Video Streaming Over Wireless LAN With Network Coding

Mo'taz Al-Hami, Abdallah Khreishah†, and Jie Wu
Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
†Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102

***Abstract*—Client diversity is one of the main characteristics of wireless networks. Due to channel diversity, multicasting a video stream in a wireless LAN to multiple clients with different channel conditions is a challenging task. A promising approach for such a problem is to use multiresolution video coding (i.e scalable video coding) with network coding. In this paper, we study a triangular approach for network coding in a one-hop wireless LAN network. Previous work searches for all possible coding opportunities, which is computationally expensive. In this work, we use regression to derive efficient transmission protocols that take into account the delivery rate, as well as the variance of the channels. We show that the regression approach is more practical than the previous method. Also, the achievable rate using the regression approach can produce competitive results.**

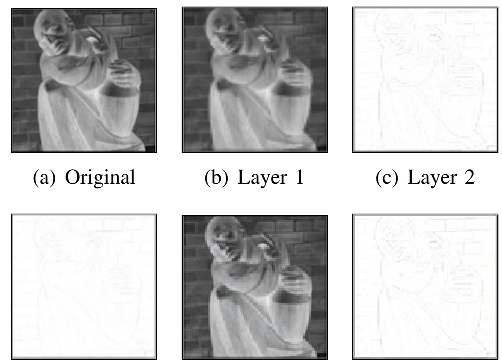
***Index Terms*-Multiresolution video streaming, network coding, inter-layer, intra-layer, video coding, regression.**

I. INTRODUCTION

Delivering media streams (i.e. YouTube video stream) over the Internet is becoming increasingly popular. Therefore, many approaches discuss the efficiency of delivering video over the Internet, how to generate video streams with different resolutions, and how to send these streams over the Internet. The use of different resolutions can efficiently satisfy different client characteristics, where clients with low channel conditions will receive low resolution streams, and clients with high channel conditions will receive high resolution streams.

In a wireless environment, where the client diversity property and shared medium exist due to the channel loss and broadcast nature, there are many approaches used to deliver the video stream to multiple clients. These approaches include unirate multicast, where only one rate stream is generated for multiple receivers. Using unirate multicast, some of the receivers with channel conditions weaker than the generated rate will starve, while other receivers with strong channel conditions are restricted to receive streams according to the generated rate [1].

Since the video stream consists of a sequence of video frames, each frame can be decomposed into multiple layers, each having a different resolution. According to this decomposition, the client diversity can be handled through decoding streams with different resolutions. In Multi-Resolution Coding (MRC) [2] the frame is decomposed into layered resolutions as shown in Fig. 1, where there is a basic layer and many refinement layers. The benefit of MRC is its ability



(a) Original (b) Layer 1 (c) Layer 2 (d) Layer 3 (e) Layers 1 & 2 (f) Layers 2 & 3

Figure 1: Multiresolution Coding Using 3 Layers.

to satisfy receivers with different channel conditions. In Fig. 1, Matlab wavelet toolbox is used to generate different layers with different resolutions. In MRC, the frequency of generated streams differs according to the layer level, which means substreams generated by the first layer have lower frequencies than substreams generated by the second layer, and substreams generated by the second layer have lower frequencies that substreams generated by the third layer.

Fig. 1 shows an original image (Fig. 1(a)), the constructed layers from this image (Fig. 1(b)-1(d)), and the effect of combining some of the layers together (Fig. 1(e)-1(f)). Layer 1 (i.e. basic layer) is the most important layer, and is necessary for any client to decode this layer. Layers 2 and 3 are considered to be refinement layers that increase the quality of the constructed image. On the other hand, adding layer 1 and layer 2 will increase the quality of the constructed image, while adding layer 2 and layer 3 alone, without layer 1, will drastically degrade the constructed image quality.

Network coding is used to maximize network throughput by utilizing network capacity. Network coding supports the ability to share the bandwidth between nodes in a way that maximizes throughput [6, 3].

The following represents a motivating example, showing the effect of network coding on network throughput. If we assume a given wireless LAN network with lossy channel, having throughput equal to Packet Delivery Rate (PDR) from the base station to the receiver equal to 0.5, this means half of the sent packets can be received by the receiver. We assume the frame deadline is 6 time slots (i.e. after 6 time slots the frame is meaningless), and the number of packets per layer is equal to 1. In this example, we use MRC. Without network coding, and using the round robin manner, the packets will be sent according the following order: $L_1, L_2, L_3, L_1, L_2, L_3$. If the channel is active through slots 3, 5, and 6, then the receiver will receive L_3, L_2, L_3; this makes the receiver unable to decode any layer, because layer 1 is not received. Let us compare the round robin approach with the following triangular network coding approach: instead of sending L_i in the round robin approach, we send a random linear combination of all the layers below, including L_i. In this case, if the channel is active in slots 3, 5, and 6, then the receiver is able to receive a linear combination of the coded packets, according to triangular scheme discussed in this paper, $\alpha_1 L_1 + \beta_1 L_2 + \gamma_1 L_3$, $\alpha_2 L_1 + \beta_2 L_2$, and $\alpha_3 L_1 + \beta_3 L_2 + \gamma_3 L_3$. From this, the receiver will have received 3 linearly independent combinations with 3 variables, and it becomes able to decode the original 3 layers. The rest of this paper is organized as follows. The related works are discussed in Section II. Problem formulation is introduced in Section III. The triangular coding schemes are described in Section IV. Section V introduces performance analysis. Finally, the conclusion is in Section VI.

II. Related Work

There are many previous studies related to video streaming. Video streaming with MRC has been studied in [2, 8]. The work in [2] includes a study that uses realistic feedback on the same unreliable channel. The idea behind this work is that when the server sends a generation of coded packets (i.e. coded packets related to a GOP), then it starts the loss recovery process, in order to recover lost packets. Realistic feedback is sent back from receiver nodes to the server in order to minimize the amount of unnecessary transmissions for recovering lost packets. Other studies use video streaming with MDC [5, 4]. For example, the work in [5] employs MDC using an optimized rate allocation algorithm to minimize overall distortion. The work uses set partitioning in the hierarchical trees (SPIHT) algorithm to generate a convenient video stream that uses adaptive bit-rate according to the network conditions. Any rate changes in the network can be accommodated by dropping unnecessary packets from the generated stream. Many other studies, like [9] study video streaming in wireless environments; the approach focuses on minimizing the congestion experienced by video stream by jointly allocating link capacity and traffic flow. The work uses a cross-layer design framework that aims to support maximum data rates and yields minimum end-to-end delay.

III. Problem Formulation

In this paper, we consider a video stream as a sequence of packets such that they have a deadline of X transmissions (i.e. the number of transmissions that Access Point (AP) can send within a deadline) in a one-hop wireless LAN. Also, we assume the existence of loss on channel, and we assume a Bernoulli channel. In this case, our problem is finding the best way (i.e. strategy) of distributing these X transmissions, among L layers that enable heterogeneous receivers with different channel conditions to decode as many layers as possible in a one-hop wireless LAN environment. Throughout this work, we will show the effect of channel variance on the decoding process, and we will try to figure out how to maximize the throughput under these environment conditions. In addition to that, we will discuss many scenarios of receivers.

IV. Triangular Coding Schemes

Triangular network coding makes the decoding process more flexible than Random Linear Network Coding (RLNC), and instead of having full cardinality coded packets (i.e. full square matrix) to decode all the layers, it allows the coded packets to deal with partial layers from 1 to L, to decode layers from 1 to L only. The structure of triangular network coding supports common prioritization, where the basic layer has more priority than the upper layers.

In inter-layer network coding, any layer i depends on all previous layers (i.e. layers $1, .., (i-1)$) to decode this layer [2]. Triangular schemes consider only L ways of coding packets from L layers, where the k^{th} way is coding packets from the first k layers, for $k \in 1..L$. The AP can send as many as X transmissions within the frame of a deadline corresponding to NL packets for L layers, where N is the number of packets per layer. In this work, to solve our problem, we consider all possible triangular schemes which are denoted as $(x_1, .., x_l)$, where $\sum_{i=1}^{L} x_i = X$, and x_i denotes the number of packets that are used to code the first i layers [18]. The unique possible ways of assigning X transmissions into L ways of generating the coded packets are equal to $\binom{X-1+L}{L-1}$ [7].

A. Effect of Choosing Value N

Throughout this work, we study the effect of choosing different values of N on the decoding process, and the effect of any value of N on the achievable rate, at which packets are received on each receiver. We define the achievable rate, at which packets are received on any receiver, as:

$$R = \frac{L \cdot N}{X} \tag{1}$$

Different values of N can result in different achievable rates.

B. Variance Problem

Calculating the received packets out of each strategy by using $PDR{\cdot}x_i$ for all layers, where x_i represents the number of packets transmitted on layer i, is a roughly inaccurate process, because it does not reflect the actual case. Fig. 2 shows the actual throughput and the ideal throughput, so any analysis should consider the actual throughput rather

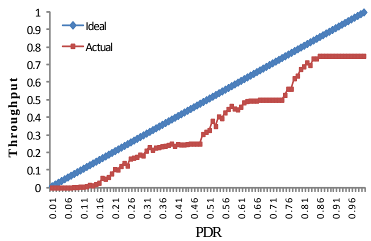


Figure 2: Comparison Between Actual Throughput and Ideal One for One Receiver.

than the ideal one. Fig. 2 was obtained by comparing the ideal case (i.e. the case where there is no variance property, and at the same time, all received packets participated in the decoding process) and the actual case, where we used triangular optimal algorithm [7]. Channel variance has a major effect on the ability to decode layers, and can affect the achievable rate at which a receiver can decode layers.

C. Expected Throughput

To alleviate the variance problem effect, we should find the characteristics of the strategy, and the choice of N that generates the maximum throughput. This can be done when the X transmissions are distributed among many layers, or only one layer.

For one-layer case: Our task is to find the best N value for given PDR, which means that we want to find the value of N where a given receiver can get higher throughput as often as possible. Since receiving N transmissions out of the original X transmissions happens with probability:

$$P[N] = \sum_{i=N}^{X} \binom{X}{i} \cdot PDR^i \cdot (1 - PDR)^{X-i} \qquad (2)$$

Then, we can find the expected throughput for each value of N, ranging from $[1..X]$ given PDR, and the original X transmissions value such that:

$$E[N] = \sum_{i=N}^{X} \binom{X}{i} \cdot PDR^i \cdot (1 - PDR)^{X-i} \cdot N \qquad (3)$$

For multiple-layers case: In this case, the original X transmissions are distributed among several layers, so our task is to find the strategy and the N value for a given PDR value that generates the maximum expected throughput. At the same time, we should consider the decoding criteria for any reception outcome. For any reception outcome $(y_1, ..., y_L)$, the receiver can decode the first i layers if:

$$\sum_{j=i}^{i-k} y_j \geq (k+1) \cdot N, \quad \forall \ k \in [0, i-1] \qquad (4)$$

Based on this decoding criteria, we want to find the value of N, and the strategy, that satisfies the decoding criteria. We also aim to generate the maximum expected throughput. For this, we extended eq. 3 for the multiple-layers case; for any given strategy $[x_1, .., x_L]$, the expected throughput $E[N]$ is such that:

$$\sum_{y_i \leq x_i} \prod_{i=1}^{L} \binom{x_i}{y_i} \cdot PDR^{y_i} (1 - PDR)^{x_i - y_i} \cdot B \cdot N$$
$$s.t. \quad \sum_{j=B}^{B-k} y_j \geq (k+1) \cdot N, \quad \forall \ k \in [0, B-1] \qquad (5)$$

In eq. 5, B represents the number of decoded layers. The eq. 5 is calculated for all strategies, and the strategy with maximum expected throughput is chosen as the best strategy. Using eq. 5, we can find the expected throughput for each value of N, given the equation parameters like PDR, original X transmissions value, and the original number of layers L, which is equal to 4 layers. Based on the generated results, we can determine the best choice of the value N, and the actual number of layers used in distributing the X transmissions. One more thing we have to mention here is how to distribute the X transmission among these actual layers. In this approach, we used majority voting, based on generated results, to decide how to distribute X transmissions among actual L layers. The main influence in regression approach regarding running time makes it more practical to implement and adopt. Table I shows the extracted equations through the regression approach. These generated values of N can be assigned as decoding criteria N(packets/layer) for their associated PDR values. Therefore, any receiver that has a given PDR value can use the related N as a required number of receptions on that layer. To detect the performance of regression technique, we applied the regression equations directly to the simulator, which is prepared using Matlab, we then compared our results with optimal technique, which is used in [7].

D. Applying Regression

Until this point, we constructed the expected throughput table (due to space limitation we just provide the basic idea), which contains the value of N that maximizes the expected throughput for a given PDR value, as well as the strategies used to produce the maximum expected throughput. We apply a regression technique in an attempt to extract equations that approximate the relationship between PDR, N, and L, for given X transmissions. Since the running time is a critical issue for finding the optimal strategy, as in the case of optimal method used in [7], the running time increases drastically as the number of transmissions increase. Simultaneously, a strategy table is used to calculate the decoded layers for each strategy. In the regression approach, there is no need to keep any table; all that we need is to use the extracted regression equations directly to calculate the required N and L. Based on these values, we adopt a given strategy.

No of Receivers	N and L Regression Equations
1	$N = \lfloor(-0.199999999999999 + 9.09090909090909 * PDR)\rfloor$ $L = \lfloor(2.46666666666667 - 0.121212121212122 * PDR)\rfloor$
2	$N = \lfloor(-0.66 - 5.333 * PDR1 + 9.2727 * PDR2)\rfloor$ $L = \lfloor(3.2 + 0.242 * PDR1 - 0.36363 * PDR2)\rfloor$
3	$N = \lfloor(-1.052380 - 0.90 * PDR1 - 1.83982 * PDR2 + 7.229437 * PDR3)\rfloor$ $L = \lfloor(3.63809 + 0.2813 * PDR1 - 0.84415 * PDR2 - 0.02164 * PDR3)\rfloor$
4	$N = \lfloor(-1.42857 - 3.67965 * PDR1 + 1.55844 * PDR2 + 2.66233 * PDR3 + 3.57142 * PDR4)\rfloor$ $L = \lfloor(3.7619 + 0.90 * PDR1 - 0.974 * PDR2 - 0.99567 * PDR3 + 0.47619 * PDR4)\rfloor$
5	$N = \lfloor(-1.10476 - 2.36363 * PDR1 - 3.24242 * PDR2 + 3.757 * PDR3 + 4.212 * PDR4 + 0.757 * PDR5)\rfloor$ $L = \lfloor(3.85714 + 1.36363 * PDR1 + 0.2121 * PDR2 - 1.969 * PDR3 - 0.03 * PDR4 + 0.333 * PDR5)\rfloor$

Table I. regression equations extracted through expected throughput analysis.

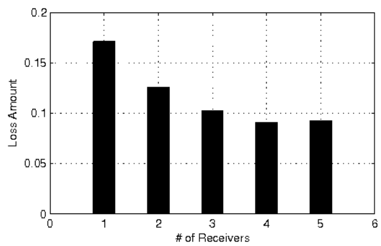


Figure 3: Comparing Loss Percentage for Different Receivers to Optimal Approach.

V. PERFORMANCE ANALYSIS

The performance analysis focuses on the achievable rate. Achievable rate is defined as the number of decoded packets per time unit. The performance of the achievable rate varies according to the number of receivers. Since the regression technique constructs a linear function that fits the dependent variables with a minimum sum of squared errors, the constructed linear function will not achieve results like optimal approach. Regardless, it will still achieve very good rates when compared with the optimal approach. We use PDR values ranging from 0.1 to 1.0, with increments of 0.1. The results in Fig. 3 show the average loss percentage in the regression approach, compared to the optimal approach for $X = 16$ transmissions. The objective function here is to maximize the total achievable rate at all receivers. The results show that the loss percentage does not exceed 0.18 for one receiver, and does not exceed 0.1 for 5 receivers. According to this graph, as the number of receivers increases, the average percentage of loss rate decreases. This is due to the diversity of PDR values in the optimal approach. Fig. 4 compares the achievable rate ratio between regression approach and optimal approach for different numbers of receivers, using the empirical CDF function for each case. In general, the graph is biased toward the right, which means the ratio is approaching 1.0 for a majority of PDR values. For example, the median of the PDR values achieves a rate with more than a 0.9 ratio between the regression approach and the optimal approach for 2 or more receivers.

VI. CONCLUSION

In this work, we present video streaming over wireless LAN, and we describe the problem for many heterogeneous receivers. Previous work in [7] showed many triangular coding schemes, including the optimal scheme, that requires constructing a table to calculate the number of decoded layers for each strategy; this is found to be time-consuming. We study the effect of choosing the number of frames per layer on the total achievable rate, and we analyze this relationship. Using regression technique, we try to find an equation that controls the relationship between PDR value, the most appropriate number of frames per layer, and the number of used layers.

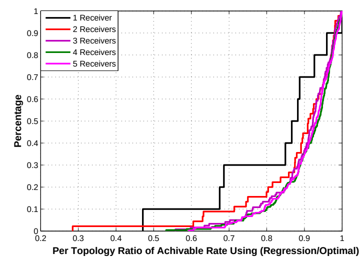


Figure 4: Empirical CDF for the Different Topologies and Different Numbers of Receivers.

REFERENCES

[1] S. Dumitrescue, M. Shao, and X. Wu. Layered multicast with inter-layer network coding. In *28th IEEE (INFOCOM)*, Apr 2009.

[2] S. Gheorghiu, L. Lima, J. Barros, and A. Toledo. On the performance of network coding in multi-resolution wireless video streaming. In *IEEE International Symposium on Network Coding (NetCod), USA*, Jun 2010.

[3] M. Halloush and H. Radha. Network coding with multi-generation mixing: analysis and applications for video communication. *IEEE/ACM Transactions on Wireless Communication*, 10(2):466 – 473, 2011.

[4] J. Kim. Layered multiple description coding for robust video transmission over wireless ad-hoc networks. In *Proceedings of World Academy of Science, Engineering, and Technology (PWASET)*, 2006.

[5] J. Kim, R. Mersereau, and Y. Altunbasak. Network-adaptive video streaming using multiple description coding and path diversity. In *International Conference on Multimedia and Expo (ICME)*, Jul 2003.

[6] R. Koetter and M. Medard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, 2003.

[7] D. Koutsconikolas, Y. Hu, C. Wang, M. Comer, and A. Mohammad. Efficient online WiFi delivery of layered-coding media using inter-layer network coding. In *International Conference on Distributed Computing System (ICDCS), USA*, Jun 2011.

[8] L. Lima, J. Barros, M. Medard, and A. Toledo. Secure network codig for multi-resolution wireless video streaming. *IEEE Journal on Selected Areas in Communications*, 28(3):377–388, 2010.

[9] T. Yoo, E. Sehon, X. Zhu, A. Goldsmith, and B. Giord. Cross-layer design for video streaming over wireless ad hoc networks. In *IEEE 6th Workshop on Multimedia Signal Processing*, Sep 2004.