Efficient Online WiFi Delivery of Layered-Coding Media using Inter-layer Network Coding

Dimitrios Koutsonikolas, Y. Charlie Hu, Chih-Chun Wang, Mary Comer
Purdue University
{*dkoutson,ychu,comerm,chihw*}*@purdue.edu*

Amr Mahmoud Salem Mohamed
Qatar University
amrm@qu.edu.qa

***Abstract*—A primary challenge in multicasting video in a wireless LAN to multiple clients is to deal with the client diversity – clients may have different channel characteristics and hence receive different numbers of transmissions from the AP. A promising approach to overcome this problem is to combine multi-resolution (layered) video coding with inter-layer network coding. The fundamental challenge in such an approach is to determine the strategy of coding the packets across different layers that maximizes the number of decoded layers at all clients.**

This paper makes three contributions. (1) We first show that even for one client, the previously proposed canonical triangular scheme for inter-layer network coding can perform poorly. We show how to enhance the triangular scheme by incorporating the estimated target number of layers which significantly improves its effectiveness. (2) We show that such an enhanced triangular scheme still performs poorly for multiple clients with diverse channel characteristics, which motivates the need for searching for the optimal coding strategy. The naive way of searching for the optimal strategy is computationally prohibitive. We present several optimizations that drastically reduce the complexity of exhaustively searching for the optimal strategy, making it feasible in real time. (3) Finally, we design and evaluate an online video delivery scheme, Percy, to be deployed at a proxy behind the AP of a wireless LAN. Our simulation results show that Percy outperforms the previous inter-layer coding heuristic by up to 22-80% with varying numbers of clients.

I. Introduction

As both media content (e.g. youtube videos) over the Internet and wireless devices (e.g. smartphones) become increasingly popular, scalable delivery of rich media content over wireless links is quickly becoming one of the most important applications today.

The prevalent approach today to delivering a video stream to multiple receivers over WiFi is to unicast an independent stream to each receiver (e.g., [1]). This approach has two obvious drawbacks. First, it does not take advantage of the broadcast nature of the wireless communication and does not scale beyond a handful of receivers. Second, each receiver has to choose the best streaming rate that it can receive over the wireless link which is often not easy.

The alternative approach of broadcasting a single video stream to all receivers faces the following obvious dilemma. Different receivers may have different channel conditions and hence may experience different packet delivery ratios (PDRs) for a fixed broadcast bitrate from the sender. Broadcasting a video stream to all receivers would either require transmitting at the lowest bitrate so all receivers can receive the (low-quality) stream, or transmitting at other bitrates so some receivers can receive better quality stream while the worst receiver(s) receive none.

Yet another approach to dealing with receiver diversity is to exploit the well-known technology in video coding, multi-resolution coding (MRC) (also referred to as scalable or layered coding) [2], [3], [4], [5][1]. MRC was originally designed for the wired Internet, where receivers largely do not share the communication links. MRC divides the video into a base layer and multiple enhancement layers, so that the individual clients can independently decide how many layers to receive from the server according to their individual available bandwidth from the server. In a wireless network, however, all layers transmitted share the medium. Sending higher layers reduces the bandwidth available for sending lower layers. The problem is similar to that in the simple scheme of broadcasting a single video stream.

A promising approach to overcome the above client diversity problem in delivering media content over WiFi is to combine using MRC streams with inter-layer network coding (NC) to maximize the number of useful layers that can be retrieved by the wireless receivers. The fundamental reason that inter-layer coding improves the number of decoded layers even for a single receiver is that it allows retrieving useful layers from more combinations of received transmissions. Although a scheme without coding could potentially decode more layers by adjusting individual transmissions based on feedback after each transmission, sending such feedback is costly even for one client and quickly becomes impractical when there are multiple clients. Further, the above fundamental reason implies that inter-layer coding can also improve the case for multiple clients, which may have different combinations of received transmissions.

[1]Two approaches that have been proposed to provide error resilient video bit streams are MRC and multiple description coding (MDC) [6]. Although MDC has the advantage that any substream of the video that is received completely can be decoded, regardless of whether other substreams are received, MDC generally performs very poorly in terms of coding efficiency, i.e., an MDC coded stream requires significantly more bits to achieve a given quality than a single description stream.

Table I
COMPARISON OF DECODABLE USEFUL LAYERS WITH AND WITHOUT INTER-LAYER CODING. EACH OF u, v, w IS TRANSMITTED TWICE.

Reception outcomes containing 3 packets	Prob.	Useful layers retrieved	
		No Coding u=a v=b w=c	Coding u=a v=$Co(a,b)$ w=$Co(a,b,c)$
u,u,v	2/64	2	2
u,u,w	2/64	1	1
u,v,v	2/64	2	2
u,v,w	8/64	3	3
u,w,w	2/64	1	3
v,v,w	2/64	0	3
v,w,w	2/64	0	3
Exp. # of layers out of above 20 outcomes		1.80	2.60
Exp. # of layers out of all 64 outcomes		1.73	2.09

Motivating example. We use a simple toy-example to illustrate how inter-layer coding can improve the number of useful layers decoded at the client. Assume a video stream has 3 layers, with 1 packet/layer per frame. We denote the 3 packets in the frame as a, b, c, and analyze how coded transmissions can help with delivering these 3 packets. We assume the PDR is 0.5, and 6 transmissions can be made before the deadline of the frame. There are 64 reception outcomes, each with probability of 1/64. Table I lists all the reception events corresponding to the 20 outcomes where the client received 3 out of the 6 transmissions, and the number of useful layers that can be decoded under each event, without inter-layer coding, and under one sample inter-layer coding scheme. There are 2 outcomes included in each event except for u, v, w, which includes 8 outcomes. Under MRC, layer i is useful to the client only if the client has received all layers lower than i. For example, if the client received u, w, w, without inter-layer coding, the three receptions correspond to original packets a, c, c, and hence the client can only retrieve layer 1. With inter-layer coding, the three receptions correspond to packets $a, Co(a, b, c), Co(a, b, c)$, and hence it can decode all three layers. Table I shows inter-layer coding can improve the expected number of useful layers that can be decoded from 1.8 to 2.6 for the 20 reception outcomes.

The primary challenge in combining inter-layer coding with MRC is how to find the optimal inter-layer coding strategy for a given channel condition, determined by the number of transmissions the AP can send before the deadline of a set of frames, and the PDR at the receiver(s). Recent practical work [7], [8] that studied this approach have proposed and showed the effectiveness of an even canonical triangular heuristic which essentially evenly splits the total number of transmissions (that can be sent before the deadline) among the K ways of coding the packets, with the ith way being coding packets from the first i layers.

In this paper, we first show that even for one client, the previously proposed even canonical triangular scheme for inter-layer network coding can perform poorly. Intuitively, if the channel can only support delivering i layers, then the coding should concentrate in mixing packets from the first i layers and give up on higher layers. We show how to enhance the triangular scheme by incorporating the estimated target number of layers which significantly improves its effectiveness.

Second, we show that such an enhanced triangular scheme still performs poorly for multiple clients with diverse channel characteristics. This is because different clients have different target numbers of delivered layers, and the simple triangular scheme cannot easily accommodate the targets of all clients. This calls for the need for searching for an optimal coding strategy that simultaneously maximizes the layers that can be decoded at all clients, out of all possible coding strategies. We extend the simple triangular scheme to allow a variable number of transmissions for each of the K ways of coding a packet, denoted as the TriangularOpt scheme. The cost of searching all such strategies is computationally prohibitive. We present a set of optimizations that drastically reduce the complexity of searching for the optimal triangular strategy, making it feasible in real time.

Finally, we demonstrate the effectiveness of the new TriangularOpt scheme in the design and evaluation of an online video delivery scheme, Percy, deployed at a proxy behind the AP of a WLAN. The proxy in real time collects loss rates for different clients, searches for the optimal coding strategy, and generates coded packets for the AP to broadcast. Our simulation results show that Percy outperforms the enhanced triangular coding scheme by up to 22-80% for multiple clients with different channel characteristics.

II. RELATED WORK

There has been a significant amount of research in the area of video streaming over wireless networks, both in the video and systems community (see [9] for a summary).

Dynamic transcoding [10], [11], [12], [13] is a standard technique for enhancing the quality of video streaming. Initial work [10], [11] involved rate-distortion optimization techniques to adjust the rate of the transmitted video based on channel quality. Another set of work [12], [13] formulates a complex optimization problem to jointly optimize the video rate as well as the amount of FEC-based redundancy. Finally, [5] combines FEC with MRC and formulates an optimization problem to find the optimal per-layer rate allocation.

Multi-resolution coding combined with network coding has been studied in the early years of the development of network coding. Recent analytical results focus on sustaining the largest possible rates for the MRC video applications with intra-layer (e.g [14]) or inter-layer network coding, both centrally [15] and distributively [16]. However, most results have focused on the wireline networks (or they convert the

wireless network into its equivalent wireline counterpart), an approach which does not take into account randomness, one of the critical features of a wireless network.

Recently, there have also been a few practical works that demonstrated the effectiveness of combining MRC video streaming with variations of the canonical triangular coding[2] scheme for multihop wireless networks [7], [8]. [7] uses the triangular coding scheme combined with ARQ and a credit-based rate control system but it assumes that each client requests the desired number of layers from the server. In contrast, in our work, the AP automatically determines the optimal number of layers and the optimal coding strategy for any objective function. In [18], the authors present Multi-Generation Mixing, which is essentially equivalent to the triangular network coding scheme, and in [8], they apply this scheme to MRC video streaming. However, they assume a simple reliability mechanism which spreads a fixed amount of redundant coded packets equally among the K different ways of coding the packets. As we show in Section IV, this simple heuristic can perform poorly even for a single client.

Note that it is a classic subject to combine unequal error protection (over erasure channels) with video delivery mechanisms, including but not limited to the growth-code-based construction [19], [20], the priority encoding transmission that partitions the payload into different blocks [21], [22], and the LDPC staircase codes [23] that rely on Gaussian elimination for near-optimal decoding. Nonetheless, most of these works focus on the regime when the number of to-be-coded-together packets (also known as the codeword length, the symbol size, the generation size, etc.) is sufficiently large 500–1000. For comparison, our work focuses on practical settings, where the number of packets for each layer is usually a very small number 8–16. This special constraint on the number of to-be-mixed packets provides a unique challenge for network code design. Unlike the traditional average-based analysis that converts an erasure channel into its error-free wireline counterpart by the law of large numbers, when considering a very small number of packets, one has to take into account the randomness of the channel by direct probability computation. In a broad sense, our results can be viewed as an NC counterpart of the above works, with a new component that directly handles the channel randomness through probability computation. As we have seen in Table I, this turns out to be critical to the performance of the overall system.

We also note that NC-based 1-hop broadcast has been widely studied in theory. Some examples include [24] for throughput/delay analysis and [25] for stability and queue/buffer management. The above theoretic analyses assume *instant, per-packet feedback* from *all* destinations, which is not practical for the existing WiFi protocols. For comparison, our scheme does not rely on per-packet feedback and only uses feedback to estimate the PDR. As a result, it has a much lighter feedback overhead, which is critical to any practical wireless network protocols.

For the practical 1-hop WiFi network considered in this paper, it can actually be shown that, in the information-theoretic setting of broadcast erasure channels, the largest possible throughput achieved by inter-layer coding is no different from the largest possible throughput without coding [26], which can be proved by the physical channel degradation argument. However, the information-theoretic results assume infinitely large batch size (thus infinitely large delay). The large batch size essentially removes any channel randomness by the law of large numbers, and one thus only needs to focus on the mean-based first-order analysis. On the other hand, for practical wireless video streaming applications, the delay and communication overhead cannot be overlooked. Any practical schemes must use small batch sizes and the mean-based analysis and design lead to serious performance degradation in practice. This work directly addresses this issue by the probability-based analysis, which leads to substantial throughput improvement over the existing mean-based heuristics.

III. Preliminaries: Triangular Coding Scheme

In popular video coding schemes such as H.264/AVC, the video content is partitioned into sequences of pictures, referred to as groups of pictures (GOPs), each beginning with an independently decodable intra-coded picture. A typical duration for a GOP is 1-2 seconds. Each GOP contains many pictures or frames. A GOP is divided into a sequence of packets for delivery over the network. Although a single frame may span multiple packets, or a single packet may contain more than one frame, we can assume that there will be multiple packets for a GOP, and in the case of constant bitrate video coding, the number of packets per GOP will be constant throughout a sequence. In layered coding, the video content is partitioned into multiple layers of substreams, and hence each GOP can be thought of as consisting of several sequences of packets, one for each layer.

We focus on inter-layer network coding within each GOP. Let L be the number of layers (typically 2-6) and Q be the number packets per layer in a GOP. The value of Q depends on the streaming rate of the video. Since Q can potentially be large, we divide up the Q packets per layer per GOP into multiple segments, so that the number of packets per segment (per layer) N is on the order of 8. This ensures that even when we code the packets from segments from all layers, the total number of packets is on the order of 32 (e.g. for 4 layers), which will not result in high coding/decoding overhead. Let X be the total number of transmissions the AP can have within the deadline of frames corresponding to the $N \cdot L$ packets for the L layers.

[2][17] uses *inter-flow* network coding with video streaming, XORing together packets from multiple video streams to reduce the total number of transmissions; this approach is orthogonal to ours.

There are many coding strategies. Whenever we generate an inter-layer coded packet from layer i and layer j, we use random linear network coding to mix the N packets from layer i with the N packets from layer j. Since there are L original layers, for each of the X transmissions, there are 2^L ways of generating the inter-layer coded packet. Thus there are a total $(2^L)^X$ ways of coding the X packets. Each way or strategy can be represented as an $X \times L$ matrix consisting of binary elements, where the ith row is a vector of length L, representing how the ith packet is coded. For example, if the ith row is (1,0,1,1), it means the ith packet is coded by mixing original packets from Layers 1, 3, and 4.

We first observe that since the X transmissions are assumed to be independent Bernoulli trails, the ordering in sending individual packets does not matter. Hence, two strategies are equivalent if their matrix presentations are the same after row swapping. This suggests we just need to search among all the strategies that are not equivalent. Since there are only 2^L possible row vectors, or "bins", the total number of nonequivalent strategies is the same as the number of unique ways of assigning X transmissions to the 2^L bins, $\binom{X-1+2^L}{2^L-1}$. This is a drastic reduction from 2^{LX}.

Triangular coding schemes. Recent work [7], [18], [8], inspired by previous work on the classic subject of combining unequal error protection with video delivery mechanisms [21], [22], [23], have proposed and showed the effectiveness of a *canonical triangular scheme*. Canonical triangular schemes only consider the following L ways of coding packets from the L layers: the kth way being coding the first k layers, for $k = 1, ..., L$. [18], [8] consider an even canonical triangular scheme, perhaps the simplest one, denoted as $(\frac{X}{L}, ..., \frac{X}{L})$, which essentially *evenly* splits the total number of transmissions (that can be sent before the deadline) among the L ways of generating the coded packets.

In this paper, we consider all possible canonical triangular schemes, each of which is denoted as $(x_1, ..., x_L)$, where $\sum_{i=1}^{L} x_i = X$, and x_i denotes the number of packets that code the first i layers. There are a total of $\binom{X-1+L}{L-1}$ unique ways of assigning X packets to the L ways of generating the coded packets in a canonical triangular scheme. As we will show in the next section, the performance of the even canonical triangular scheme can be far from optimal even with a single client.

IV. Motivation

We compare the performance of the following 4 different canonical triangular coding schemes using the Glomosim simulator in delivering a 4-layer, 2.56 Mbps video stream with each 1-second GOP divided into 10 segments of $N = 8$ packets each per layer (see Section VIII for the simulation settings):

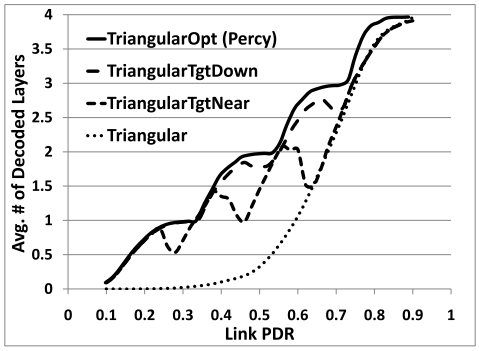


Figure 1. Performance comparison of different triangular schemes for a single client.

- **Triangular:** the scheme from [18], [8], denoted as $(\frac{X}{L}, ..., \frac{X}{L})$, which splits the X transmissions equally among the L canonical triangular ways of coding.
- **TriangularTgtNear:** denoted as $(\frac{X}{L_E}, ..., \frac{X}{L_E}, ...)$, which always splits the X transmissions equally among the L_E canonical triangular ways of coding, where $L_E = \frac{X \cdot p}{N}$, rounded to the nearest integer, is the expected (target) number of decoded layers.
- **TriangularTgtDown:** same as TriangularTgtNear, but with L_E always rounded down.
- **TriangularOpt:** the optimal way of assigning X transmissions among the L canonical triangular ways of coding.

Single client. We first consider the case with a single client. Figure 1 shows the average number of decoded layers under the different schemes. We make the following observations.

First, the performance of the Triangular heuristic degrades rapidly and drops to very low levels even for medium PDRs (0.4-0.5). Specifically, the average number of decoded layers drops below 1 for PDRS lower than 0.6 and is almost 0 for PDRs lower than 0.4. In contrast, the other three schemes deliver at least 1 layer with PDRs as low as 0.3. Hence, it is clearly counter-productive to code packets from more layers than the expected number of decoded layers, e.g., coding packets from 3 layers when the expected number of receptions is $2 \cdot N$ packets.

Second, we observe three PDR areas, starting at 0.24, 0.36, and 0.58, where the stream quality with TriangularTgtNear initially drops rapidly as the PDR increases and then it starts improving again. In these areas, the TriangularTgtNear scheme prematurely switches to sending $L_E + 1$ layers, but this choice results in delivering less than L_E. As an example, with a PDR of 0.64, TriangularTgtNear sends packets from all 4 layers, delivering only about 1.5 layers; in contrast, TriangularOpt conservatively sends only 3 layers and delivers 2.93. In these areas, TriangularTgtNear performs significantly worse than TriangularOpt.

Third, "rounding down" when calculating the target number of layers L_E is surprisingly effective; the performance difference between TriangularTgtDown and TriangularOpt

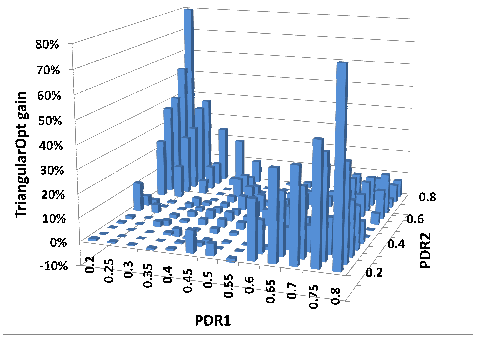


Figure 2. Gain of TriangularOpt (Percy) over TriangularTgtDown for two clients as their PDRs vary from 0.2 to 0.8 with a step of 0.05.

is always less than 18% and for most PDR values less than 10%. In other words, with a single client, carefully choosing the number of layers to transmit L_E is much more important than the coding strategy itself, and the simple coding strategy of equally splitting the total transmissions among the L_E triangular ways of coding often performs close to optimal.

Two clients. We now consider 2 clients. We assume the objective function is the sum of the decoded layers of all clients. Figure 2 plots the gain of TriangularOpt over TriangularTgtDown in terms of the total number of decoded layers for two clients, as their PDRs vary from 0.2 to 0.8 with a step of 0.05. For TriangularTgtDown, we adjust the estimation of the expected number of decoded layers to be based on the average of the 2 client PDRs.[3] Similar to the single-client case, we observe that the two schemes perform similarly in a large subset of PDR pairs. However, there are two areas, in the upper left and lower right corner (i.e., one client with a high PDR and the other one with a low PDR), where TriangularOpt clearly outperforms TriangularTgtDown, with the gain ranging from 20-80%. In these two areas, the *average* PDR over the two clients, used by TriangularTgtDown to estimate L_E, results in underestimation of L_E.

Summary. We have seen that the simple Triangular scheme which evenly splits X transmissions among the L triangular ways of coding often performs very poorly even with a single client. We have also showed how to enhance the triangular scheme by conservatively incorporating the estimated target number of layers which significantly improves its effectiveness. However, the performance of such a simple scheme can still be much lower than the performance of the *optimal* triangular scheme in the case of 2 clients with diverse channel characteristics (and hence, different target numbers of delivered layers), because it cannot easily accommodate the targets of all clients. This calls for the need for searching for an optimal coding strategy that simultaneously maximizes the layers that can be decoded at all clients, out of all possible coding strategies.

[3]Other ways of combining the PDRs are possible, though it is unclear whether there is a clear winner.

V. Efficient Search for the Optimal Triangular Scheme

In this section, we present a technique that efficiently searches for the optimal triangular scheme given a set of clients along with their PDRs to maximize any objective function, e.g., the sum of all layers retrieved at all clients. In Section VII, we will present an online protocol that tracks client PDRs and delivers MRC layers based on the calculated optimal triangular scheme.

In the following, we first present three optimizations that drastically reduce the complexity of scanning all canonical triangular schemes to be low enough to be performed online. We then construct a *strategy performance table* (SPT) which records the number of layers decoded under each triangular strategy for a range of PDRs. Such a table is then used to pick the strategy that optimizes any given objective function.

A. Reducing Inter-Layer Coding Strategies: Assignment resolution

The first optimization introduces the notion of coding resolution to reduce the effective number of triangular schemes in coding the X packets. Instead of considering all possible ways of group X packets into the L possible ways of coding (i.e., L "bins") in the canonical triangular scheme, we introduce a resolution parameter R, which controls the granularity of assigning X packets into the L bins. Let $Z = \frac{X}{R}$ be the number of groups of R packets. All packets in a group are assumed to be coded the same way. We only consider all the possible ways of coding all packets in each group. Since there are Z groups, it is easy to see there are a total of $\binom{Z-1+L}{L-1}$ unique ways of assigning the Z groups to the L bins, i.e., we only need to search $\binom{Z-1+L}{L-1}$ coding strategies to find the optimal strategy. Reducing the resolution, i.e., increasing R, can potentially lower the quality of the resulting optimal strategy.

B. Reducing the Number of Outcomes

Assume one client for now. Given a coding strategy for X transmissions and under a PDR p from the AP to the client, although the expected number of received packets is $X \cdot p$, since X is a usually a small integer, calculating the average number of decoded packets simply assuming $X \cdot p$ received packets can be fairly inaccurate, as we have shown in Table I. Hence, we need to enumerate and consider all 2^X possible outcomes in terms of the number of packets received at the client.

Our second optimization exploits the observation that for each canonical triangular scheme $(x_1, ..., x_L)$ for X transmissions and L layers, there are a total of $\prod_{i=1}^{L}(x_i+1)$ reception outcomes. Each outcome, denoted as $(y_1, ..., y_L)$

Table II
RUNNING TIME OF CONSTRUCTING THE STRATEGY PERFORMANCE TABLE.

Parameters (L, N, X, R)	# of Strategies	Time (sec.) (20 PDRs)
$(4, 8, 64, 16)$	35	0.001
$(4, 8, 64, 8)$	165	0.014
$(4, 8, 64, 4)$	969	0.127
$(4, 8, 64, 2)$	6545	0.999
$(4, 8, 64, 1)$	47905	7.984

represents that y_i packets are received out of the x_i packets which coded the first i layers, happens with a probability of $\prod_{i=1}^{L} \binom{x_i}{y_i} \cdot p^{y_i} \cdot (1-p)^{x_i - y_i}$. This optimization reduces the number of outcomes to consider for any given strategy from 2^X to $\prod_{i=1}^{L}(x_i + 1)$.

C. Avoiding Gaussian Elimination

Finally, since we focus on canonical triangular schemes, we show there is no need for Gaussian Elimination in calculating the number of layers that can be decoded for each reception outcome $(y_1, ..., y_L)$ for a given strategy $(x_1, ..., x_L)$.

Lemma 1: Under a canonical triangular coding scheme, for a reception outcome $(y_1, ..., y_L)$, the client can decode the first i layers (i.e. the N original packets in the first i layers) if $\sum_{j=i}^{i-k} y_j >= (k+1) \cdot N, \forall\ k \in [0, i-1]$.

For example, the client can decode the first 2 layers iff $y_2 >= N$ and $y_2 + y_1 >= 2 \cdot N$. The proof is straightforward: the condition implies that the rank of the matrix of the encoding coefficients of the received transmissions from coding any of the first i layers is at least $N \cdot i$ and hence Gaussian Elimination will be able to decode the first i layers. The complexity of calculating the number of useful decodable layers based on Lemma 1 is $O(L^2)$.

D. Summary and Running Time

Figure 3 lists the pseudocode for constructing a *strategy performance table* (SPT) after combining the three optimizations above. Such a table consists of the number of layers decoded for any PDR ranging from 0.05 to 1.0 with increments of 0.05, for all unique strategies under (L, N, X, R). The table is used to find the optimal strategy for a single client or for multiple clients with diverse PDRs under any objective function.

Table II shows the running time of the algorithm for a set of given parameters on a machine with a 2.5GHz AMD 2380 processor. We see that the running time for $(L, N, X, R) = (4, 8, 64, 4)$ is under 0.13 seconds.

E. Searching for the Optimal Strategy for Multiple Clients

In multicasting layered streams to multiple clients with diverse PDRs, the clients will not receive and decode the same number of layers. This is effectively a multi-objective optimization problem, where each client has a single objective of maximizing the layers it can decode. Hence,

§ CONSTRUCTING THE STRATEGY PERFORMANCE TABLE
Input: L, N, X, R, PDR resolution 0.05 (assume $L = 4$ for clarify)

```
Z = X/R;                          // the number groups to be assigned
for i1 = 0 to Z {                 // assigned to coding 1st layer (1,0,0,0)
for i2 = 0 to Z − i1 { // assigned to coding 1st 2 layers (1,1,0,0)
for i3 = 0 to Z − i1 − i2 {
for i4 = 0 to Z − i1 − i2 − i3 {
    if ((i1 + i2 + i3 + i4)! = Z) continue;
    strategy id si = ((i4 ∗ R + i3) ∗ R + i2) ∗ R + i1;
    for j1 = 0 to i1 · R { //reception outcome: j1 of i1 Tx. received
    for j2 = 0 to i2 · R {
    for j3 = 0 to i3 · R {
    for j4 = 0 to i4 · R {
        decoded = 0;
        if (j1 >= N) decoded = 1;
        if(j2 >= N && j1 + j2 >= 2N) decoded = 2;
        if (j3 >= N && j3 + j2 >= 2N &&
            j3 + j2 + j1 >= 3N)
            decoded = 3;
        if (j4 >= N && j4 + j3 >= 2N &&
            j4 + j3 + j2 >= 3N && j4 + j3 + j2 + j1 >= 4N)
            decoded = 4;
        for p = 0.05, 1.0, p += 0.05 {
            q = prob{receiving (j1, j2, j3, j4) out of (i1, i2, i3, i4)}
            table(si, p) = q · decoded;
}}}}}}}}}
```

Figure 3. Pseudocode for constructing the strategy performance table.

the optimal strategy depends on the aggregate objective function. We note that the choice of aggregate objective function is a policy question; different network operators may prefer different functions, and hence we leave the objective function as a configuration parameter.

For any given objective function, e.g., the sum of the layers that can be retrieved at each client, the proxy scans through all the coding strategies in the SPT, and finds the one that maximizes the objective function for the set of clients, based on their PDRs. The running time is insignificant as the complexity of scanning is linear.

F. Extension to Varying Numbers of Packets per Layer

In MRC video encoding, it is possible that different layers are encoded at different bitrates, and hence there can be different number of packets per layer in each coding segment. Our algorithm for constructing the strategy performance table shown in Figure 3 can be trivially extended to handle this. In particular, we just need to change the four **if** statements in the loop body for determining the number of layers that can be decoded. For example, if there are N_1, N_2, N_3, N_4 packets in the original layers 1, 2, 3, 4, the condition for decoding the first 2 layers is $(j2 >= N_2\ \&\&\ (j2 + j1) >= N_2 + N_1)$.

VI. ANALYSIS OF TRIANGULAR OPTIMAL STRATEGIES

Using the optimized algorithm for constructing the SPT, we analyze the distribution of the qualities of all the triangular strategies. We also consider a non-triangular scheme **No-Coding** which always transmits $\frac{X}{L_E}$ intra-layer (FEC) coded packets for each of the L_E expected layers, i.e., there is no inter-layer coding. Each receiver can decode

Table III
COMPARISON OF DIFFERENT STRATEGIES, $(L, N, X, R) = (4, 8, 64, 4)$. TRIANGULARTGTNEAR AND TRIANGULARTGTDOWN ESTIMATE THE SAME L_E FOR ALL 3 PDRS.

Strategy/Avg. decoded layers	P=0.3	P=0.5	P=0.8
TriangularOpt	(20,44,0,0)/1.6891	(12,16,36,0)/2.9509 (12,12,40,0)/2.9509 (12,0,52,0)/2.9509 (12,4,48,0)/2.9509 (12,8,44,0)/2.9509	(0,0,0,64)/4 (0,0,16,48)/4 (0,0,32,32)/4 (0,16,0,48)/4 (0,16,16,32)/4
TriangularTgtNear/TriangularTgtDown	(32,32,0,0)/1.6320	(16,16,16,16)/2.2751	(16,16,16,16)/3.9985
Triangular	(16,16,16,16)/0.0293	(16,16,16,16)/2.2751	(16,16,16,16)/3.9985
No-Coding	(32,32,0,0)/1.4095	(16,16,16,16)/1.2981	(16,16,16,16)/3.9853

N original packets for layer i if it receives N or more intra-layer coded packets for that layer. We will overload the notation for canonical triangular coding schemes here to denote the details of No-Coding. For example, No-Coding $(x_1, x_2, 0, 0)$ denotes sending x_1 and x_2 transmissions on intra-layer coded packets from layers 1 and 2, respectively.

A. Performance of Different Strategies

We consider $(L, N, R) = (4, 8, 4)$, and fix the total number of transmissions before the deadline to be $X = 64$. With $R = 4$, $Z = \frac{X}{R} = 16$, and hence there are a total of $\binom{Z-1+L}{L-1} = \binom{19}{3} = 969$ inter-layer coding strategies to be searched.

We first consider the PDR from the AP to the client to be $p = 0.5$. In this case, the expected number of packets received by the client is $X \cdot p = N \cdot L = 32$, and hence it appears the client should be able to decode all 32 original packets (and hence 4 layers) under the right inter-layer coding.

Figure 4 shows the number of original useful layers that can be decoded under the 969 strategies for $p = 0.5$. We observe that across all 969 strategies, the expected number of decoded layers never drops below 1, is less than 2 for 371 strategies, is about 2 for 17 strategies, and is more than 2 for 581 strategies. Somewhat surprisingly, the maximum number of decoded layers is bounded by 3, i.e., the best strategy cannot deliver more than $L - 1$ layers.

It is easy to understand with $X = 64$ transmissions and $p = 0.5$, any strategy should be able to achieve at least the basic layer, i.e., decode the original 8 packets from Layer 1. For example, the simplest strategy, which transmits all 64 packets from intra-layer coding of original packets from Layer 1, easily achieves this, as under $p = 0.5$, the probability 8 or more transmissions out of 64 are received is (1-1.81907E-11).

To understand why it is difficult to achieve more than 3 layers, we first look at the two optimal strategies. Table III shows 5 strategies $(12, 16, 36, 0)$, $(12, 12, 40, 0)$, $(12, 0, 52, 0)$, $(12, 4, 48, 0)$, $(12, 8, 44, 0)$ all achieve the most expected decoded layers 2.95. It is clear that all these

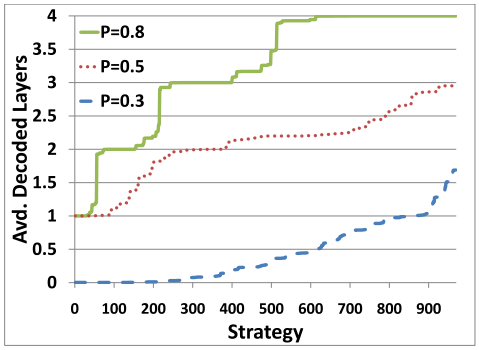


Figure 4. CDF of number of decoded layers over all strategies under $(L, N, X, R) = (4, 8, 64, 4)$ for varying PDRs.

strategies simply give up on spending any transmissions on packets coding all 4 layers.

Table III also shows that the TriangularTgt schemes, which in this example are the same as the Triangular heuristic $(16, 16, 16, 16)$, i.e., it also attempts to code packets from all 4 layers, could only decode on average 2.28 layers, again much lower than 2.95. Again this can be explained by looking closely at the binomial distribution of the reception outcomes. For example, to be able to decode all 4 layers, it would require in total at least 32 transmissions to be received, at least 8 coded original packets from 4 layers, at least 16 coded original packets from 3 or 4 layers, at least 24 coded original packets from 2, 3, or 4 layers. The probability for the reception outcomes that satisfy the above is only 0.104442. We thus see that there can be even scenarios where the performance of the usually very efficient TriangularTgtDown scheme can be far from optimal even for a single client.

We next consider the PDR from the AP to the client to vary from 0.3 to 0.8 while fixing all other parameters. In this case, the expected number of packets received by the client, $X \cdot p$, can be lower or higher than the number of expected received transmissions to decode all 4 layers, 32.

Figure 4 shows varying the PDR (the expected number of receptions) can significantly affect the number of decoded layers. When the number of expected received transmissions

Table IV
COMPARISON OF DIFFERENT OPTIMAL TRIANGULAR STRATEGIES WITH A FIXED NUMBER OF SLOTS, $X = 64$ AND VARYING RESOLUTION $R = 16, 8, 4, 2, 1$.

Strategy/Avg. decoded layers	**P=0.25**	**P=0.4**	**P=0.5**	**P=0.8**
R=16	(32,32,0,0)/1.1154	(16,16,32,0)/2.1138	(0,0,64,0)/2.9509 (0,16,48,0)/2.9509	(0,0,0,64)/4 (0,0,16,48)/4 (0,0,32,32)/4 (0,16,0,48)/4 (0,16,16,32)/4
R=8	(32,32,0,0)/1.1154	(16,24,24,0)/2.1252	(8,16,40,0)/2.9509	(0,0,0,64)/4
R=4	(28,36,0,0)/1.1193	(16,24,24,0)/2.1252	(12,16,36,0)/2.9509	(0,0,0,64)/4
R=2	(30,34,0,0)/1.1199	(16,24,24,0)/2.1252	(12,18,34,0)/2.9509	(0,0,0,64)/4
R=1	(29,35,0,0)/1.1201	(15,24,25,0)/2.1257	(12,18,34,0)/2.9509	(0,0,0,64)/4

is more than enough (for PDR > 0.5), many strategies perform well. For $p = 0.8$, 162 out of 969 strategies deliver on average all 4 layers and 195 more strategies deliver at least 3.99 layers. In contrast, when the number of expected received transmissions is less than enough (for PDR < 0.5), a large number of strategies fail to deliver even the basic layer and even the best strategy delivers only a small number of layers (1.69 with $p = 0.3$). From Table III we see again the importance of only coding the target number of L_E layers under a low PDR: TriangularTgtNear/Down delivers 1.63 layers with with $p = 0.3$, very close to TriangularOpt, while Triangular delivers only 0.02 layers.

B. Varying the Resolution

We now vary the resolution R to study its impact on the quality of the resulting chosen optimal strategy. Table IV lists the specific optimal strategies and the corresponding number of layers decoded for $(L, N, X) = (4, 8, 64)$ with varying $R = 16, 8, 4, 2, 1$, and varying $p = 0.25, 0.5, 0.8$. While increasing the resolution increases the number of triangular strategies, we observe that the number of decoded layers under the optimal strategies does not change significantly, and it is almost identical for $R \geq 4$ for all 4 PDRs. The intuitive reason for this again comes from the binomial distribution of the reception outcomes and the corresponding number of layers that can be decoded.

While increasing the resolution in searching for the optimal strategies does not seem to improve the quality of the optimal strategy for a single client, i.e., with a particular PDR, it does offer more diversity in the strategy space; it gives more unique strategies that achieve any particular number of decoded layers, and it gives strategies that achieve a particular number of decoded layers not achieved under lower resolutions. This diversity could potentially help to improve the aggregate number of decoded layers when there are multiple clients with diverse PDRs. We examine this in the next section.

C. Multiple Clients

We consider two clients, C_1, C_2, with varying PDRs p_1, p_2, respectively. We assume the objective function is the sum of the numbers of useful layers decoded by the two clients.

Table V
COMPARISON OF DIFFERENT STRATEGIES FOR 2 CLIENTS C_1, C_2 WITH DIFFERENT PDRS.

	($p_1, p2$)=(0.3,0.4)		
	Dec. Layers		
Strategy	C_1	C_2	$C_1 + C_2$
Max(sum), R=16 (16,48,0,0)	1.6879	1.9919	3.6798
Max(sum), R=8 (16,48,0,0)	1.6879	1.9919	3.6798
Max(sum), R=4 (20,44,0,0)	1.6891	1.9919	3.6810
Max(sum), R=2 (22,42,0,0)	1.6894	1.9917	3.6811
Max(sum), R=1 (21,43,0,0)	1.6893	1.9918	3.6811
Max(C_1) (20,44,0,0)	1.6891	1.9919	3.6810
Max(C_2) (16,24,24,0)	0.4623	2.1252	2.5875
	(p_1, p_2)=(0.3,0.8)		
	Dec. Layers		
Strategy	C_1	C_2	$C_1 + C_2$
Max(sum), R=16 (32,0,16,16)	0.7882	3.4763	4.2645
Max(sum), R=8 (24,8,0,32)	0.4355	3.9970	4.4335
Max(sum), R=4 (28,8,12,16)	0.6353	3.9458	4.5811
Max(sum), R=2 (28,8,12,16)	0.6353	3.9458	4.5811
Max(sum), R=1 (28,9,11,16)	0.6355	3.946	4.5815
Max(C_1) (20,44,0,0)	1.6891	2.000	3.6891
Max(C_2) (0,0,0,64)	0.0025	4.000	4.0025
	(p_1, p_2)=(0.5,0.8)		
	Dec. Layers		
Strategy	C_1	C_2	$C_1 + C_2$
Max(sum), R=16 (0,32,16,16)	2.3196	3.9985	6.3181
Max(sum), R=8 (12,18,24,10)	2.3197	3.9985	6.3182
Max(sum), R=4 (12,16,24,12)	2.4499	3.9274	6.3773
Max(sum), R=2 (12,18,22,12)	2.4522	3.9274	6.3796
Max(sum), R=1 (12,18,21,13)	2.4181	3.9700	6.3881
Max(C_1) (12,16,36,0)	2.9509	3.0000	5.9509
Max(C_2) (0,0,0,64)	2.1987	4.000	6.1987

We first evaluate if increasing the resolution R can improve the sum of decoded layers for the two clients. Table V shows performance of the optimal triangular strategies, listed as **Max(sum)**, for a set of PDR pairs as R varies from 16, 8, 4, 2, to 1. We observe the sum of the layers decoded by the two clients, listed under $C_1 + C_2$, barely improves when R is smaller than 4. Hence we conclude that $R = 4$ offers sufficiently fine granularity in searching for the optimal coding strategies, and will be used in the rest of the paper.

Next, we use the strategy table we constructed to show how each of all the possible strategies, including the optimal strategy, balances the decoded layers for the two clients (e.g. PDRs). From Table V, we observe that, when both clients have low PDRs, (0.3, 0.4), and hence the expected number of received transmissions $X \cdot p$ is less than enough to decode all 4 layers ($L \cdot N$), the strategy that maximizes the sum of the decoded layers for the two clients also maximizes the number of decoded layers (or achieves performance very close to the optimal) for the worse client. On the other hand, when one of the clients has a high PDR to receive enough transmissions ($L \cdot P$ or more), the above does not hold. The reason is that the optimal strategies for low-PDR clients generally avoid sending packets mixing all L layers, but such strategies would limit the number of decoded layers for the high-PDR clients and hence reduce the sum. For example, from Table III, we saw that the optimal strategies for $p = 0.5$ code only from the first 3 layers, bounding the sum for two clients to < 6 layers. Instead, by sending 12 packets that mix all 4 layers (see Table V for (0.5,0.8)), a total of 6.38 layers can be obtained.

Also shown in Table V in the rows labeled "Max(C_1)" and "Max(C_2)" are the number of decoded layers for the two clients if using the optimal triangular strategy that optimize the decoded layers for client 1 or client 2, respectively. As expected, these strategies underperform "Max(Sum)".

VII. Percy: Online WiFi Multicast of Layered Media

In this section, we present an online system, Percy, that exploits optimal inter-layer coding, for streaming layered media in a WLAN. The system assumes an unchanged media server, and is implemented in a media-aware Percy proxy behind the AP which performs inter-layer NC before relaying the coded packets to an unchanged 802.11 AP for broadcast transmissions. The clients are modified to decode NC packets before passing the packets up the network stack.

The clients interested in the same stream (e.g. a scheduled live stream) send requests to the server for that stream. The proxy intercepts and processes these requests, and groups these clients interested in a particular stream into one "virtual" multicast group. It then forwards one request to the streaming server, asking for all layers. We assume there is enough bandwidth between the proxy and the server for all layers to be delivered to the proxy. The server starts the initialization phase, during which it sends back control messages containing a set of parameters including the streaming rate, the duration of a GOP, and the number of layers used. This control information is also intercepted by the proxy before it is broadcast to the clients. After this control information exchange, the server starts sending the video frames.

The proxy (1) intercepts all IP packets corresponding to the video frames of the stream and buffers a segment of them, (2) estimates X and PDRs and calculates optimal inter-layer coding strategies, and (3) forwards to the AP coded packets for each segment.

PDR feedback from clients. The AP transmits each packet it receives from the proxy using 802.11 broadcast. The clients periodically send feedback to the proxy to allow it to obtain an estimate of their PDRs. We use a lightweight scheme in which each client reports every 200 ms the *total* number of packets since the last report. These feedback messages are forwarded by the AP to the proxy.

Online Estimation of X and PDRs. The proxy (1) continuously monitors the number of transmissions X' it can make in each GOP. The total transmission X' is divided equally among the segments constituting the GOP, i.e., X per segment; and (2) receives the periodic PDR feedbacks from each client, which are sent back to the proxy in fixed intervals during every GOP.

At the end of GOP i, the proxy uses the measured X and PDRs as the predicted values for GOP $i+1$, to calculate an SPT that lists the number of layers decoded for all possible strategies for the given L and X, using resolution $R = 4$, for all PDRs ranging from 5% to 100% with increments of 5%. As shown in Section V-D, this calculation can be finished in less than 0.13 sec for typical values of (L, N, X, R), e.g. (4,8,64,4).[4] During the first GOP, since there is no estimate of X and PDRs, the proxy simply transmits intra-layer coded packets from Layer 1.

Calculating the optimal coding strategy. As we mentioned in Section V-E, the optimal strategy in the case of multiple clients depends on the aggregate objective function, which is a policy question. For any given objective function, e.g., the sum of the layers that can be retrieved at each client, the proxy scans through all the coding strategies in the SPT, and finds the one that maximizes the objective function for the set of clients, based on their PDRs. This strategy is then used for all the segments consisting the next GOP.

Overhead control. As discussed in Section V, to limit the complexity of coding/decoding operations and header overhead from including the coding coefficients in each packet header, we use $N = 8$ packets per layer per coding segment which allows up to $L \cdot N$ packets to be coded together with low overhead for typical L values.

Table VI summarizes all the parameters. The values for the parameters come either from the video encoding (e.g. GOP, L), or from the Percy design (e.g. N, R), or from online measurement (e.g. X, p).

VIII. Evaluation

Methodology. We used the Glomosim simulator [27]. We placed an AP in the center of the simulation area and the

[4] In practice, the prediction is offset by one feedback interval from the GOP boundary. The proxy starts calculating the new SPT after receiving 4 feedbacks into a 1-sec GOP, i.e., leaving about 200ms till the start of the next GOP.

Table VI
PERCY SYSTEM PARAMETERS.

Parameter	Symbol	From	Value used in evaluation
GOP duration	GOP	video	1 sec
# of layers	L	video	4
Segment size/layer	N	design	8
Tx budget/segment	X	measurement	N/A
PDR	p	measurement	N/A
Resolution	R	design	4

clients uniformly on a circle around the AP. To evaluate the performance of the protocols under different loss scenarios, the clients were placed close to the AP and we generated link loss rates in a controlled manner, by artificially dropping packets at each client following a Bernoulli model.

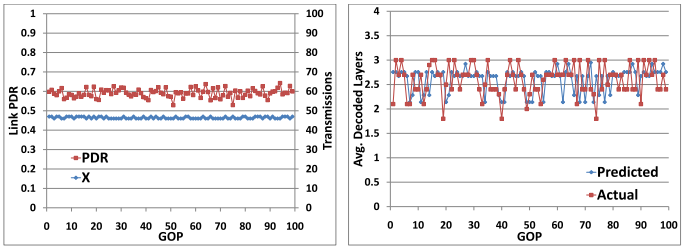
We used the 802.11 MAC layer with a fixed bitrate of 5.5Mbps and RTS/CTS disabled, as in most operational networks. Data packets were broadcast at the MAC layer. The feedback messages sent by Percy clients were unicast at the MAC layer for increased reliability.

The video stream was a constant bit rate (CBR) traffic over UDP at 2.56 Mbps for a duration of 100 sec. The GOP duration was set to 1 sec. The stream consisted of $L = 4$ layers. Each layer included 80 1000-byte packets and was divided into 10 segments of $N = 8$ packets each.

A. Evaluation for a Single Client

We have already shown the performance comparison among different triangular schemes in Figure 1. Here, we take a closer look at Percy's online algorithm. Figure 5(a) plots the estimated link PDR (based on the periodic reports from the client) and the measured number of available packet transmissions per segment X for each GOP for a stream data rate of 2.56Mbps and an actual link PDR of 0.6. Remember that the PDR and X values at the end of the i-th GOP are used to construct the table and choose the optimal coding strategy for the $(i + 1)$-th GOP. We observe that both the PDR and the number of transmissions X remain quite stable over time. The PDR varies from 0.52-0.64; the number of transmissions X is even more stable taking only two values, 46 and 47 (the actual number of transmissions over the whole GOP exhibits a higher variation but we always round it up to a value divisible by the number of segments, here 10, in order to assign the same X to every segment of a given GOP and calculate the table only once).

Figure 5(b) compares the predicted number of decoded layers for each GOP as given by the constructed table at the end of the previous GOP and the actual number of decoded layers at the client. We observe that the actual number of decoded layers can be higher or lower than the predicted one with almost equal probability (it is lower in 52 out of 99 GOPs and higher in the remaining 47 ones). In the long term, the actual number is almost equal to the predicted one – the average predicted value over 99 GOPs is 2.60 and the actual number 2.59. This shows that the online PDR-aware heuristic used in Percy is very efficient in practice.



(a) Measured PDR and number of transmissions per segment (X).

(b) Predicted and actual number of decoded layers.

Figure 5. Percy dynamics per GOP over 100 GOPs for a 2.56Mbps stream and PDR = 0.6.

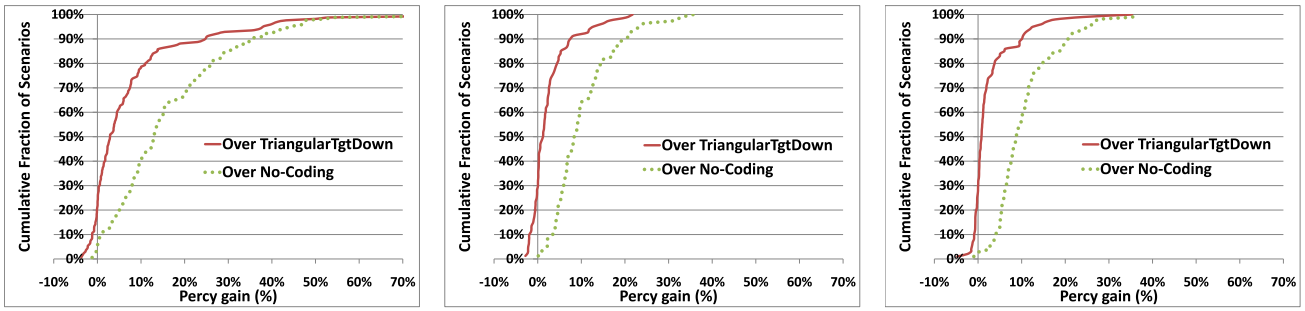
B. Evaluation for Multiple Clients

We have seen from Figure 2 that, in the case of 2 clients, Percy outperforms TriangularTgtDown by as much as 80%. Here, we compare the performance of Percy against TriangularTgtDown and No-Coding when the number of clients varies from 2 to 6. For TriangularTgtDown and No-Coding, we adjust the estimation of the target number of decoded layers to be based on the average of the client PDRs. For each case, we ran 100 different simulation scenarios; in each scenario the client PDRs are chosen uniformly randomly from the range [0.2, 0.8]. For each scenario, we calculated the gain of Percy over the other two schemes in terms of the total number of decoded layers for all clients. Figures 6(a), 6(b), 6(c) plot the cumulative distribution functions (CDF) of the gain of Percy over the other two schemes in 100 scenarios with 2, 4, and 6 clients, respectively.

Similar to the single-client case, TriangularTgtDown outperforms No-Coding. Percy performs similarly to TriangularTgtDown in about 60% of the simulated scenarios (the performance difference is less than $\pm 5\%$) and outperforms it in the remaining scenarios, with the gain over TriangularTgtDown being up to 80%, 22%, and 35%, with 2, 4, and 6 clients, respectively. Compared to No-Coding, Percy's performance is superior in almost all scenarios and the median gain over No-Coding is higher than over TriangularTgtDown. Generally, the gain of Percy over the TriangularTgtDown scheme is higher for scenarios including a mixture of clients with high and low PDRs; in these cases, TriangularTgtDown cannot easily accommodate the target numbers of layers for all clients.

IX. CONCLUSION

In this paper, we presented the design and evaluation of Percy, an online video delivery scheme, to be deployed at a proxy behind the AP of a wireless LAN. The proxy in



(a) 2 clients. (b) 4 clients. (c) 6 clients.

Figure 6. CDFs of the gain of Percy over TriangularTgtDown and No-Coding in 100 different scenarios with 2, 4, and 6 clients.

real time collects loss rates for different clients, searches for the optimal triangular coding strategy, and generates coded packets for the AP to broadcast. The core component of Percy is a search algorithm that combines a set of optimizations to drastically reduce the complexity of scanning all triangular coding strategies to become feasible in real time. Our simulation results show that Percy outperforms the previous inter-layer coding heuristic by up to 22-80% with varying numbers of clients.

ACKNOWLEDGMENT

This work was supported in part by NSF grants CCF 0845968, CNS 0905331, and Qatar National Research Fund (QNRF) No.08-374-2-144.

REFERENCES

[1] "Vlc. streaming howto with videolan manager. vlc multiple streams."
[2] U. Horn, K. Stuhlmller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15(1), 1999.
[3] M. Effros, "Universal multiresolution source codes," *IEEE Trans. on Information Theory*, vol. 47(6), 2001.
[4] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proc. of the IEEE*, vol. 89(1), 2001.
[5] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12(6)), 2002.
[6] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74–93, Sept 2001.
[7] S. Gheorghiu, L. Lima, J. Barros, and A. L. Toledo, "On the performance of network coding in multi-resolution wireless video streaming," in *Proc. of IEEE NetCod 2010*, 2010.
[8] M. Halloush and H. Radha, "Practical network coding for scalable video coding in error prone networks," in *Proc. of Picture Coding Symposium (PCS)*, 2009.
[9] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. of the IEEE*, vol. 86(5), 1998.
[10] J. Chakareski and P. A. Chou, "RaDiO edge: rate-distortion optimized proxy-driven streaming from the network edge," *IEEE/ACM Transaction on Networking*, vol. 14(6), 2006.
[11] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, 2006.
[12] H. Seferoglu, Y. Altunbasak, O. Gurbuz, and O. Ercetin, "Rate distortion optimized joint ARQ-FEC scheme for real-time wireless multimedia," in *Proc. of IEEE ICC*, 2005.
[13] H. Seferoglu and Y. A.-B. Ozgur Gurbuza, Ozgur Ercetina, "Rate-distortion based real-time wireless video streaming," *Elsevier Image Communication*, vol. 22(6), 2007.
[14] N. Sundaram, P. Ramanathan, and S. Banerjee, "Multirate media stream using network coding," in *Proc. of 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.
[15] S. Dumitrescu, M. Shao, and X. Wu, "Layered multicast with interlayer network coding," in *Proc. of IEEE INFOCOM*, 2009.
[16] M. Kim, D. Lucani, X. shi, F. Zhao, and M. Médard, "Network coding for multi-resolution multicast," in *Proc. of IEEE INFOCOM*, 2010.
[17] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE JSAC*, vol. 27(5), 2009.
[18] M. Halloush and H. Radha, "Network coding with multi-generation mixing," in *Proc. of the Conference on Information Sciences and Systems (CISS)*, 2008.
[19] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: maximizing sensor network data persistence," in *Proc. of ACM SIGCOMM*, 2006.
[20] R. Razavi, M. Fleury, M. Altaf, H. Sammak, and M. Ghanbari, "H.264 video streaming with data-partitioning and growth codes," in *Proc. of IEEE Int'l Conf. Image Processing (ICIP)*, 2009.
[21] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.
[22] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, 1996.
[23] V. Roca, C. Neumann, and D. Furodet, "Low density parity check (LDPC) staircase and triangle forward error correction (FEC) schemes," in *IETF Request for Comments*, June 2008, rFC 5170.
[24] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. of 4th Workshop on Network Coding, Theory, & Applications (NetCod)*, 2008.
[25] J. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *ISIT*, 2008.
[26] T. M. Cover, "Comments on broadcast channels," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, 1998.
[27] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," in *Proc. of PADS Workshop*, May 1998.