A PEER-TO-PEER OVERLAY NETWORK FOR REAL TIME VIDEO COMMUNICATION USING MULTIPLE PATHS

Andrea De Mauro[*], *Dan Schonfeld*

University of Illinois at Chicago
Electrical and Computer Engineering department
Chicago, IL, USA

Claudio Casetti

Politecnico di Torino
Dipartimento di Elettronica
Torino, Italy

ABSTRACT

This paper presents a peer-to-peer (P2P) service for the transmission of real-time video content, exploiting the contemporary usage of multiple network paths over the current Internet. Before starting the transmission, the sender probes the available paths for their Round Trip Time and other parameters using simple ICMP packets. Then it chooses the set of paths to be used in order to maximize the expected Signal to Noise Ratio (SNR) while meeting the delay requirements of the streaming content. The video is encoded using a Multiple Description Coding (MDC) technique. Simulations show that the proposed algorithm always chooses one of the group of paths that yields the best SNR.

1. INTRODUCTION

Real-time video distribution is becoming more and more popular among Internet users. It is characterized by stringent delay/loss requirements and still remains an open issue. Many studies have proposed to exploit the simultaneous utilization of multiple paths for relaying real-time packets from the sender to the receiver, i.e. multipath, in order to improve the error resilience. Many of them are based on MDC, [1], i.e. a source coding technique which encodes a signal into a number of separate bitstreams called descriptions. These are sent through different network paths to a destination. The receiver can play back the signal when at least one of the descriptions is received. The quality of the played back signal is proportional to the number of descriptions received. This property makes MDC highly suitable for lossy packet networks such as Internet.

In [2] the authors showed that minimizing the number of shared links among the paths results in a better quality of MDC video. [3] proposed another multi-path selection method for MDC-based streaming that takes into account other important aspects of network conditions such as bandwidth, packet loss probability, delay and jitter. The consequent path-selection algorithm attains some improvements over the maximally-disjoint method. Nevertheless, it is hardly viable because of the huge computational effort and the difficulty of retrieving the estimation of such network parameters.

Nowadays, Internet users are familiar with P2P systems, based on the philosophy of resources sharing. Skype, [4], is a proprietary P2P-based Voice over IP (VoIP) service that is meeting the approval of the public. With the recently released version 2.0, Skype also added video communication and became a noteworthy teleconferencing system. One of its most important features is that it lets users place calls even if they are behind firewalls or inside a Network Address Translation (NAT) area. In order to do that, each Skype user is part of an application-level P2P overlay network, [5]. An overlay network is a virtual network of nodes and logical links that is built on top of an existing network (the Internet, in our case) with the purpose of implementing a service that is not available in the infrastructure below. In this work, an overlay network similar to the one used in Skype is proposed with a different aim: obtaining multiple paths within the current Internet. Indeed, today the IP layer does not let the sender force a path to a receiver in order to provide Quality of Service (QoS), unless a specific route reservation policy is used (for instance IntServ with RSVP that, unfortunately, does not scale to the global Internet and was never endorsed by IETF).

2. MULTIPLE PATH NETWORKS

The idea presented in this paper is to use an overlay P2P network in order to set up an association (set of different paths) between the sender and the receiver of a streaming content, in order to minimize the distortion due to the unreliability of the service offered by IP (error-prone links, congestion of the nodes). If an overlay network is built in a P2P fashion, there is no full knowledge of the network below. An algorithm has to find the best relays in order to maximize the SNR, without relying on a centralized server nor on a complete knowledge of the network. With the present work we want to exploit the current Internet *as is*, without deploying a new infrastructure or updating the current routers' behavior. The idea is to build a P2P network where users can either be:

[*] Andrea De Mauro is with both the University of Illinois at Chicago and the Politecnico di Torino.

1-4244-0367-7/06/$20.00 ©2006 IEEE

- senders / receivers of real-time data flows;
- unaware relays of other peers' data flow.

The relays let the participants in a real-time communication exploit multiple paths by forwarding the packets from the sender to the receiver. Since it is a completely distributed protocol, the peer-to-peer network can grow without scaling efforts.

In order to implement such network, it is conceivable that a list of randomly selected collaborating peers (*friends*) is received from a fully-distributed P2P network, such as GNUtella [6]. Assuming that only two-hop paths are used (source-relay-destination) and a list of n friends scattered in the Internet is available, $n+1$ network paths could be used (n relayed paths plus one direct path from source to the destination, offered by IP as usual). Once the availability of a number of paths is consolidated, one of the main issues is to choose the "best couple" (supposing that we adopt a 2-description coding), i.e. the one that hopefully maximizes the SNR perceived at the receiver.

The main steps of the proposed protocol are:

1. by sending a series of RTT probes (similar to ICMP Echo packets used in the Ping application), some simple parameters are computed for each path and "eligible couples" are identified, i.e. couples of paths meeting the timing requirements of the streaming application;
2. an overall performance parameter (identified as proportional to the consequent SNR) is obtained for each eligible couple; the "best couple" will be the one with the highest performance parameter;
3. the streaming is performed by sending the two descriptions onto the two best couple of paths.

The performance obtained by the transmission will be affected by the level of "correlation" among the different paths. If r_{ν_i,ν_j} is the correlation coefficient, also known as the *Pearson's* r, between the distortion introduced by the paths i and j, it can be shown that:

$$\mathrm{SNR} = K \frac{N}{1 + \frac{1}{N} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} r_{\nu_i,\nu_j}}, \tag{1}$$

where SNR is the average Signal to Noise Ratio obtained after the decoding, N is the number of available paths and K is a positive constant depending on the overall condition of the network and on the effectiveness of the implemented coding technique. If we suppose that the correlation level is the same for each couple of paths and equal to r, we get:

$$\mathrm{SNR}\,(r, N) = K \frac{N}{1 + r(N-1)}\,. \tag{2}$$

This confirms the intuition that:

> **"in a transmission of a video through multiple paths the performance increases as the correlation among the paths decreases and the number of the paths increases."**

2.1. Correlation among paths

One of the aims of the initial probing phase is to estimate the correlation among the couples of paths in terms of the fraction of shared links out of the total. The higher the number of shared links, the more the paths are correlated, the smaller the expected SNR may be. In fact, if many links are shared, the probability to lose two descriptions of the same frame is higher because of the bursty nature of packet losses in congested nodes. Since the perceived distortion greatly depends on whether losses affect both descriptions of the same frame, different bitstreams should not go through the same nodes.

Let $T_{1,2}\,[n]\,, n = 0, 1, \ldots, N-1$ be the sequence of RTTs estimated each ΔT seconds for two different paths and $T'_{1,2}\,[n]\,, n = 0, 1, \ldots, N-2$ the numerical derivative of $T_{1,2}\,[n]$, computed as:

$$T'_{1,2}\,[i] = T_{1,2}\,[i+1] - T_{1,2}\,[i]\,, i = 0, 1, \ldots, N-2\,. \tag{3}$$

An RTT of a path could be seen as the double of the sum of the delays added by each link belonging to that path. If two paths have many links in common, many terms of this sum will be in common. In this case, an increase of the RTT for a path will likely be related to an increase of the RTT for the other path and vice versa. For this reason, we chose to estimate the correlation among the couples of paths by evaluating the statistical correlation between $T'_1\,[n]$ and $T'_2\,[n]$, as:

$$C = Cr(T'_1\,[n]\,, T'_2\,[n]) = \frac{Cov(T'_1\,[n]\,, T'_2\,[n])}{\sqrt{Var(T'_1\,[n])\,Var(T'_2\,[n])}}\,. \tag{4}$$

2.2. Network area between paths

Let us consider the triangle where the vertices are three nodes of a network (for instance the sender and two of its "friends") and the edges are given by the propagation delays (i.e., half the RTTs) among them. This is not a triangle in the Euclidean sense. For example, in this new "space of RTTs" a triangle can have an edge longer than the sum of the other two. In [7] the authors computed the sender-relays angle in order to estimate the length of the shared path, i.e. how much the two paths are joint. They follow the observation that a wider client-relays angle corresponds to a shorter shared path. In the present work we want to widen this concept, introducing the idea of "network area".

Consider the quadrilateral placed in the space of RTTs whose vertices are the sender (S), the receiver (D) and two friend-nodes ($\mathrm{R}_1, \mathrm{R}_2$). We can compute its area by summing the areas of the triangles relay-source-relay and relay-destination-relay, as shown in Figure 1.

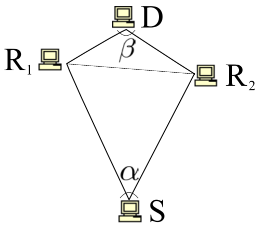

Fig. 1. The quadrilateral placed in the space of RTTs.

The angles α and β can be estimated using Carnot's theorem. For instance:

$$\cos\alpha = \frac{d(\mathrm{S},\mathrm{R}_1)^2 + d(\mathrm{S},\mathrm{R}_2)^2 - d(\mathrm{R}_1,\mathrm{R}_2)^2}{2\, d(\mathrm{S},\mathrm{R}_1)\, d(\mathrm{S},\mathrm{R}_2)}\,, \qquad (5)$$

where $d(a,b)$ is the average delay between nodes a and b. Since the "network" angles α and β are computed using Carnot's theorem that is based on triangles while the delays among peers could not be the edges of an Euclidean triangle, the inverse cosine is not computable as usual. Thus, in our computations, we supposed:

$$\alpha = \begin{cases} 0\,, & \cos(\alpha) > 1 \\ \mathrm{acos}(\cos\alpha)\,, & |\cos(\alpha)| < 1 \\ \pi\,, & \cos(\alpha) < -1\,. \end{cases} \qquad (6)$$

Now we can compute the area of the quadrilateral by using the well-known formula for the triangles' area:

$$S = \frac{d(\mathrm{S},\mathrm{R}_1)\, d(\mathrm{S},\mathrm{R}_2)\sin\alpha}{2} + \frac{d(\mathrm{R}_1,\mathrm{D})\, d(\mathrm{R}_2,\mathrm{D})\sin\beta}{2}\,. \qquad (7)$$

2.3. Packet loss probability

By simply counting the correctly received packets, N_r, and the sent ones, N_s, we can easily get an estimation of the Packet Loss Probability (PLP) for each eligible path as:

$$P = 1 - \frac{N_r}{N_s}\,. \qquad (8)$$

If a peer has a wireless access link, the path including it as relay will likely be characterized by an appreciable PLP, due to the radio channel and, so, it should be discouraged.

2.4. Overall performance parameter

Observing that:

- The greater the correlation among the derivatives of the delays, C, the worse the expected SNR because the paths are more likely shared.
- The greater the area, S, of the quadrilateral in the space of RTTs, the better the expected SNR because the paths are more likely separated.
- The greater the PLP, P, the more the path is to be discouraged, because of the presence of lossy links.

we can define the overall performance parameter as:

$$\gamma = k_1\, S - k_2\, C - k_3\, P\,, \qquad (9)$$

where the positive coefficients k_1, k_2 and k_3 should be properly tweaked, considering the importance and the relative magnitude of the three terms (in our simulations, for simplicity, $k_1 = k_2 = k_3 = 1$).

Equation (2) can also be interpreted (with minor modifications) to prove a general result: the lower the correlation (r) among multiple probes measuring the same quantity and the higher their number (N), the higher the overall measured performance (SNR). In our case, S, C and P are three different probes that estimate what the streaming quality will be. Experimental results showed that their values are uncorrelated and, so, we decided to use all of them for defining γ.

3. PROTOCOL OVERVIEW

The main steps for a streaming session are:

1. The sender is instructed about the collaborative peers through a bootstrap primitive similar to those used by GNUtella and Skype.
2. The sender starts to send a certain number of RTT probes each ΔT seconds to its friends: they will be relayed to the destination of the streaming session. In our simulations 60 batches of RTT probes have been sent, one each $\Delta T = 0.030$ s.
3. After a timeout expiration, the sender computes the average RTT for each path and decides which couples are *eligible*, verifying the following conditions:
    - The average delay between the arrival of a description and the other one cannot be too high because of the limited size of the receiver buffer. Knowing this size and making some statistical considerations based on the expected jitter, we can define `MaxDeltaRTT` such as, for an eligible couple (a, b):

      $$\left|\overline{\mathrm{RTT}(a) - \mathrm{RTT}(b)}\right| \leq \texttt{MaxDeltaRTT}\,,$$

      where $\mathrm{RTT}(x)$ is the overall RTT for the path x.
    - In real-time two-way interactive applications, the RTT between the participants should be limited, in order to avoid an annoying delay perception. Thus, another timing conditions for eligibility can be:

      $$\max(\mathrm{RTT}(a), \mathrm{RTT}(b)) \leq \texttt{MaxPathRTT}\,.$$

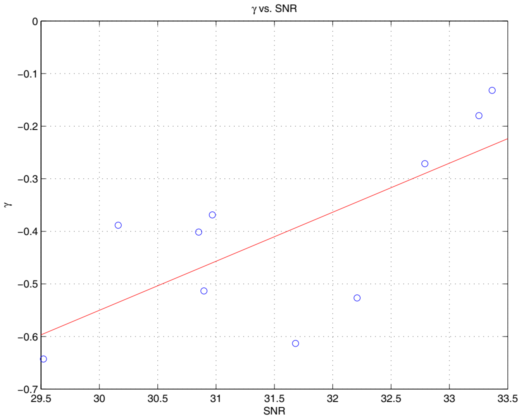

Fig. 2. Linear regression between γ and the SNR. Experimental points are in blue while the best-fit line is in red. Using the chosen couple (the one with the highest γ), the obtained SNR is better than using the other eligible couples. In this run, the average SNR gain is 2 dB and $Cr(\mathrm{SNR}, \gamma) = 0.71$.

4. The sender has to compute the correlation among the derivatives of the RTTs, C, only for the eligible couples.
5. The sender sends other RTT probes (20 in our case) to compute the distance between the "eligible" relays., i.e. the missing side of the triangle or, also, the diagonal of the quadrilateral.
6. After a timeout expiration, the sender can compute the area S and the performance parameter γ for each eligible couple and can create a list of eligible paths, sorted by γ.
7. Finally, the sender starts to send the video content using the eligible couple with the highest γ.

4. SIMULATION RESULTS

Our simulations have been performed by using the NS2 network simulator, [8]. We adopted a 2-level hierarchical network with 8 nodes at the first level and 37 nodes at the second one. Sender, receiver and all the other peers are second-level nodes. A loss model has been added to some peers' access links (modeled as wireless). All of the links have a bandwidth of 2.4 Mbps. The link delay is, on average, 30 ms at the first level and 15 ms at the second. We added FTP and UDP cross traffic with the ON/OFF periods having exponential and Pareto distributions.

A rough MDC technique (based on splitting odd/even frames) and a simple concealment algorithm (when a frame is not correctly decoded, the previous available good frame is repeated) have been implemented. Each description has been encoded using MPEG-4 and sent with a streaming rate of 15 fps. In these simulations we sent the *Carphone* QCIF video sequence. The whole probing phase lasted roughly 3.0 s for each run. Results (Figure 2) show that the proposed algorithm effectively increases the final SNR.

5. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a new framework for video streaming using path diversity and MDC that, exploiting the P2P paradigm:

- implements a real multipath scenario in the current Internet, deployable at any time by simply distributing new software, without modifying the infrastructure;
- chooses the right set of available paths in order to maximize the perceived performance, by sending a few RTT probes (Ping) without great costs in terms of bandwidth, by taking into account the timing requirements and by using a computationally light algorithm.

Other contributions of this paper are the introduction of the new concept of "Network Area" and a proof of the multi-path streaming usefulness, (2). Our next objectives include: exploring the possibility of using other network parameters with the aim of enhancing the robustness of the choice and implementing an adaptive algorithm that swaps to a better available couple in case of a sudden change in the path conditions.

6. REFERENCES

[1] S. Somsundaram, K. P. Subbalakshmi and R. N. Uma, "MDC and Path Diversity in Video Streaming," IEEE ICIP, 2004.

[2] J. Apostolopoulos, W. Tan, S. Wee, G. Wornell, "Modeling Path Diversity for Multiple Description Video Communication," IEEE ICASSP, 2002.

[3] Ali C. Begen, Y. Altunbasak, O. Ergun, M. H. Ammar, "Multi-path Selection for Multiple Description Video Streaming over Overlay Networks," Image Communication 20, 2005.

[4] Skype, <http://www.skype.com>.

[5] S. A. Baset, H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," IEEE INFOCOM, 2006.

[6] Gnutella Protocol Development Wiki, <http://www.the-gdf.org>.

[7] M. Guo, Qi. Zhang, W. Zhu, "Selecting Path-Diversified Servers in Content Distribution Networks," IEEE GLOBECOM, 2003.

[8] The Network Simulator ns-2, <http://www.isi.edu/nsnam/ns>.