Minimising delay for video conference with network coding

Hui Zhang

3-421 FIT Building,
Tsinghua Universtiy,
Beijing 100084, China
E-mail: zhanghui04@mails.tsinghua.edu.cn

Jin Zhou

3-419 FIT Building,
Tsinghua Universtiy,
Beijing 100084, China
E-mail: zhoujin@gmail.com

Jun Li*

4-412 FIT Building,
Tsinghua Universtiy,
Beijing 100084, China
E-mail: junl@tsinghua.edu.cn
*Corresponding author

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 attempts to introduce the prosperous technology of network coding (NC) into video conference to minimise the maximal transmission delay between source and all the participants while retaining high bandwidth utilisation at the same time. Extensive numerical analysis and PlanetLab experiments demonstrate that the proposed innovative NC scheme can significantly improve the delay performance for video conference like multi-party interactive multimedia applications than conventional application level multicast.

Keywords: NC; network coding; video conference; interactive multimedia application; multicast**.**

Reference to this paper should be made as follows: Zhang, H., Zhou, J. and Li, J. (2011) 'Minimising delay for video conference with network coding', *Int. J. Information and Coding Theory*, Vol. 2, No. 1, pp.38–58.

Biographical notes: Hui Zhang received her PhD degree in Control and Information in 2010 from Tsinghua University and BS degree in Information Engineering in 2004 from Xi'an Jiaotong University. Her research interests are distributed network and multimedia processing. Now she has more than ten academic publications.

Jin Zhou received his PhD and BS degree in Control and Information in 2005 and 2000, respectively, from Tsinghua University. His research focuses on distributed network and system. Now he has more than 20 academic publications.

Jun Li received his PhD degree in Computer Science from the New Jersey Institute of Technology in 1992 and MS and BS degrees in Control and Information in 1985 and 1988, respectively, from Tsinghua University. His research interests include network architecture and security. Now he has published more than 80 academic publications and several patents.

1 Introduction

With the proliferation of the internet, there are growing demands for multi-party video conferencing which allows face-to-face inter-communion or discussion for people regardless of geographical distance between them. Similar requests also arise in long-distance learning, telemedicine and global business. These applications require a kind of multi-point small group interactive communication which simultaneously delivers same content to all participating nodes. Compared with another prosperous one-to-many application video streaming, video conference has two properties:

1 *High interactivity*: It brings particular stringent end-to-end delay requirement. ITU-T Recommendation G.114 specifies that one-way transmission delay of voice over Internet Protocol (VoIP) below 150 msec is considered to be the same quality with Public Switched Telephone Network, and delay above 400 msec is unacceptable (ITU-T Recommendation G.114, 2000). As video should be synchronous with voice during communication, video conference has to achieve similar delay performance with VoIP. In contrast, in real video streaming system such as CoolStreaming and PPlive, playback delay, which is the time interval between packet generated at the source and packet played out at a receiver, can be as long as several minutes (Jiang and Jin, 2006).

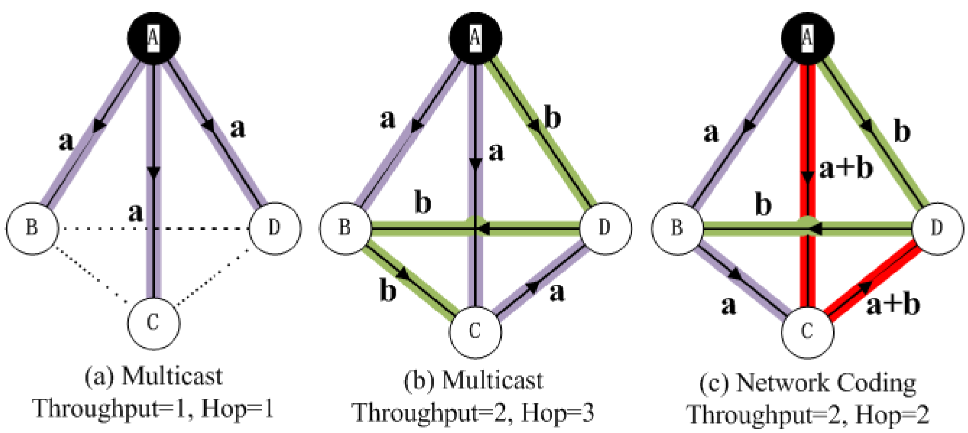
2 *Relatively small scale*: In general, there are only several to scores of participants in one conference, whereas tens of thousands of spectators watching the same channel concurrently in internet video streaming.

These two properties determine that video conference system usually exploits tree-based push strategies (e.g. Chu et al., 2001; Horiuchi et al., 2007; Hosseini and Georganas, 2003; Lim et al., 2009; Luo et al., 2004; Wu et al., 2007) for data delivering, rather than mesh-based pull strategies, which is widely applied in current video streaming systems (e.g. CoolStreaming, Zhang et al., 2005). Tree-based strategies organise participants into one or more multicast trees and push content along these trees. Mesh-based strategies organise participants into a directed graph referred to as a mesh, in which content segments availability are periodically exchanged among neighbouring peers, then pulling requirements are sent to neighbours according to these segments availability. In relation to tree-based strategies, mesh-based strategies have the advantages of more resilient to nodes dynamics and simpler to implement. However, such resilience is achieved at the cost of incremental playback delay (Zhang et al., 2007), which is a crucial metric that affects users' experience in two-way communication like conference. Since the group size of video conference is not too large, group state maintenance can be realised simply by keeping the full member list at a rendezvous point or on every multicast participant. We argue that tree-based strategies are more appropriate for video conference like applications.

Nowadays, the strict delay restriction still limits the scale and the popularity of long-distance multi-party video conferencing. This paper aims to introduce network coding (NC) technology into current tree-based application level multicast (ALM) to reduce the property of transmission delay without much throughput degradation, which might be the first attempt to our knowledge.

In literature, Chou and Wu (2007) bring forward an example which minimises delay in virtue of NC (see Figure 1). In this example, the network contains a single-source node and three sink nodes, connected by directed unit-capacity edges. If the overall delay is measured by the maximum number of hops for a packet to reach a sink, the best spanning tree in the metric of delay is shown in Figure 1(a), where the delay is only one hop but the throughput is not optimised as the capacity of the three bottom edges are wasted. According to the max-flow min-cut theorem in graph theory (Bollobas, 1979), the throughput upper bound between the source and any sink is their min-cut two, and Edmonds' theory (Edmonds, 1973) guarantees that this upper bound is reachable in broadcast situation. So, there exist two edge-disjoint spanning trees along which the source can route two unit-rate streams to the three sinks. Figure 1(b) presents exclusive such spanning trees, note that the maximum depth of which is three (see the green tree). In contrast, Figure 1(c) exhibits that it is possible to reduce the number of the hops to two when NC is applied, where stream *a* routes along the purple path, stream *b* routes along the green path, and their exclusive or *a* + *b* routes along the red path. NC improves the overall delay by one hop if the participants communicate at their maximum throughput of two.

Figure 1 The principle of using NC to minimise delay (see online version for colours)



It should be emphasised that throughput is another crucial factor which influences the performance of video conference. 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 in Figure 1(a) will cost 4 Mbps outbound bandwidth at the sender side, which is unachievable for most of home or hotspot users, although directly deliver content to the receivers in this scheme might be delay efficient.

The objective of this paper is to investigate whether and how much the NC scheme like Figure 1(c) could better the metric of delay, comparing with conventional multicast under the same throughput achievement. The rest of this paper is organised as follows: Section 2 outlines and analyses related work to this paper. Section 3 models delay minimising for one-to-many communication with NC as an optimisation problem and proposes a polynomial time heuristic algorithm to address this optimisation. Section 4 implements an ALM protocol with NC strategy on PlanetLab testbed and conducts a series of experiments to evaluate its performance. Finally, Section 5 concludes this paper with a discussion of future work.

2 Related work

2.1 Multicast scheme for internet conference

Internet protocol (IP) multicast (Deering, 1988) is the earliest approach for internet conferencing. This approach handles multicast at the network layer, which can work reasonably well in networks that support them. However, several inherent architectural problems, such as high complexity, poor scalability and lack of security against malicious attacks, have impeded the global deployment of IP multicast.

In such a condition, video conferencing systems using ALM technologies have been introduced due to their ease of deployment and low cost of operation. ALM leverages participating nodes' resources and capability to replicate and forward packet from one end-node to other intermediate or destination nodes at application layer. Based on the topology organising manner, ALM technologies can be classified into two categories: share tree (e.g. Horiuchi et al., 2007; Wu et al., 2007) and source-specific trees (e.g. Chu et al., 2001; Hosseini and Georganas, 2003; Lim et al., 2009; Lennox and Schulzrinne, 2003). Shared tree topology constructs one tree only to delivery data for all the multicast sessions with different source nodes. The management cost is less than that of using multiple source-specific trees. However, shared tree does not have the delay properties as good as source-specific trees (Wei and Estrin, 1994).

In the scale of conferencing application, we argue that the better delay properties should prevail over the management considerations, and the analysis and investigation in the following sections are based on source-specific trees scenario. Though we believe the proposed NC algorithm is also beneficial and can be applied to share tree topology.

2.2 Roadmap of NC technology

NC is first proposed in the pioneering work (Ahlswede et al., 2000), which showed that with NC, as symbol size approaches infinity, a source could multicast information with throughput reaching the smallest min-cut between the source and any sink. Then, the following work abounds and consummates the theory of NC gradually. In Li and Yeung

(2003), the authors proved that linear coding with finite symbol size was sufficient to achieve multicast capacity. Literature (Koetter and Medard, 2003) brought forward an elegant algebraic framework for NC. The work in Sanders et al. (2003) provided a polynomial time algorithm for finding encoding and decoding coefficients over single-source graph. In Ho and Medard (2006), the authors proved that randomised linear NC could work reasonably well and highlighted a way of random and distributed NC manner in large-scale peer-to-peer networks.

In recent studies, NC has been successfully applied to many practical areas, such as increasing throughput in wireless networks (Katti et al., 2006), decreasing energy cost in *ad-hoc* networks (Wu et al., 2005), improving robust and downloading delay in content distribution (Chou et al., 2003; Gkantsidis and Rodriguez, 2005) and video streaming systems (Feng and Li, 2008). In video streaming system, random NC can ease the problem of finding the rarest segment and eliminate the process of periodically segment requests, thus playback delay can be shortened. However, periodically segment availability exchange is still required, and playback delay could reach 5–10 sec in such a mesh-based data delivering topology (Feng and Li, 2008). This delay achievement is still unacceptable for two-way interactive applications. Therefore, this paper employs traditional determined NC in source-specific trees-based ALM for media distribution, instead of the random NC technology, which has already been widely studied and applied in large-scale peer-to-peer systems.

3 NC scheme for delay minimising

3.1 Preliminaries

Our investigation is based on the algebraic framework advanced in prior work (Koetter and Medard, 2003). In that framework, a network is represented by a directed graph $G(V,E)$ with vertex set V representing nodes and directed edge set E representing links. Each link is unit-capacity and multiple edges are allowed between two vertices, hence edge is denoted by $e(v,v',i)$ where the last integer enumerates edges between two vertices. The head and the tail of a directed edge are denoted by $v = \text{head}(e)$ and $v' = \text{tail}(e)$, respectively. $\Gamma_{\text{I}}(v)$ is defined as the set of edges which end at a vertex v and $\Gamma_{\text{O}}(v)$ is defined as the set of edges which originate at v, namely:

$$\Gamma_{\text{I}}(v) = \{e \in E : \text{head}(e) = v\}$$
$$\Gamma_{\text{o}}(v) = \{e \in E : \text{tail}(e) = v\}$$

Then, the in-degree of v can be expressed as $\delta_{\text{I}}(v) = |\Gamma_{\text{I}}(v)|$, while the out-degree can be expressed as $\delta_{\text{o}}(v) = |\Gamma_{\text{o}}(v)|$.

Let $\underline{x}(s) = (X(s,1), X(s,2), \ldots, X(s,\mu))$ be a vector of discrete random input processes that are observed at the exclusive source node s (this paper considers one source only, but the framework in Koetter and Medard (2003) is applicable for multiple sources). Let $\underline{z} = (Z(v_1,1), Z(v_1,2), \ldots, Z(v_1,\upsilon(v_1)), Z(v_2,1), \ldots, Z(v_{|V|},\upsilon(v_{|V|})))$ denote a vector of the output processes, $\upsilon(v_j)$ is set to be zero if node v_j is not a sink node. The length of vector $\underline{z}$ is $\upsilon = \sum_i \upsilon(v_i)$. M is defined as a transfer matrix from the source processes to the output processes if $\underline{z} = M\underline{x}$, which characters global transfer relationship between the input processes and the output processes.

The investigation in Koetter and Medard (2003) illustrates that a linear NC can be specified by the triple matrices of (A, F, B) with:

$$M = A(I-F)^{-1}B^T \tag{1}$$

where I is the $|E|\times|E|$ identity matrix. The $\mu\times|E|$ matrix A can be viewed as a transfer matrix from source process to media data on source node's outgoing links, with its element:

$$A_{i,j} = \begin{cases} \alpha_{l,e_j} & x_i = X\left(\text{tail}\left(e_j\right), l\right) \\ 0 & \text{otherwise} \end{cases}$$

The $|E|\times|E|$ adjacency matrix F specifies how media is transmitted and linear combined between incident links, with

$$F_{i,j} = \begin{cases} \beta_{e_i,e_j} & \text{head}\left(e_i\right) = \text{tail}\left(e_j\right) \\ 0 & \text{otherwise} \end{cases}$$

The $\upsilon\times|E|$ transfer matrix B characters how media outputs from terminal links to the output processes, with

$$B_{i,j} = \begin{cases} \varepsilon_{e_j,l} & z_i = Z\left(\text{head}\left(e_j\right), l\right) \\ 0 & \text{otherwise} \end{cases}$$

The coefficients of $\alpha_{l,e_j}, \beta_{e_i,e_j}, \varepsilon_{e_j,l}$ are elements in the field of $\mathbb{F}_{2^n}$. In Koetter and Medard (2003), the authors prove the following theorem:

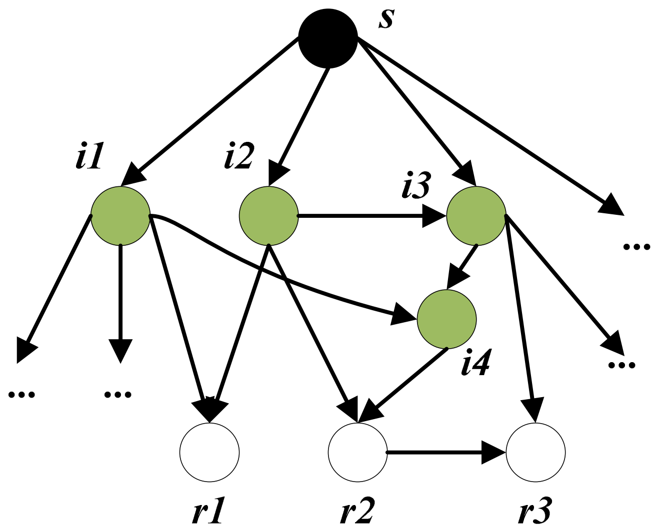
Theorem: $C = \{(s, r_1, X(s)), (s, r_2, X(s)), \ldots, (s, r_{|R|}, X(s))\}$ *denotes a set of connection from a single-source node s to the* $|R|$ *destination sinks* $r_1, r_2, \ldots, r_{|R|}$. *The following three statements are equivalent:*

1 *The networking coding problem is solvable, namely all the connections in C are feasible or can be established.*

2 *The min-cut max-flow bound is satisfied for all connections in C.*

The number of $|R|$ sub-matrices M_{s,r_i} is non-singular $\mu\times\mu$ matrix, where M_{s,r_i} describes the transfer matrix between the input processes at *s* and output processes at sink r_i and $M = [M_{s,r_1}^T, M_{s,r_2}^T, \ldots, M_{s,r_{|R|}}^T]^T$.

Figure 2 The architecture of a multicast session (see online version for colours)



3.2 Problem formulation

This section begins the expatiation of our NC strategy for video conference like multi-party interactive multimedia applications with the following hypotheses:

1. All the nodes are logical equal: each of them maintains a member list and takes full charge of its own 'multicast trees'. A multicast session is composed of a single media source, several intermediate nodes and a number of receivers (see Figure 2). There are multiple multicast sessions in one conference concurrently, since every conference participant is a media source.
2. All the nodes have the same and limited bandwidth capacity, namely maximum inbound bandwidth plus outbound bandwidth a node can contribute is fixed to a constant.
3. All the links in the graph have the same unit-capacity, but a pair of nodes can have more than one links between them if they have more than unit available bandwidth between them.
4. End-to-end delay is dominantly caused by the propagation time of signal travelling along internet route and the buffering time at the intermediate nodes. The packets processing time and network encoding/decoding computation time are trivial and negligible.

The proposed NC scheme intends to determine what and how to deliver, coding and replicate packets on each node and edge in a multicast session to minimise the maximum delay from source to a group of receivers and guarantees high throughput achievement at the same time. Let the vector of $\underline{d} = (d_{e_1}, d_{e_2}, \ldots, d_{e_{|E|}})$ represent the delay along each directed edge, and $\underline{D} = (D_{e_1}, D_{e_2}, ..., D_{e_{|E|}})$ denote the delay from source node to the tail of corresponding edge; according to the fourth assumption, we have

$$D_{e_i} = d_{e_i} + \max \sum_{l=1}^{|E|} U\left(\beta_{e_l,e_i}\right) \bullet D_{e_l} \tag{2}$$

where $U(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}$ is a jump function. The second item of formula (2) is zero if edge e_i satisfies $\text{head}(e_i) = s$. If the stream along edge e_i combines more than one stream along its up-streams, videlicet more than one coefficients β_{e_l,e_i} are greater than zero, it has to wait until all of the up-streams arrive at node $\text{head}(e_i)$. While if only one of the β_{e_l,e_i} is greater than zero, $\text{head}(e_i)$ can replicate and forward the arriving packets directly to its down-streams the same as multicast.

Another parameter needs to be concerned is throughput. In ALM conference, all the participants are ordinary end host with restricted bandwidth resource. Increasing the number of links on one node not always improves bandwidth consumed on that node proportionally when the summation of the bandwidth exceeds the end-node's capacity. So, this paper introduces the notion of normalised throughput (NT) in formula (3) instead of absolute throughput.

$$\text{NT} = \frac{\mu}{\max(\delta_{\text{I}}(v) + \delta_{\text{o}}(v))} \tag{3}$$

where the numerator of the equation is the length of the source vector $\underline{x}$, which equals to the min-cut (absolute throughput) if corresponding networking coding problem is solvable (refer to the Theorem in Section 3.1); the denominator of the equation is the maximal degree of a vertex in the graph consumes. For example, in Figure 1(a) NT = 1/3, while in Figure 1(b) and (c) NT = 2/3.

Then the minmising delay with NC problem can be formulated as devising a solvable NC triple matrices (A, F, B) which could minimise $\max(D_{e_m})$ under a series of constraints, as summarised in the following optimisation.

$$\min\left(\max\left(D_{e_m}\right)\right)$$

$$\text{s.t.} \begin{cases} \text{NT} = \dfrac{\mu}{\max\left(\delta_{\text{I}}\left(v_j\right) + \delta_{\text{o}}\left(v_j\right)\right)} \ge \text{NT}_{\min} \\ D_{e_i} = d_{e_i} + \max \sum_{l=1}^{|E|} U\left(\beta_{e_l,e_i}\right) \bullet D_{e_l} \\ M = A(I-F)^{-1} B^T \\ \det\left(M_{v,r_k}^T\right) \ne 0 \\ \text{tail}\left(e_m\right) \in R \\ 1 \le i \le |E|,\ 1 \le j \le |V|,\ 1 \le k \le |R| \end{cases} \tag{4}$$

where $\text{NT}_{\min}$ is the minimal normalised throughput the NC solution should reach.

3.3 Heuristic algorithm for optimisation addressing

This section discusses how to efficiently solve the optimisation of formula (4) and construct relevant NC. One brute force way is to traverse across the algebraically closed field for each element in the triple matrices of (A, F, B) and pick up the delay optimal solution. However, this algorithm involves checking a multivariate polynomial identity with an exponential number of recombination coefficients and becomes unbearable along with the growing of the number of nodes. For example, consider a graph consists of 20 nodes, each node maintains 10 neighbours, and the average bandwidth of each pair of nodes is 2 unit-capacities. The size of matrix (A, F, B) is close to 400×400 with approximately 400×20 possible non-zero elements. Therefore in the finite field of $\mathbb{F}_{2^3}$, up to 8^{8000} times of checking has to be executed exclusively on matrix *F*.

If flow solutions from source to each sink are determined, NC could be constructed by polynomial time algorithm proposed in existing work (Sanders et al., 2003). Thus, optimisation problem (4) can be transformed as finding a set of edge-disjoint paths with shortest delay for each sink individually in the first stage, and then generating NC solution in the second stage among the union of the selected edge-disjoint paths. Edge-disjoint paths means the paths do not share any common edges with each other. The number of edge-disjoint paths for each sink is equal to its min-cut in the graph produced after the first stage. Based on this fundamental thought, this paper devises an approximate polynomial time algorithm to solve the optimisation (Equation (4)) (see the pseudo-code in Figure 3). The basic procedure is listed as follows:

1 Enumerate a series of discrete degree pairs which satisfy the throughput constraint $\mathrm{NT}_{\min}$. One degree in the pair equals to the denominator $\max(\delta_{\mathrm{I}}(v) + \delta_{\mathrm{o}}(v))$ in formula (3), representing the maximal degrees (inbound plus outbound) a node could contribute. The other is the minimal inbound degree a sink should offer, videlicet the numerator $\delta_{\mathrm{I}}(r)$ in formula (3), which also equals to the length of the input vector $\underline{x}(s)$.

2 Calculate optimised routines under each degree pair constraint. The Theorem in Section 3.1 guarantees that networking coding problem is solvable if and only if min-cut max-flow bound is satisfied from the source to all of the sinks. Therefore, the number of min-cut ($\delta_{\mathrm{I}}(r)$) edge-disjoint paths should be found for each sink. In this paper, the path is selected by Dijkstra shortest path algorithm under the degree constraints.

3 Generate NC by the algorithm in Sanders et al. (2003) and compute maximal delay from source to the set of sinks.

Figure 3 Polynomial time heuristic algorithm on NC construction: the objective is to devise a linear NC solution, which achieves the normalised throughput $\mathrm{NT} \geq \mathrm{NT}_{\min}$ and minimises the maximal delay from source to each sink r

```
For each degree pair of {max(δ_I(v)+δ_o(v)), δ_I(r)}
  Initialize the topology as original G(V,E)
  For i = 1: δ_I(r)
    For j = 1: |R|
      Temporarily eliminate edges
      e ∈ {P_{1,j}, P_{2,j}, ..., P_{i-1,j}}
      Find the ith edge-disjoint path P_{i,j} from s
      to sink r_j by Dijkstra algrithom
      Recover the edges e ∈ {P_{1,j}, P_{2,j}, ..., P_{i-1,j}}
      Mark the edges e ∈ P_{i,j}
      For k = 1: |E|
        If head(e_k) | tail(e_k) reaches degree
          constraints & e_k ∉ selected Path set
          Eliminate edge e_k
        End
      End
    End
  End
  Construct network coding and calculate
  max(D_{e_m})
End
Output (A,F,B) which gentrates min(max(D_{e_m}))
```

Section 4 verifies the superiority of our NC scheme to conventional multicast. One source multicast is a typical Steiner-tree problem, and a variety of approximate algorithms have been proposed in the past to solve this non-deterministic polynomial-completeness problem. This paper adopts the algorithm proposed in Kou et al. (1981) with slight modifications. The pseudo-code is given in Figure 4. It differs from the NC algorithm mainly in two aspects:

1 The stream delivered in one spanning tree takes the same information on all the edges and independent with that in the other spanning trees, therefore the spanning trees necessitate being edge-disjoint with each other.

2 To avoid a small number of spanning trees use up all the outbound degrees resource at source s (e.g. in Figure 1(a), one spanning tree takes up all the three outbound degrees of the source node A), the number of outbound degrees each spanning tree could take up at the source is limited and allocated during the initialisation stage.

Figure 4 Heuristic algorithm on multicast: this algorithm builds several edge-disjoint spanning trees, which achieve the same NT as NC and also provide optimal maximal delay from source to the sink set

```
For each degree pair of {max(δ_I(v)+δ_o(v)), δ_I(r)}
  Initialize the topology as original G(V,E)
  Allocate outbound degrees of s on each spanning tree
  For i=1:δ_I(r)
    Initialize spannin tree T_i={s}
    While exist a sink r_j ∉ T_i
      Find closest path from s to sink r_l ∉ T_i by Dijkstra
      For k=1: |E|
        If head(e_k) | tail(e_k) reaches degree
          constraints && e_k ∉ selected Tree set
          Eliminate edge e_k
        End
      End
    End
    Eliminate and marked edges e ∈ T_i
  End
  Calculate max(D_{e_m})
End
Output edge-disjoint trees that gentrate min(max(D_{e_m}))
```

3.4 Complexity analysis

1 Brute force NC

Let δ be the total degrees of each node and δ_{I} be the inbound degrees of each sink in graph $G(V,E)$. Then, it is expected that there are $\delta_{\mathrm{I}}\times\delta$ possible non-zero elements in the $\mu\times|E|$ matrix A, $\delta\times|E|$ possible non-zero elements in the $|E|\times|E|$ matrix F and $\delta_{\mathrm{I}}^{2}\times|R|$ possible non-zero elements in the $\upsilon\times|E|$ matrix B. Hence in the finite field size of $|\mathbb{F}|\geq|R|$, exponential times of testing $O(|\mathbb{F}|^{|E||R|^2\delta^2\delta_{\mathrm{I}}^3})$ have to be carried out by this traversal algorithm. It is substantial computation time demanded and unpractical algorithm.

2 Proposed heuristic NC

Finding an edge-disjoint path from source to one sink takes time in $O(|V|^2)$ by Dijkstra algorithm. Edges eliminting, recovering and checking before and after Dijkstra spend $O(|E|)$ time. Hence, producing one edge-disjoint path for a sink spends time of

$O(|V|^2+|E|)$. The two main loops in the path-finding stage collectively iterate over all sinks' inbound degrees, so that there is a total number of $|R|\times\delta_I$ iterations. The expected cost for finding flow solution in the first stage is $O(|R|\delta_I(|V|^2+|E|))$. The execution time of NC generation in Sanders et al. (2003) is $O(|E||R|\delta_I(|R|+\delta_I))$. This flow finding and NC construction process will be iterative $O(\delta)$ times. Combining all the parts, we get the total polynomial running time of $O(|R|\delta\delta_I(|V|^2+|E||R|+|E|\delta_I))$.

3 Conventional multicast

As shown in Figure 4, to build one edge-disjoint spanning tree, the multicast algorithm exerts one Dijkstra process until all the sinks join current spanning tree. It will cost time in $O(|V|^2+|E||R|)$ which involves the degree constraints checking steps across each edge. Building the number of δ_I spanning trees across all the enumerated degree pairs takes expected bound of $O(\delta\delta_I(|V|^2+|E||R|))$ time in total. Accordingly, this computational complexity is expected to be approximately $|R|$ times faster than our NC algorithm.

3.5 Discussion of multicast vs. NC

On condition that all the coefficients in the triple matrices of (A,F,B) are set to be one or zero, no linear combination needs to be processed on any edge or node, so that we get a conventional multicast scheme which can be viewed as a special case of NC. Accordingly, the performance of NC will no worse than multicast in theory. This section inquires into the inherent mechanism in NC which can benefit delay from intuitive point of view.

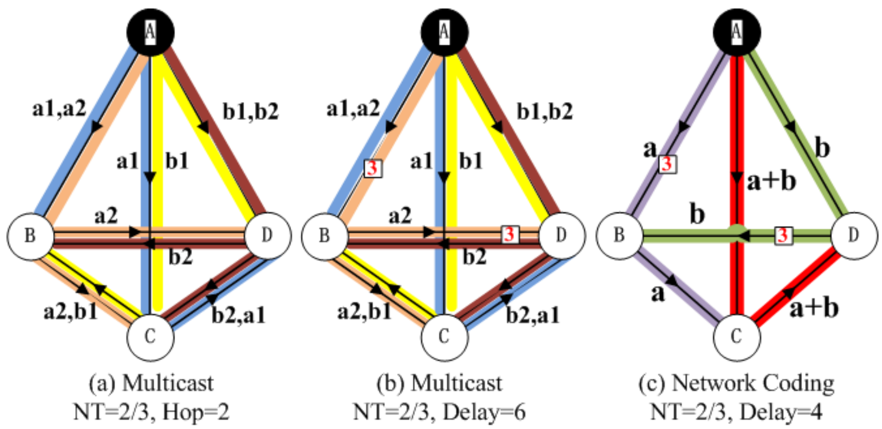
1 NC reuses more edges

Multicast requires each spanning tree to be edge-disjoint, whereas NC only necessitates the paths destination to the same sink to be edge-disjoint, therefore NC has the ability of reusing the edges already accommodated in other 'trees', which widens the possibility of picking up edges with shorter delay. For example, take Figure 1(c), the first path ('tree') from source to the three sinks is $A\rightarrow B$, $A\rightarrow C$ and $A\rightarrow D$, respectively, and the second edge-disjoint path for B is $A\rightarrow D\rightarrow B$, which takes up edge $A\rightarrow D$ in the first 'tree'.

2 NC reduces the number of paths

In some special cases, multicast can attain the same delay and throughput as NC through increasing the number of degrees on each node. Figure 5(a) gives such an example. Compared with Figure 1(b), Figure 5(a) extends the degree pair (δ,δ_I) from $(3,2)$ to $(6,4)$. Each pair of nodes is connected by two links and some of the links are reversed with each other. The depth of all the four spanning trees in Figure 5(a) is two. Note that it performs equivalently to the NC scheme given in Figure 1(c), in term of both NT and maximal delay. However, consider a more general and realistic scenario where the length of the links are diverse from each other, NC scheme in Figure 5(c) could shorten delay from six in multicast scheme Figure 5(b) to four with identical NT achievement 2/3. As can be seen in Figure 5(b), there are 12 distinct paths from source to the 3 sinks in total, compared with 6 of that in Figure 5(c). The length bottleneck link $B \leftrightarrow D$ is accommodated twice bidirectional along path $A \to B \to D$ (length six) and path $A \to D \to B$ (length four) in Figure 5(b), whereas in Figure 5(c), only the latter path exists. As a result, relatively fewer paths in NC scheme potentially reduce the possibility of routing through longer path.

Figure 5 Second example of using NC to minimise delay (see online version for colours)



Note: The directed edges with the same colour deliver the same media content. The length of the edges $A \to B, B \to D, D \to B$ is three, and the length of other edges is one.

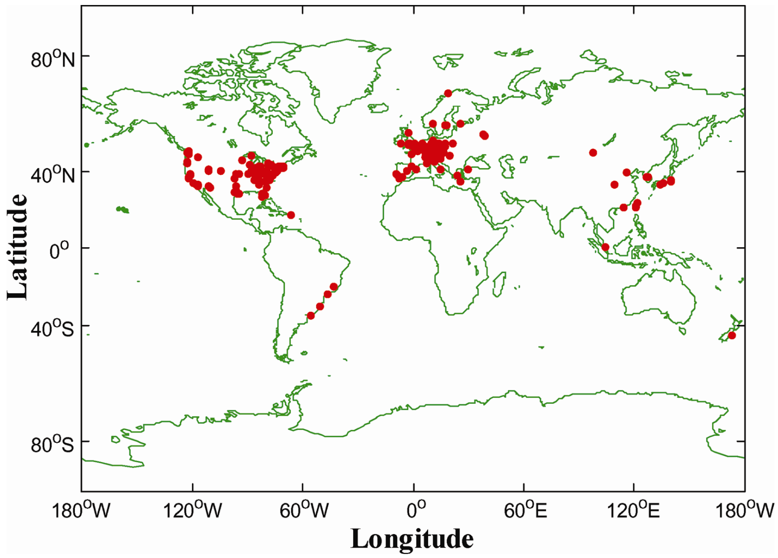
4 Performance evaluation

This section contains some numerical analysis and real internet measurement results of the proposed NC scheme. The experiments were deployed on PlanetLab testbed, covering 203 nodes, which scatter in 203 distinct sites, 28 countries and 5 continents. The detail distribution of these nodes is depicted in Figure 6.

4.1 Numerical simulation

To evaluate the proposed NC scheme in a controllable and impartial environment, we carry out experiments to collect delay between every pair of the selected 203 PlanetLab nodes with ping command and then generate corresponding directed graph. Successions of comparative simulations are conducted on this exclusive topology in identical circumstance for network both coding and multicast. In each simulation, one source and several receivers are randomly picked up, afterwards routing/coding parameters and prospective delay achievement in NC and multicast are calculated in accordance with the algorithms described in Figures 3 and 4, respectively.

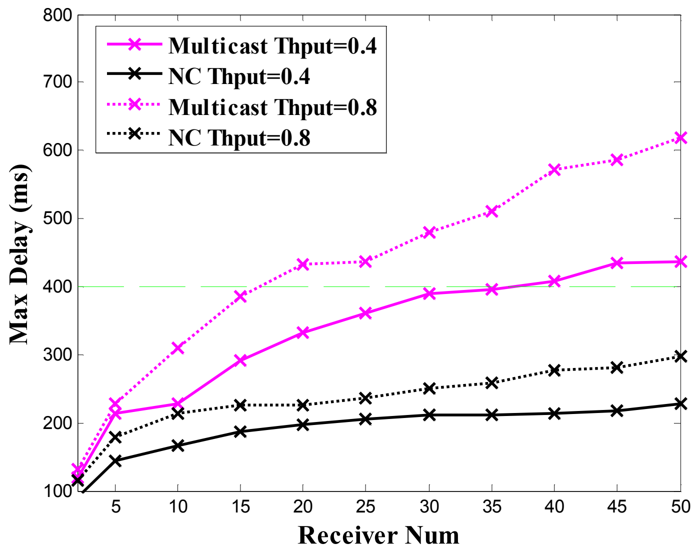
Figure 6 Geographical distribution of the PlanetLab nodes in our evaluation (see online version for colours)



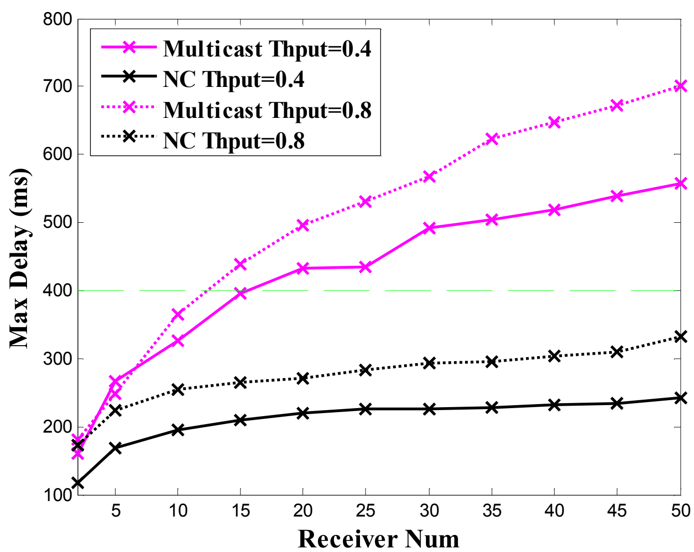
The size of the receiver set range from 2 to 50 and the simulation is repeated several hundred times under each scenario. As shown in Figure 7(a), with the throughput constraint of 0.4, NC shortens maximal delay by 25% (about 60 msec) when there are more than ten receivers, which goes up to 30% (about 100 msec) with throughput constraint of 0.8. The number of receivers has distinctly more significant impact on multicast. As can be seen, the delay in multicast grows as ~3.5 times fast as that in NC with the inflation of receiver set. Especially, the metric of delay becomes unacceptable (exceeding the threshold of 400 msec) under the situation of more than 35 receivers with 0.4 throughput requirement and more than 15 receivers with 0.8 throughput requirement. These results indicate that NC is quite useful under challenging condition of more receivers and high throughput demands.

Although application layer nodes can set up connection to almost all the others, it is rather costly to maintain and probe the metric of delay to all the others in pace with expansion of the network scale. Thus, we investigate how NC and multicast perform when a node only detects part of other nodes' information (see Figure 7(b) and (c). The detected nodes are referred to as neighbours.). Delay performance of multicast degrades rapidly along with the scale-up of the receivers under the situation of limited neighbours. It can merely support less than 15 receivers under 20 neighbours' circumstance and less than 10 receivers under 10 neighbours' circumstance. In contrast, the performance of NC is only affected apparently under the most critical scenario in regard to 10 neighbours, 0.8 throughput restriction and more than 40 receivers. Intuitively, the reduction of neighbours depresses the number of available edges as well as paths in the graph, as discussed in Section 3, multicast is apt to take up more edges and paths than NC, so that it is more impressible to the size of neighbour set.

Figure 7 Simulation results with different neighbour set size: (a) full topology (maintains all possible neighbours); (b) 20 neighbours; (c) 10 neighbours (see online version for colours)

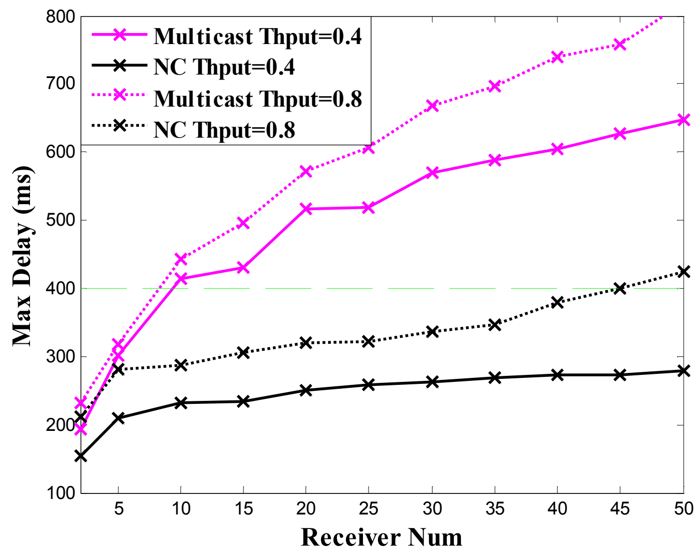


(a)



(b)

Figure 7 Simulation results with different neighbour set size: (a) full topology (maintains all possible neighbours); (b) 20 neighbours; (c) 10 neighbours (see online version for colours) (continued)



(c)

4.2 Internet implementation

To verify whether or not the proposed NC scheme can practically improve the delay performance for internet video conference, we fully implemented it on PlanetLab testbed and carried out several real-world experiments. The implemented protocol consists of three components: control panel, decision panel and data panel. Control panel is responsible for organising and monitoring the overlay network; decision panel covers our core idea of calculation optimal routing and coding scheme by addressing optimisation (Equation (4)) in real-time; data panel takes charge of packets delivering, scheduling, encoding/decoding and replicating during communication.

Our protocol in term of control panel refers to the thought of the well-known multicast protocol Narada (Chu et al., 2000). Narada constructs a mesh-based overlay structure in a self-organising and fully distributed manner. The members monitor a number of neighbours and may drop poor links. Members also probe non-neighbours at random manner and may add new links with good quality. Narada runs a distance vector protocol on top of the mesh and exchanges routing update messages between neighbours. Our control panel modified Narada by running a link state like protocol and flooding link updating messages to all the overlay members. Thus, every member is aware of the whole topology knowledge of current overlay network and can generate its own source-specific ‘multicast trees’ solution with the help of decision panel.

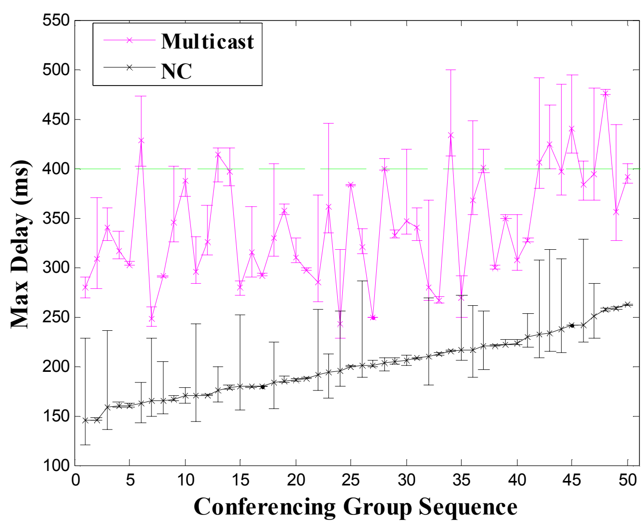
In the series of PlanetLab experiments, each member maintains 20 neighbours and contributes 1 Mbps inbound plus outbound bandwidth at most. User Datagram Protocol data packets with timestamp and sequence number are sent out according to the strategies advised from the data panel. If there are multiple packets available to be delivered or multiple destinations for one packet, scheduling module would transmit the packet with the longest expectation arriving time in priority.

The experimental results are summarised in Figure 8 and Table 1. Each curve in Figure 8 contains 50 random generated conferencing groups with each session lasting 2 min. The *Y*-axis plots the median of maximal delay from source to all the receivers during the 2-min lasting multicasting, with the confidence interval from 5% to 95%. As can be seen, all the delay results in multicast scheme exceed 400 msec with the receivers set of 40. When take the delay variety into consideration, only 60% and 30% sessions in multicast scheme can always guarantee less than 400 msec delay in the two 10 receivers' scenarios (see Figure 8(a) and (b)). In contrast, most of the multicast sessions in NC scheme can well satisfy the 400 msec delay demand and only a small proportion of them in Figure 8(d) exceed 400 msec slightly. These preliminary experiments manifest that our NC scheme is agreeable and operative on delay reducing and scale enlarging for video conference in real-world environment.

Table 1 Average maximal hop and maximal delay (msec) comparison across 50 multicast groups in PlanetLab experiments

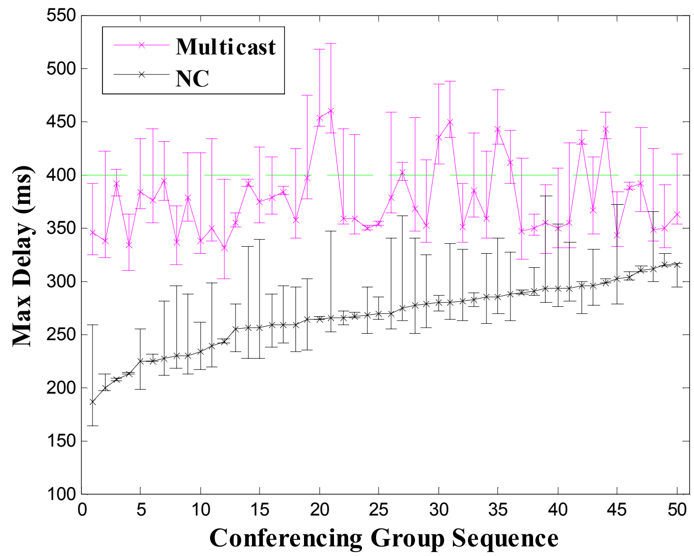
		Multicast		*NC*	
Receiver numbers	*Throughput*	*Max delay*	*Max hop*	*Max delay*	*Max hop*
10	0.4	342	6.9	200	4.1
10	0.8	376	7.6	267	5.6
40	0.4	522	11.4	234	4.7
40	0.8	659	14.2	325	7.4

Figure 8 PlanetLab experiment results: (a) 10 receivers throughput = 0.4; (b) 10 receivers throughput = 0.8; (c) 40 receivers throughput = 0.4; (d) 40 receivers throughput = 0.8 (see online version for colours)

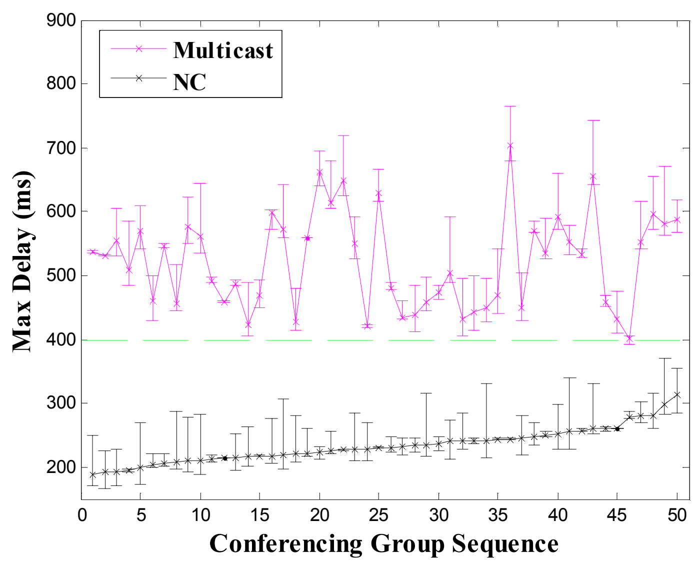


(a)

Figure 8 PlanetLab experiment results: (a) 10 receivers throughput = 0.4; (b) 10 receivers throughput = 0.8; (c) 40 receivers throughput = 0.4; (d) 40 receivers throughput = 0.8 (see online version for colours) (continued)

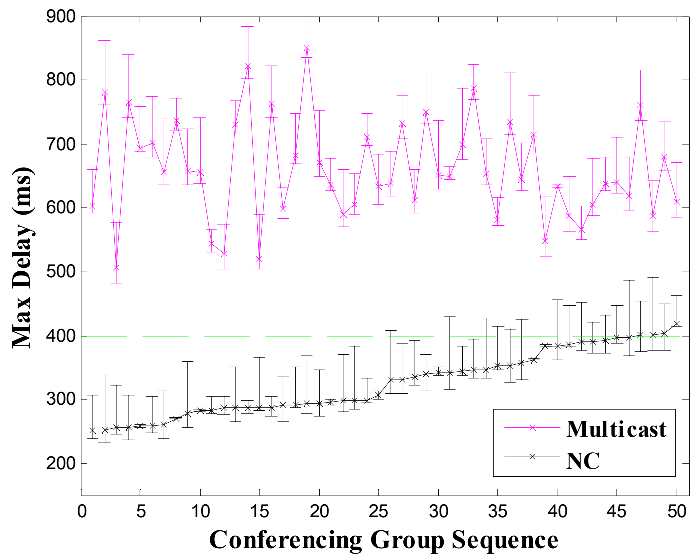


(b)



(c)

Figure 8 PlanetLab experiment results: (a) 10 receivers throughput = 0.4; (b) 10 receivers throughput = 0.8; (c) 40 receivers throughput = 0.4; (d) 40 receivers throughput = 0.8 (see online version for colours) (continued)



(d)

Note: Each node maintains 20 neighbours. The confidence interval of the error bars is 90%.

5 Conclusion

This paper introduces determined NC into video conference to minimise delay while maintaining high throughput, which are two crucial factors impacting its quality. A series of theoretical analysis and experiments on PlanetLab testbed demonstrate the effectiveness of the proposed NC algorithm compared with broadly used conventional multicast. We believe NC is a promising way to scale and popularise video conference like applications.

It should be pointed out that this paper mainly focuses on whether and how much NC could favour delay for video conference. Our future work will take network dynamics, node heterogeneity and interaction between multiple multicast sessions into consideration to make the proposed NC scheme more practical and robust in realistic network environment.

References

Ahlswede, R., Cai, N., Li, S-Y.R. and Yeung, R.W. (2000) 'Network information flow', *IEEE Transactions on Information Theory*, Vol. 46, No. 4, pp.1204–1216.

Bollobas, B. (1979) *Graph Theory: An Introductory Course*. New York: Springer-Verlag.

Chou, P.A. and Wu, Y. (2007) 'Network coding for the Internet and wireless networks', *IEEE Signal Processing Magazine*, Vol. 24, No. 5, pp.77–85.

Chou, P.A., Wu, Y. and Jain, K. (2003) 'Practical network coding', Proceedings of Allerton *Conference on Communication, Control, and Computing*.

Chu, Y., Rao, S.G., Seshan, S. and Zhang, H. (2001) 'Enabling conferencing applications on the Internet using an overlay multicast architecture', *Proceedings of ACM SIGCOMM*.

Chu, Y., Rao, S.G. and Zhang, H. (2000) 'A case for end system multicast', *Performance Evaluation Review*, Vol. 20, No. 8, pp.1456–1471.

Deering, S.E. (1988) 'Multicast routing in internetworks and extended LANs', *Proceedings of ACM SIGCOMM*.

Edmonds, J. (1973) 'Edge-disjoint branchings', in B. Rustin (Ed.), *Combinatorial Algorithms*. New York: Academic Press.

Feng, C. and Li, B. (2008) 'On large-scale peer-to-peer streaming systems with network coding', *Proceedings of ACM Multimedia*.

Gkantsidis, C. and Rodriguez, P. (2005) 'Network coding for large scale content distribution', *Proceedings of IEEE INFOCOM*.

Ho, T. and Medard, M. (2006) 'A random linear network coding approach to multicast', *IEEE Transactions on Information Theory*, Vol. 52, No. 10, pp.4413–4430.

Horiuchi, H., Wakamiya, N. and Murata, M. (2007) 'Proposal and evaluation of a network construction method for a scalable P2P video conferencing system', *Proceedings of International Conference on Information Networking (ICOIN)*.

Hosseini, M. and Georganas, N. (2003) 'Design of a multi-sender 3D video conferencing application over an end system multicast protocol', *Proceedings of ACM Multimedia*.

ITU-T Recommendation G.114 (2000) 'One way transmission time'.

Jiang, H. and Jin, S. (2006) 'NSYNC: network synchronization for peer-to-peer streaming overlay construction', *Proceedings of the ACM NOSSDAV*.

Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M. and Crowcroft, J. (2006) 'XORs in the air: practical wireless network coding', *Proceedings of ACM SIGCOMM*.

Koetter, R. and Medard, M. (2003) 'An algebraic approach to network coding', *IEEE Transactions on Network*, Vol. 11, No. 5, pp.782–795.

Kou, L., Markowsky, G. and Berman, L. (1981) 'A fast algorithm for Steiner trees', *Acta Informatica*, Vol. 15, No. 2, pp.141–145.

Lennox, J. and Schulzrinne, H. (2003) 'A protocol for reliable decentralized conferencing', *Proceedings of ACM NOSSDAV*.

Li, S.Y.R. and Yeung, R.W. (2003) 'Linear network coding', *IEEE Transactions on Information Theory*, Vol. 49, No. 2, pp.371–381.

Lim, B.P., Ettikan, K.K., Lin, E.S., Khoa, T., Thoai, P.N., Muramoto, E. and Tan, P.Y. (2009) 'Bandwidth fair application layer multicast for multi-party video conference', *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC)*.

Luo, C., Li, J. and Li, S. (2004) 'DigiMetro: an application-level multicast system for multi-party video conferencing', *Proceedings of IEEE GLOBALCOM*.

Sanders, P., Egner, S. and Tolhuizen, L. (2003) 'Polynomial time algorithms for network information flow', *Proceedings of 15th ACM Symposium on Parallel Algorithms and Architectures*.

Wei, L. and Estrin, D. (1994) 'The trade-offs of multicast trees and algorithms', *Proceedings of International Conference on Computer Communications and Networks (ICCCN)*.

Wu, X., Dhara, K. and Krishnaswamy, V. (2007) 'Enhancing application-layer multicast for P2P conferencing', *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC)*.

Wu, Y., Chou, P.A. and Kung, S-Y. (2005) 'Minimum-energy multicast in mobile ad-hoc networks using network coding', *IEEE Transactions on Communication*, Vol. 53, pp.1906–1918.

Zhang, M., Zhang, Q., Sun, L. and Yang, S. (2007) 'Understanding the power of pull-based streaming protocol: can we do better?', *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 9, pp.1678–1694.

Zhang, X., Liu, J., Li, B. and Yum, T.S.P. (2005) 'CoolStreaming/DONet: a data-driven overlay network for efficient live media streaming', *Proceedings of IEEE INFOCOM*.