# CollectCast: A Tomography-Based Network Service for Peer-to-Peer Streaming

Mohamed Hefeeda, Ahsan Habib, Dongyan Xu, Bharat Bhargava

Computer Sciences Department, Purdue University

{mhefeeda, habib, dxu, bb}@cs.purdue.edu

http://www.cs.purdue.edu/homes/mhefeeda/promise.html

We propose a new network service (called CollectCast) suitable for highly dynamic environments such as peer-to-peer (P2P) systems. CollectCast operates entirely at the application level. Nonetheless, it does try to infer and exploit the properties of the underlying network. CollectCast has a pattern of "multiple senders and one receiver". CollectCast reflects the P2P philosophy of dynamically and opportunistically collecting the limited capacity of peers (in this case suppliers) to perform a task (streaming) traditionally performed by a dedicated entity (a media server).

The major functions of CollectCast are: (1) it infers and leverages the underlying network topology and performance information for the selection of suppliers. This is based on a novel application of several network tomography techniques; (2) it selects the "best" supplying peers that will likely yield the best quality for the streaming session; (3) it adaptively assigns rate and data to suppliers based on the capacity of each supplier and the current network conditions; and (4) it monitors the status of peers and connections and transparently switches suppliers so that the full quality is maintained throughout the session.

CollectCast is to be layered on top of a P2P substrate. A session in CollectCast is established as follows. First, CollectCast issues a lookup request to the P2P substrate, which returns a set of candidate peers who have the requested movie. CollectCast then constructs and annotates the topology connecting the candidate peers with the receiver. The topology is annotated with segment-wise loss rates and available bandwidth. Using the annotated topology, the suppliers selection algorithm determines the best active set from the candidate set. The rest of the candidates are kept in a standby set, from which replacement peers will substitute failed or degraded peers. Once the active set is determined, the receiver establishes parallel connections with all peers in the active set. Two connections are established with each peer: a UDP connection for sending the stream packets, and a TCP connection for sending control packets. The receiver assigns a sending rate to each of the active supplier based on the supplier's offered rate and the goodness of the path from that supplier to the receiver. The streaming session continues as far as there is no need to "switch" the active set. A switch is needed if a peer fails or the network path becomes congested. At which time, the topology is updated with new values measured *passively* during streaming and a new active set is selected and notified.

We briefly describe the selection problem and refer to [2] for the details of CollectCast and other issues such as the overhead imposed and the TCP-friendliness of the approach. The selection algorithm is given the annotated topology $\mathcal{T}$ (see Figure 1) that interconnects the candidate peers and the receiving peer. It then finds the set of active peers $\mathbb{P}^{actv} \subseteq \mathbb{P}$ that maximizes the expected aggregate rate at the receiver, provided that the receiver inbound bandwidth is not exceeded. The suppliers selection problem can be stated as: find $\mathbb{P}^{actv}$ that

$$\text{Maximize} \quad E\left[\sum_{p \in \mathbb{P}^{actv}} \boldsymbol{G}_p R_p\right] \quad (1)$$

$$\text{Subject to} \quad \alpha_l R_0 \leq \sum_{p \in \mathbb{P}^{actv}} R_p \leq \alpha_u R_0, \quad (2)$$
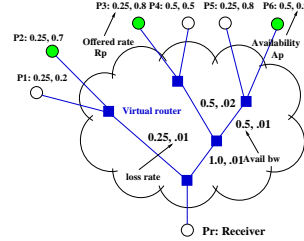


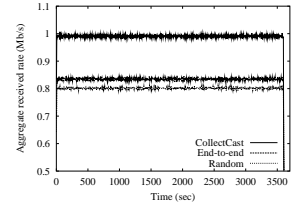**Figure 1: Topology-aware selection. P2, P3, P6 will likely provide the best quality.**



**Figure 2: Aggregate streaming rate at the receiver. CollectCast achieves higher rate and maintains the full quality.**

where $\boldsymbol{G}_p$ and $R_p$ are the *goodness* and offered rate of peer $p$, respectively, $\alpha_l, \alpha_u$ are two parameters used to control the level of loss tolerance in FEC and limit the required bandwidth by the receiver, and $R_0$ is the playback rate of the media file [2]. $\boldsymbol{G}_p$ indicates how good the peer $p$ is for the session being established. It is a function of the availability of $p$ and the quality of the network path from $p$ to the receiver. The quality of the network path is computed from the parameters (e.g., loss rate and available bandwidth) of the individual *segments* composing this path. Furthermore, the path quality depends on the number of peers sharing that path.

**Evaluation** A sample of our simulation results is shown in Figure 2. The figure shows that CollectCast achieves a steady aggregate rate of 1 Mb/s, which renders the full quality of playback. In contrast, selecting suppliers either randomly or based only on end-to-end information did not yield the desired quality.

**Applications** We have implemented and tested over the Internet a P2P media streaming system, called PROMISE [2]. PROMISE integrates CollectCast with a modified version of FreePastry (a P2P substrate). The modified version returns multiple suppliers for each request. PROMISE demonstrates the feasibility of streaming from multiple limited capacity, heterogeneous, and failure-prone peers. It also showed that significant gain in the quality of playback is achieved by using CollectCast. Finally, CollectCast and PROMISE show a concrete example of how tomography techniques can benefit network applications.

We are currently extending CollectCast to consider peers' social properties such as credibility and trustworthiness into the selection problem. Other network applications such as context-sensitive information dissemination and distribution of time-varying information items (e.g., stock quotes) are being developed on top of CollectCast [1].

## 1. REFERENCES

[1] Collaborative information sharing project. http://www.cs.purdue.edu/homes/bb/p2p.

[2] M. Hefeeda, A. Habib, B. Boyan, D. Xu, and B. Bhargava. PROMISE: A peer-to-peer media streaming system. In *ACM Multimedia 2003 Conference*, Berkeley, CA, USA, November 2003. To appear.