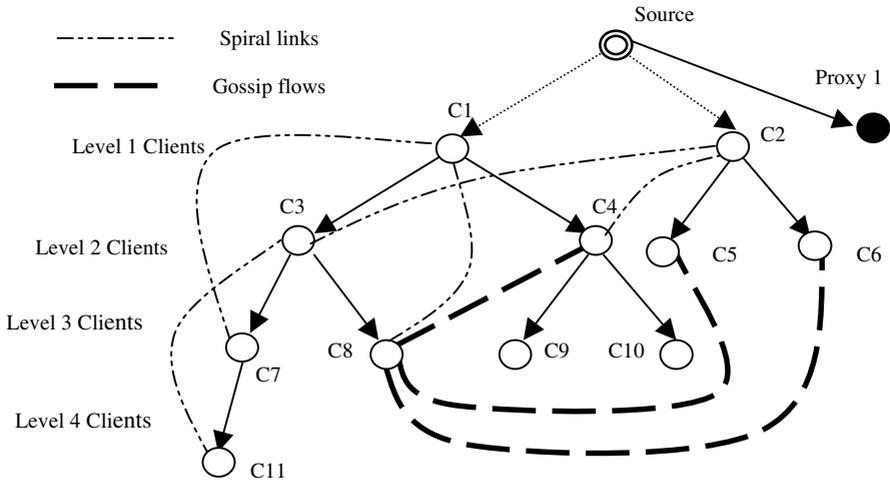


Fig. 2. Example of an Overlay Tree with Spiral and Gossip Mechanisms

### 2.3 Overlay Robustness

Membership on the overlay tree is dynamic as clients join and leave the tree and experience failures. Spirals shown in Fig. 2 are incorporated to strengthen the tree without incurring the complexity of a full mesh. Spirals can basically withstand node failures in any of its overlay tree branches so long as these failures are not consecutive nodes of the same branch. Client maintains a connection with its grandparent so that should a parent fail, it simply connects to its grandparent without needing to request for a new list of potential parents from DS. Information of the grandparent is passed to the client when it first establishes connection with its parent. Fig. 2 shows spirals from C11 to C3, C7 to C1 which can withstand the failure of clients C7 and C3 respectively. Level 2 clients who do not have grandparents will spiral with the siblings of their parents, e.g. C3 to C2 and C4 to C2.

For consecutive node failures in the same branch, recovery is via the gossip mechanism. If all else fails, the client can simply request DS for a new parent. Client who leaves voluntarily will inform its children, parent, grandparent and grandchildren about its impending departure. The child nodes will then connect to their grandparent (which is the leaving client's parent) immediately. The children who spirals with the leaving client will similarly switch to the leaving client's parent for spiralling.

## 3 Performance

The framework is implemented in Java using JDKv1.3 and JMF2.1. It has been tested on Win 98/NT and Solaris. Fig. 3 shows the overlay used in the experiments. All the

clients are connected via a 100 Base-T switch in a Local Area Network. The clients are Intel P3 500 MHz PCs with 128 MB SDRAM, installed with Win 98 OS. An MPEG2 source with a peak rate of 3.5 Mbps is used. Results are compared to 6 unicast clients sourced by a conventional single static source and the ideal case of 6 multicast-abled clients connected to a multicast source.

### 3.1 Loss Measurements

The average loss rate per client, shown in Fig. 4, is captured over 10 runs and the experiment is repeated by varying the number of Level 2 and 3 clients from 2 to 4 to 6. Multicast is most efficient for streaming multimedia data regardless of client number with an average loss rate of 0.27%. The static server unicast setup is the least efficient as it cannot scale unlike the overlay tree which scales much better given its distributed nature. The average loss packet per client increases from 0.72% to 1.25% when the number of clients increases from 4 to 6.

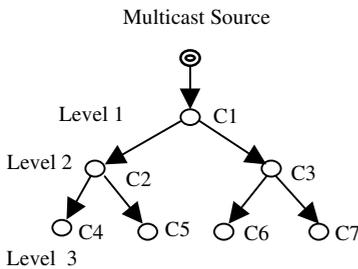


Fig. 3. Overlay used in Experiments

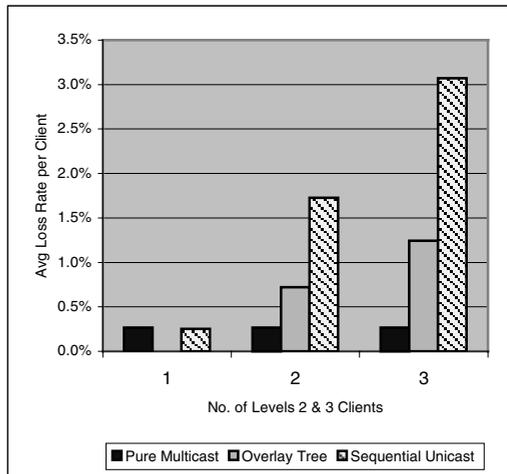


Fig. 4. Average Loss Rate

### 3.2 Inter-Level Latency

The inter-level latency, shown in Table 1, is measured through a series of ping requests between client and parent as the number of clients varies. The delay is insignificant although it should be noted that it increases with the client number as the response time of the parent gets longer when the parent services more children.

Table 1. Inter-level Latency

# of Level 2 & 3 Clients	2	4	6
Latency (ms)	0.14	0.195	0.32

## 4 Conclusion

An application level overlay for ubiquitous streaming of multimedia data is proposed. The self-organising and self-improving abilities of the overlay are accomplished through the monitoring of network dynamics. By adapting itself to the prevailing network conditions, better overlays are configured.

## References

1. S. E. Deering, Multicast Routing in a Datagram Internetwork, *PhD Thesis*, Stan. U. (1991).
2. M.H. Willebeek-LeMair, Bamba-Audio & Video Streaming over Internet, *IBM J. Res. Develop.*, vol. 42, no. 2, Mar (1998) 269-279.
3. Peter Parnes, et al, mSTAR: Enabling Collaborative Applications on the Internet, *IEEE Internet Computing*, Sep-Oct (2000) 32-39.
4. Peter Parnes, et al, Lightweight application level multicast tunnelling using mTunnel, *Computer Communications*, vol. 21 (1998) 1295-1301.
5. Ross Finlayson, Internet Draft: Describing Session Directories in SDP, <http://search.ietf.org/internet-drafts/draft-ietf-mmusic-sdp-directory-type-02.txt>
6. M. Kadansky, et. al., Reliable Multicast Transport Building Block: Tree Auto-Configuration, IETF RMT WG, draft-ietf-rmt-bb-tree-config-02.txt, 2 Mar (2001).
7. Y.H. Chu, S.G. Rao, S. Seshan, H. Zhang, Enabling Conferencing Applications on the Internet using an Overlay Multicast Arch., SIGCOMM, San Diego, California, Aug (2001).
8. P. Francis, Yoid: Extending the Internet Multicast Arch. ,<http://www.aciri.org/yoid> (2000).
9. Prim, R. C., Shortest connection networks and some generalizations, *Bell Sys. Tech Journal*, 36, (1957), 1389-1401.
10. T. Hansen, J. Otero, T. McGregor, H.W. Braun, Active Measurement Data Analysis Techniques, Int. Conf. On Communications in Computing, Las Vegas, Jun (2000) 105-135.
11. Internet Traffic Report, <http://www.internettrafficreport.com/index.html>.
12. Q. Sun, Sturman, D.C., A gossip-based reliable multicast for large-scale high-throughput applications, *Proc. Conf. Dependable Systems & Networks* (2000) 347 -358.