Minimizing Delay for Video Conference with Network Coding

Hui Zhang [a,b], Jin Zhou[b], Zhen Chen[b,c] and Jun Li[b,c]
a Department of Automation, Tsinghua University, China
b Research Institute of Information Technology, Tsinghua University, China
c Tsinghua National Lab for Information Science and Technology, China
zhanghui04@mails.tsinghua.edu.cn, {zhoujin, zhenchen,junl}@mail.tsinghua.edu.cn

ABSTRACT

Video conference is an attractive and promising application which allows immersive communication and discussion among people at different and distant places. However, its stringent delay and bandwidth requirements limit its scale and spread over current Internet. This paper introduces a network coding algorithm for video conference system to minimize the maximal transmission delay during multicast while retaining high throughput at the same time. The simulation results show that the proposed network coding scheme can reduce delay by 10% to 20% compared with ordinary multicast.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: [Data communications]

Keywords

Network Coding, Video Conference, Interactive Multimedia

1. INTRODUCTION

With the proliferation of the Internet, there are increasing demands for multi-party video conferencing which allows face-to-face intercommunion regardless of geographical distance among the attending parties. However, the two inherent properties limit its scale and the popularity in current Internet: (1) High interactivity, which brings particular stringent delay requirement. ITU-T Recommendation G.114 specifies that one-way transmission delay of VoIP below 150ms is considered to be the same quality with PSTN, and delay above 400ms is unacceptable. As video should be synchronous with voice during communication, video conference has to achieve similar delay performance with VoIP. (2) High bandwidth consumption. Consider a scenario that the sender intends to transmit a stream with the rate of 400$kbps$ to ten receivers in the conference, the simple scheme like Figure 1(a) would cost 4$Mbps$ outbound bandwidth at the sender side, which is unachievable for most of the home or hotspot users.

In literature [1], the authors bring forward an example which minimizes the delay in virtue of network coding (see Figure 1). In this example, the network contains a single sender and three receivers, connected by directed unit-capacity edges. If the overall delay is measured by the maximum number of hops for a packet to reach each receiver, the best spanning tree in the metric of delay is shown in

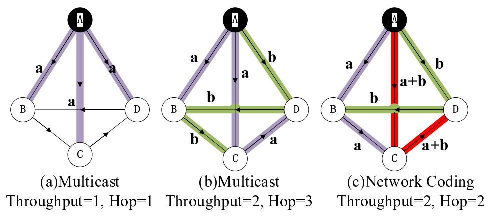

Figure 1: A minimizing delay example

Figure 1(a), where the delay is one hop but the throughput is not optimized as the capacity of the three bottom edges are wasted. Figure 1(b) shows a scheme with better throughput achievement of two at the price of increasing delay to three (see the green path from A to C). In contrast, Figure 1(c) shows that it is possible to reduce the hop to two, when network coding is applied, where stream a routes along the purple path, stream b routes along the green path, and their XOR $a + b$ routes along the red path. Network coding improves the overall delay by one if the participants communicate at the throughput of two.

The objective of this paper is to investigate whether and how determined network coding scheme like Figure 1(c) could improve the metric of delay comparing with multicast under the same throughput achievement, which is the first attempt to our knowledge.

2. MINIMIZING DELAY SCHEME

2.1 Problem Formulation

This section formulates minimizing delay with network coding strategy as an optimization problem, based on the algebraic framework advanced in [2]. In that frame work, a network is represented by a directed graph $G(V, E)$ with vertex set V representing nodes and directed edge set E representing links. Then a linear network code can be specified by triple matrices of (A, F, B), where A denotes source process transfer matrix, F denotes adjacency matrix on edges and B denotes output transfer matrix.

The proposed network coding scheme is based on the following assumptions: (1)All nodes are logically equal: each maintains a member list and takes full charge of its own multicast tree. (2)All nodes have the same and limited bandwidth. (3)Link is with unit-capacity, but a pair of nodes may have more than one links between them. (4)End-to-end delay is mainly caused by the propagation delay along the Internet route and buffering delay at intermediate nodes.

Let the vector $\underline{d} = (d_{e_1}, d_{e_2}, ..., d_{e_{|E|}})$ represents the delay along each directed edge, and $\underline{D} = (D_{e_1}, D_{e_2}, ..., D_{e_{|E|}})$ denotes the delay from source node to the tail of corresponding edge, according to the fourth assumption, we have:

$$D_{e_i} = d_{e_i} + \max \sum_{l=1}^{|E|} U(\beta_{e_l,e_i}) \bullet D_{e_l} \tag{1}$$

where $U(x) = \begin{cases} 1 x > 0 \\ 0 x \leq 0 \end{cases}$ is a jump function, and β_{e_l,e_i} is the element in matrix F. If a stream along edge e_i combines more than one stream along its upstream, namely more than one coefficients $\beta_{e_l,e_i} > 0$, it has to wait until all of the up streams arrive at the head of this directed edge $head(e_i)$.

As the participants are ordinary end host with limit bandwidth resource, increasing the number of links on one node not necessarily increases the bandwidth consumed on that node proportionally when the summation of the bandwidth exceeds the end node's capacity. So the notion of relative throughput is introduced instead of absolutely throughput.

$$T_r = \frac{\mu}{\max(\delta_I(v) + \delta_o(v))} \tag{2}$$

where the numerator is the length of the source vector, which equals to the Min-Cut (absolutely throughput) if corsponding networking coding problem is solvable, and the denominator is the maximal degree of the vertex set (maximal bandwidth consumption).

Now the problem can be formulized as devise a solvable network coding triple (A, F, B) which could minimize $\max(D_{e_n})$ under the constraint of $T_r \geq T_{\min}$.

2.2 Heuristic Algorithm

One straightforward way to solve the proposed optimization is to traverse across algebraically closed field for each element in the triple of (A, F, B). However, such exponential-time computation is unbearable when the number of node increases. If flow solutions from source to each receiver are found, network coding scheme could be generated by polynomial-time algorithm proposed in existing work [3]. So the problem can be transformed as finding paths with short delay and "suitable for" network coding. Based on this idea. we propose an approximate polynomial-time algorithm (see the pseudo-code in Figure 2).

The basic procedure is as follows: (1)Produce a series discrete degree pairs which satisfy the throughput constraint $T_{\min}$. One degree represents the maximal inbound and outbound degree a node could contribute, and the other is the minimal inbound degree a receiver should have. (2)Calculate optimal routines under each degree pair constraint. The Theorem in [2] shows that networking coding problem is solvable if and only if Min-Cut Max-Flow bound is satisfied from the source to all the receivers, therefore the number of Min-Cut edge-disjoint paths from sender to each receiver are needed. Each path is found by Dijkstra algorithm under degree constraints. (3)Generate network coding by the algorithm in [3] and compute maximal delay from source to the receivers.

3. SIMULATION

We generate a topology includes 179 nodes scattered in 32 countries and 5 continents on PlanetLab testbed. The receiver set ranges from 2 to 40. Sender and receivers are generated randomly. As shown in Figure 3, under the throughput constraint of 0.3, network coding (NC) reduces maximal delay by 10% (about $20ms$) when there are more than ten receivers, which goes up to 15% (about $40ms$) under throughput constraint of 0.6, and more than 20% under throughput constraint of 0.9. Especially, under throughput constraint of 0.9, the delay in multicast grows as four times fast as that of network coding, and is unacceptable (exceeding the threshold of $400ms$) under the situation of more than 25 receivers. The experimental results demonstrate that network coding is quite useful under challenging condition of more receivers and high throughput requirement.

```
For each $\max(\delta_I(v)+\delta_o(v))$ and $\delta_I(u)$ degree pair
    For $i=1:\delta_I(u)$
        For $j=1:|u|$
            Find the $ith$ edge-disjoint path $P_{ij}$ from the source $s$
            to one of the receiver $u_j$ by Dijkstra algrithom
            Mark the edges along the path $P_{ij}$
             If $head(e)$ or $tail(e)$ reach dgree constraints
                Eliminate edge $e$
            End
        End
    End
    Gentrate network coding and compute $\max(D_{e_n})$
End
Output $(A,F,B)$ gentrates $\min(\max(D_{e_n}))$
```

Figure 2: Heuristic algorithm of network coding

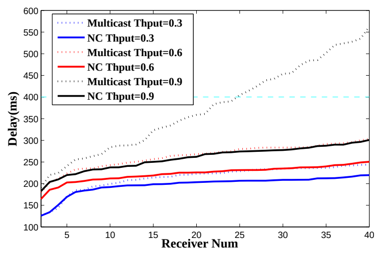

Figure 3: Performance comparison

4. CONCLUSION

This paper introduces determined network coding into video conference to minimize delay and maintain high throughput, which are two crucial factors impacting its quality. Primary simulation demonstrates its effectiveness. The algorithm proposed in this paper can be applied to many other interactive multimedia applications in addition to video conferencing. Our future work will take network dynamics, node heterogeneity, and interaction between multiple multicast sessions into consideration to make proposed scheme more practical and robust, and evaluate the enhanced algorithm in real network environment.

5. REFERENCES

[1] P. A. Chou and Y. Wu. Network coding for the internet and wireless networks. *IEEE Signal Processing Magazine*, 2007.

[2] R. Koetter and M. Mdard. An algebraic approach to network coding. *IEEE Transactions on Network*, 2003.

[3] S. E. Peter Sanders and L. Tolhuizen. Polynomial time algorithms for network information flow. In *ACM Symposium on Parallel Algorithms and Architectures*, 2003.