Internet Media Streaming Using Network Coding and Path Diversity

Dong Nguyen, Tuan Tran, Tuan Pham, and Viet Le
School of Electrical Engineering and Computer Science
Oregon State University, Corvallis, OR 97331, USA
{nguyendo, trantu, pham, lev}@eecs.oregonstate.edu

Abstract—Delivering live media content over best-effort Internet is a challenging task due to a number of factors such as packet loss, delay, and bandwidth fluctuation. As a result, many approaches have been proposed ranging from Auto Repeat Request-type solutions, Forward Error Correcting code to routing and scheduling protocols and network architecture design. Multiple path streaming is an important solution which exploits the path diversity to satisfy the high transmission bandwidth requirement of streaming applications. Recently, network coding has been proposed for efficient information dissemination over packet-switched networks. In this paper, we propose a network coding framework for multiple path streaming over the Internet. In particular, we use a number of random network coding schemes in order to both exploit the benefits of path diversity and prevent packet loss. Our framework works at the application layer using the UDP service and is independent of the current network infrastructure.

Index Terms—Multimedia streaming, video streaming, network coding, multiple path streaming, path diversity.

I. INTRODUCTION

The last decade has witnessed a growing popularity of multimedia streaming applications over the Internet. There are thousands of Internet multimedia services such as online music, online radios and Internet television, and video-on-demand. Recently, several live Peer-to-Peer (P2P) streaming systems such as PPLive, CoolStreaming provide live video services to a larger number of users worldwide. In addition, Internet Service Providers also attempt to serve video-on-demand to their subscribers through their IPTV services. However, due to the inherent characteristics of media data including high bit rates, delay and loss sensitivity that are different from the normal data, providing high quality multimedia over the lossy Internet with limited and fluctuated bandwidth is still challenging. As such, there are many approaches to improving the quality of Internet media streaming applications. The goals of those approaches can be classified into two categories: (1) achieving reliable communications over lossy Internet; (2) exploiting the current Internet transport infrastructure (i.e., UDP/TCP/IP) in order to make it efficient for media streaming applications.

From reliable communication perspective, Auto Repeat Request (ARQ) or Forward Error Correcting (FEC) code are used to prevent packets from packet loss due to the Internet congestion. In the ARQ scheme, the lost packets are recovered by a retransmission mechanism. Streaming applications based on the TCP service are examples of this scheme. Using FEC, the sender transmits some redundant information in addition to the original data. For example, using a systematic Reed-Solomon code to protect $m$ source packets, $k$ redundant packets are formed and transmitted in addition to original packets. If the number of correctly-received packets is equal or greater than $m$ then all $m$ source packets can be recovered. FEC is usually used to prevent packet loss in unreliable UDP communication service.

From the Internet transport architecture perspective, new application-layer routing, scheduling algorithms and network architectures (i.e., application-layer overlay networks) have been designed. For example, P. A. Chou and M. Ziao [1] proposed a scheduling algorithm using Markov decision process to efficiently transport media data from a server to a client in a rate-distortion optimization way. P2P networks, which are application-layer overlay networks, are developed to run at the application layer to disseminate data to a large number of users. P2P networks works based on the current Internet transport protocols but use their own network architectures and their own scheduling and routing protocols.

Recently, multiple path streaming frameworks which are able to stream different subsets of media data over different paths as opposed to typical scenario where data travels along a single path have been proposed [2] [3]. Content Distribution Network (CDN) and Peer-to-Peer (P2P) streaming network can be viewed as multiple path streaming systems. In CDN such as Youtube, a media file is multiply stored in multiple servers. A client can download parts of file from several servers via a number of links to make up a whole file. This multiple downloading alleviates the bandwidth limitation when a single server is used. Similarly, in P2P streaming, media packets belong to a sequence can travel over different paths from the server, relaying over several peers before arriving at the destination to reconstruct the full media sequence. For instance, in Joost P2P streaming system, a peer concurrently connects with 6 to 10 other peers from different geographic locations to download a media stream [4]. The rapidly growing number of users of those systems proves their success for multimedia streaming in the current best-effort Internet infrastructure.

Inspired by such development, we propose a server-client multiple path streaming system using network coding. Our system is employed at the application layer using the UDP communication service. Our network coding technique works similarly to the FEC scheme to provide reliable communications over the unreliable UDP communication service, and at the same time, exploit the advantage of multiple path streaming. Our network coding frameworks are simple, yet provide significant improvement of the media quality.

Our paper is organized as follows. Section II presents some background on network coding and multiple path Internet streaming and the system model for multimedia streaming.
Network coding has been proved to be very efficient for information multicast in both wireline and wireless paradigms. Network coding even becomes robust in situations where the network topology is difficult to obtain or link failures are often congested at link WX; (b) With XOR network coding at node W, two packets can be sent to two sinks without congestion; (c) The same transmission efficiency can be achieved with random network coding.

In Section III, we state our Internet multimedia streaming problems. Section IV illustrate our three network coding schemes. We provide some NS2 simulation results in Section V and the conclusion in Section VI.

II. BACKGROUND AND SYSTEM MODEL

A. Network Coding

Network coding was firstly proposed by R. Ahlswede et al. [5] as a solution to efficiently utilize the network bandwidth. Compared to the traditional store and forward transmission, i.e., network nodes store and forward information without modification, the network coding solution allows network nodes to store, combine or encode data before forwarding to the outgoing links. As shown in an example in Fig. 1 in which source S wants to multicast two packets a and b to two sinks Y and Z. To avoid congestion when transmitting two packets a and b via link WX (Fig. 1a), node W can encode them into one packet as $a \oplus b$ (we call this combined packet as netcod packet or coded packet). Upon receiving this combined packet, sinks Y and Z are able to reconstruct both a and b by simple XOR operations (Fig. 1b). Random network coding which was introduced by T. Ho et al. is another form of network coding which is independent of the network topology [6]. As shown in Fig. 1c, node S generates a pair of random coefficients $\{a_1, b_1\}$ and encodes two packets as $a_1a + b_1b$ that is sent out on link ST. Similarly, another pair of random coefficients $\{\alpha_2, \beta_2\}$ is generated and used to form another encoded packet $\alpha_2a + \beta_2b$ that is sent out on link SU. Node T and U can forward the information to the outgoing links without modification. Node W, however, uses random coefficients to encode two coming packets $\alpha_1a + \beta_1b$ and $\alpha_2a + \beta_2b$ into one new coded packets $\alpha_3a + \beta_3b$ that is sent out on link WT. Now sinks Y receives two coded packets $\alpha_1a + \beta_1b$ and $\alpha_3a + \beta_3b$ and sink Z receives $\alpha_2a + \beta_2b$ and $\alpha_3a + \beta_3b$ from which, they can decode the original packets by solving the system of equations with unknowns a and b.

Network coding has been proved to be very efficient for information multicast in both wireline and wireless paradigms. For wireless networks, S. Chachulski et al. [8] proposed a network coding scheme using opportunistic routing for multi-hop wireless multicast. In this scheme, wireless nodes opportunistically receive, encode and forward packets to other nodes without using any centralized routing and queueing technique. The authors’ simulation system proved that network coding remarkably outperforms the current state-of-the-art routing schemes.

B. Internet Multiple path Media Streaming

There has been a significant amount of research dealing with multiple path streaming. In [2], [3], Apostolopoulos designed a multiple path streaming system using multiple state encoding. T. Nguyen and A. Zakhor [9], on the other hand, proposed an FEC framework for distributed video streaming in which several video sources stream a media sequence to a client. The authors proposed a Reed-Solomon code scheme to combat the packet loss problem and to efficiently take the advantage of path diversity. In another work, T. Nguyen and S. Cheung [10], set up a multiple TCP connection streaming framework that optimizes the transmission rates over several concurrent TCP connections to enhance the streaming throughput. In addition, there are many papers on designing live P2P streaming networks. The difficulty with designing live P2P streaming network is that it must not only be scalable with a large number of users but also be able to distribute the realtime media content to all users. D. A. Tran et al. [11] designed a live P2P streaming system that was able to meet such requirements.

The idea for using network coding for Internet multimedia streaming is still in its infancy. According to our understanding, there have been a few literatures on this topic. The most significant work is done by M. Wang and B. Li [12] who implemented a live P2P streaming system using the random network coding framework proposed by C. Gkantsidis and P. Rodriguez [7]. The author’s system called LAVA is able to enhance the quality of video applications that serve a large number of clients at the same time. Our work is different from this work in which our network coding framework is designed for server-client streaming. Because there are only one sender and one receiver, there are more options for designing the encoding and decoding process based on the characteristics of media data in order to maximize the media streaming performance.

C. Media System Model

We assume that the multimedia signal is segmented into independent equal-length segments or frames. Each frame is then encoded into L layers presented as a directed line graph as in Fig. 2. Layers are sent as same-size packets which have different degrees of importance in reconstructing the original media segment and are dependent on each other. The packet corresponding to the first layer has the largest degree of importance while the packet corresponding to the last layer has the smallest degree of importance. Hence, the packet order $P_1, P_2, ..., P_L$ of a frame is the decreasing order of importance. Packets belonging to a frame are assumed to have the same deadline D. Because of dependency among packets, $P_i$ can be
used for playback only when it is received before the deadline $D$ and all packets that it depends on $P_1, P_2, \ldots, P_{L-1}$ have been received correctly.

The importance of media packets can be quantified into amounts of distortion reduction contribution in to the media frame. Suppose that packet $P_i$ has a distortion reduction contribution $\triangle d_i$; $(0 < \triangle d_i < 1, \sum_{i=1}^{L} \triangle d_i = 1)$. Consequently, the quality of a reconstructed frame at a receiver can be measured by the normalized average distortion defined as

$$\theta = \sum_{i=1}^{L} \triangle d_i \prod_{j \leq i} (1 - \varepsilon_j).$$

where $\varepsilon_i$ is the probability that packet $P_i$ is received correctly. The term $\prod_{j \leq i} (1 - \varepsilon_j)$ expresses the dependency of packet $P_i$ on packets $P_1, P_2, \ldots, P_{i-1}$ by the directed line graph above. The normalized distortion being 1 means that no packets of the frame is received and being 0 means that all packets are received correctly before deadline and the frame is fully reconstructed.

For the transmission system we assume that the multiple paths is established between the server and the client. Those multiple paths, for example, can be several links via different paths between the server and the client and different packets belonging to a media frame are transmitted via different paths before reaching the client. Consider an example in Fig. 3b in which each media frame is divided into $K$ packets $P_1, P_2, \ldots, P_K$. Depending on which packet has been lost to retransmit when the transmission opportunities are available.

**Multiple path streaming:** In this scenario, there are multiple paths between the server and the client. Each frame has totally $N$ relay nodes to simultaneously relay packets from the server to the client. Those packets that are received correctly before deadline and the frame is fully reconstructed.

Consequently, packet $P_1$, even received correctly, becomes useless in playing back because it depends on $P_2$. Since the UDP service is used, the server does not know which packet has been lost to retransmit when the transmission opportunities are available.

**Multi path streaming:** In this scenario, there are multiple paths between the server and the client. Each frame has totally $N$ relay nodes to simultaneously relay packets from the server to the client. Consider an example in Fig. 3b in which each media frame is divided into $K$ packets $P_1, P_2, \ldots, P_K$. Depending on which packet has been lost to retransmit when the transmission opportunities are available.

**Multi path streaming:** In this scenario, there are multiple paths between the server and the client. Each frame has totally $N$ relay nodes to simultaneously relay packets from the server to the client. Consider an example in Fig. 3b in which each media frame is divided into $K$ packets $P_1, P_2, \ldots, P_K$. Depending on which packet has been lost to retransmit when the transmission opportunities are available.

**IV. NETWORK CODING SOLUTION**

Suppose there are $N$ transmission opportunities to transmit $L$ media packets $P_1, P_2, \ldots, P_L$ so that they can arrive at the clients before the deadline. The server employs network coding to encode those packets and send to the client. At the client, an appropriate decoding process is used to recover the original media packets. We have following network coding schemes:

**Scheme A (Uniform Network Coding):** This is the network coding scheme used in [12]. The server uniformly combines all packets of a media frame into coded packets $CP = \sum_{i=1}^{L} \alpha_i P_i$, where $\alpha_i, i = \{1, \ldots, L\}$, are elements randomly taken from a selected finite field. Because there are totally $N$ transmission opportunities on all links, the server forms $N$ netcod packets with different coefficients transmit them to the client. By encoding this way, the set of packets with different degrees of importance is mapped into a set of equally-important packets. The client must receive at least $L$ netcod packets in order to recover the media frame. Consequently, the media quality depends on how many packets are received, not on which packets are received as in the transmission without using any combination.

**Scheme B (Priority Network Coding):** Using network coding scheme B, the server gives priority to combine some important packets, instead of uniformly combining all packets as in Scheme A. For example, the server can combine the most $K$ ($K < L$) important packets $P_1, P_2, \ldots, P_K$ among $L$ packets as $CP = \sum_{i=1}^{K} \alpha_i P_i$. The server then dedicates all $N$ transmission opportunities to transmit those netcod packets. The client can recover those $K$ important packets whenever it receives enough $K$ coded packets. Clearly, this scheme is suitable for the situation with limited bandwidth, or high packet loss rate. Because the less important packets $P_{K+1}, \ldots, P_L$ are never combined into coded packets, so they are never received by the client. Thus this scheme does not produce the full quality of the media frame. However, it guarantees a level of media quality at client(s).

**Scheme C (Systematic Network Coding):** In this scheme, the server divides the transmission of $L$ packets into two phases.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.
At the first phase, the server uses $L$ transmission opportunities to plainly sends the source packets in the order $P_1, P_2, ..., P_L$ without using any network coding combination. After this phase, the server sends netcod packets if there are transmission opportunities available ($L < N$). Thus the server sends $CP = \sum_{i=1}^{L} \alpha_i P_i$ in the remaining transmission opportunities. As a result, the packets transmitted are $L$ source packets and $N - L$ netcod packets as $P_1, P_2, ..., P_L, CP_1, ..., CP_{N-L}$. Scheme C also shares a property with the scheme A in which if a client receives $L$ packets, it is able to recover $L$ source packets. An important advantage of this scheme compared to Scheme A is that a packet in the first transmission phase can be used immediately for playing back without having to wait until successfully decoding the whole frame. The systematic term represents that the source packets are sent followed by netcod packets similar to the systematic block code, e.g., Reed-Solomon systematic code, in which the server sends $L$ source packets and an additional number of parity packets. If any $L$ packets are received correctly, $L$ source packets are guaranteed to be decoded.

**How to send netcod packets:** As noted earlier, we assume that the transmission of media packets at the server over multiple paths are divided into transmission opportunities $t_0, t_1, ..., t_{N-1}$. Because network coding maps media packets with different priorities into packets of equal priorities, the server just sends packets via any path if it has a transmission opportunity. In other words, we do not need to any special transmission scheduling to send packets over multiple paths.

**Remarks on overhead:** There are two types of overhead in random network coding: transmission overhead and delay overhead. The earlier is the overhead in which random coefficients must be transmitted with the coded packets and the later is the playback time delay because the client has to wait until getting enough number of netcod packets for decoding and then takes time to decode the original packets. To combine $L$ packets of a frame, $L$ coefficients must be used to produce each netcod packet. As a result, the transmission overhead depends on the length of the packet being combined. In practise, coefficients are normally elements of GF(256) (i.e., one byte) and the packet has a length of more than thousand bytes, so this overhead can be ignored. The waiting time to get enough number of netcod packets are unavoidable for Scheme A and B. For example with $L = 5$ and $K = 3$, the client has to wait until getting enough 5 packets and 3 packets if using Scheme A and B, respectively, before starting decoding. Scheme C, however, does not have this delay. The decoding is a process of solving a system of equations. In Scheme A and B, the number of unknowns (in the system of equations) is $L$ and $K$, respectively, while in Scheme C, the number of unknowns is the number of lost packets. Normally, the Gaussian elimination method is used for decoding, hence the time complexity is $O(L^3)$ and $O(K^3)$ for Scheme A and B, respectively. If the packet length is $l$ symbols, then those complexities are $O(lL^3)$ and $O(lK^3)$. We note that $L$ is not very large since a media frame typically has up to more than 10 layers while $l$ is in the order of hundreds or more than one thousand.

Fig. 5 introduces the delay of the streaming system using three network coding schemes: Scheme A has the largest delay while Scheme C has the smallest delay.

### Decoding Complexity of Three Network Coding Schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Worst-case complexity</th>
<th>Best-case complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$O(lL^3)$</td>
<td>$O(lL^3)$</td>
</tr>
<tr>
<td>B</td>
<td>$O(lK^3)$</td>
<td>$O(lK^3)$</td>
</tr>
<tr>
<td>C</td>
<td>$O(L^3)$</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>

**V. SIMULATION RESULTS AND DISCUSSIONS**

To evaluate our network coding approach, we implement some NS2 simulations. We set up a network as in Fig. 6 in which a server $S$ streams a media sequence through routers $R_1, R_2, R_3$ to client $T$. Cross traffic generator $G$ is used to simulate the network congestion and packet drop at the routers. All links are set to have the same transmission bandwidth of 0.2Mbps. The server send out with the total rate of 0.6Mbps on each link. All transmissions in our simulation are based on UDP, thus there is no feedback from the client to the server. We assume each media frame having 5 packets $P_1, P_2, ..., P_5$ and their corresponding distortion reductions $\triangle d_1 = 0.5, \triangle d_2 = 0.3, \triangle d_3 = 0.15, \triangle d_4 = 0.10, \triangle d_5 = 0.05$ (in percentage of distortion, $\sum_{i=1}^{5} \triangle d_i =$...
1). We simulate the transmissions of a sequence with 500 frames and take the average distortion for every 10 frames. We evaluate the performances of the network coding schemes in comparison with the non-network coding scheme.

In the non-network coding scheme, the server sends media packets in a Round-Robin fashion. For example, if there are 7 transmission opportunities $t_0, t_1, \ldots, t_6 (N = 7)$, then the order of transmission is $P_1, P_2, \ldots, P_5, P_1, P_2$. The server does not know which packet is lost, so it prioritizes the transmissions of important packets.

Using Scheme A, the server combines all five packets into $CP = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3 + \alpha_4 P_4 + \alpha_5 P_5$ and transmits to the client at all $N$ transmission opportunities. There is no transmission priority given to important packets. Using Scheme B, the server combines the first four important packets $CP = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3 + \alpha_4 P_4$ and sends to the client. As presented above, this scheme gives priority to important packets in order to maintain a certain level of media quality under a limited bandwidth condition. Using Scheme C, the server transmits 5 source packets (without combining) using the first 5 transmission opportunities. In the remaining transmission opportunities, the server transmits the random network coding packets which are random combinations of five source packets.

Fig. 7 shows the simulation results of three network coding schemes and the non-network coding scheme with different $N$. We observe that, with $N = 5$ (Fig. 7a), Scheme A produces the most distorted media on average. This is because there are some durations with high packet loss rates that the client does not receive enough 5 netcod packets for decoding original media packets, resulting overall frame losses. However, Scheme B requires 4 netcod packets for decoding the first four important packets, so it provides higher quality than Scheme A in almost all frames. Scheme C gives the same media distortion as non-network coding scheme because, with $N = 5$, Scheme C has only one transmission phase for transmitting non-coded packets similarly to the non-network coding scheme.

When $N$ increases to 6 (Fig. 7b), Scheme A still performs worst. However, Scheme C now improves the media quality compared with the non-network coding scheme. It is plausible since there is the second phase of transmitting one netcod packet which helps to recover any one lost packet if any. From frame 200 to 500, Scheme B gives the highest media quality.

With higher number of transmission opportunities, the network coding schemes A and C enhances the media quality considerably. With $N = 7$ (Fig. 7c), Scheme A’s performance is slightly better than that of the non-network coding scheme because there are more opportunities so that the client can receive enough 5 netcod packets for decoding. Scheme B always yields the distortion of about 20% since the client always receive enough number of packets for decoding.

REFERENCES


