Distributed Rate Allocation in P2P Networks with Inter-Session Network Coding

Eirina Bourtsoulatze, Nikolaos Thomos, and Pascal Frossard
Signal Processing Laboratory (LTS4)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
{eirina.bourtsoulatze, nikolaos.thomos, pascal.frossard}@epfl.ch

***Abstract*—In this paper, we propose a distributed rate allocation algorithm for delay minimal data delivery in overlay networks where multiple sources compete simultaneously for the available network resources. In order to efficiently utilize the network resources, we propose to use inter-session network coding. We devise a distributed algorithm that allows peers to determine the optimal coding combinations and the packet rates for each type of combinations to be requested from the parent peers. The rate allocation problem is formulated as a decoding delay minimization problem, where every peer seeks the rates that minimize the average expected delay of the peer and its children peers. To solve this non-convex optimization problem, we introduce the concept of equivalent packet flows, which permits to estimate the expected number of packets that every peer needs to collect for decoding. We then decompose our original rate allocation problem into a set of convex subproblems, which we eventually combine to obtain the solution to the delay minimization problem. The results demonstrate that the proposed scheme eliminates effectively the bottlenecks and reduces the delay time experienced by users with limited resources.**

I. Introduction

P2P network architectures have rapidly developed over the past few years and constitute nowadays the basis for a vast variety of applications, such as for example file sharing, multimedia streaming and multiparty conferencing. The essential advantage of P2P systems over the traditional client-server architecture is their scalability, as peers contribute their upload bandwidth to the system. This renders P2P networks especially suitable for streaming applications, which are associated with high bandwidth demands. However, P2P networks are usually characterized by dynamics that make centralized scheduling highly inefficient. A broad spectrum of distributed algorithms exists in the literature [1], [2], [3], which address the problem of media streaming. Most of the existing works deal with the scenario where one streaming server or multiple servers provide the same data content. The optimal routing over multicast trees in the presence of multiple servers is known to be NP-hard. Polynomial time algorithms for computing P2P network capacity and the associated trees are presented in [4] for various scenarios including the multi-source multicast problem. In parallel to traditional routing algorithms, network coding techniques have been also considered recently in P2P streaming systems [5], [6] as a way of distributing the data delivery control among the peers and enhancing the quality of the reconstructed video.

This work has been partly supported by the Swiss National Science Foundation, under grants 200021-118230 and PZ00P2-121906.

In this paper, we present a distributed rate allocation algorithm for data delivery over P2P networks where multiple concurrent sessions stream data to the network clients. Specifically, we consider a set of streaming servers $\mathcal{S}$ and a set of peer nodes $\mathcal{N}$ that request different data assets. The peers are organized in an overlay P2P network topology. The network is assumed to be directed and free of cycles. Thus, it can be modeled as a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ represents the set of network nodes, such that $\mathcal{V} = \mathcal{S} \cup \mathcal{N}$. $\mathcal{E}$ is the set of connecting links between the network nodes. The directed link connecting any two nodes i and j is denoted as $(i, j) \in \mathcal{E}$. It is characterized by the channel capacity b_{ij} expressed in packets/sec and the average packet loss probability π_{ij}.

The data servers simultaneously stream independent data sequences to the network at a rate of U_s, $s \in \mathcal{S}$, packets/sec. Prior to transmission, each stream is partitioned into generations of packets of size N_s [7]. Subsequently, random linear combinations of packets of the same generation are formed and the coded packets are forwarded to the children peers. The peer nodes act both as end users and relay nodes. Each peer is interested in receiving only one of the streams available at the servers. Since the upload bandwidth of the servers is limited and only a small number of peers can acquire the requested packets directly from the servers, the majority of the peers are served by other peers that receive, encode and relay packets to downstream nodes. The peers are able to implement inter-session network coding [8] and form linear combinations of packets from different streams. Depending on the available set of packets at the parent peers, every peer may request from its parents not only intra-session network coded packets of the requested stream, but also inter-session network coded packets, *i.e.*, packets that are combinations of different streams. These combinations do not necessarily involve packets from the desired stream. Thus, a peer may request and transmit not only packets of the stream that it has subscribed to, but also packets that are useful for its children peers. Upon receiving a sufficient number of network coded packets, the peers decode the received packets with Gaussian elimination. Since for generating the network coded packets, the coding coefficients are drawn randomly according to a uniform distribution from the GF(q), a header of length $\sum_{s\in\mathcal{S}} N_s \log(q)$ bits is appended to the network coded packets. This header identifies all the transformations undergone by the packets while they travel through the network; it renders the decoding process feasible, since the encoding structure is implicit.

Network coding permits to increase the network throughput and alleviate the bottlenecks created by the limited network resources. Inter-session network coding, however, requires proper coding strategy. We determine the optimal coding decisions and the rate of each coded stream by solving distributedly an optimization problem, which aims at minimizing the average expected decoding delay of a peer and its children peers. Such rate allocation algorithm is essential in order to avoid the shortcomings of the random mixing of all the sessions that exist in the network, which would eventually lead to an unacceptable increase in the decoding delay. In order to effectively solve the rate allocation problem, which involves the minimization of a non-convex objective, we introduce the concept of equivalent flows. Using the equivalent flow representation, we decompose the original problem into convex subproblems and then combine the solutions of the subproblems in order to obtain the solution of the initial problem. The experimental results validate the effectiveness of the proposed algorithm. They show reduced decoding delays and efficient exploitation of the available network resources compared to baseline intra-session network coding methods. Finally, we should mention that inter-session network coding has been studied mostly in the context of wireless networks [8], [9]. However, in such settings the challenges and the opportunities for network coding are different from those in wired networks examined in this paper.

II. Distributed streaming with inter-session network coding

A. Communication protocol

The distributed rate allocation solution requires some exchange of information between the peers. Let us consider the peer i and its local neighborhood that consists of the set of parent peers $\mathcal{A}_i$ and the set of children peers $\mathcal{D}_i$. Whenever the peer i wants to optimize the requested packet flow rates, it asks the peers in its neighborhood to provide all the necessary information about the local conditions of the network. Every parent $k \in \mathcal{A}_i$ sends to the peer i a vector $\boldsymbol{R}_k = (R_k^t),\ \forall t \in \mathcal{T}$, where R_k^t represents the total input innovative flow rate of packets of type t available at the parent peer k at the time instant when the peer i performs the optimization of the rate allocation. R_k^t is given as

$$R_k^t = \sum_{n \in \mathcal{A}_k} r_{nk}^t,\ \forall t \in \mathcal{T} \tag{1}$$

The set $\mathcal{T}$ includes all the possible packet types that can be generated in the network. Every element t of $\mathcal{T}$ represents a particular combination of streams. In a network with $|\mathcal{S}|$ concurrent streaming sessions, the number of possible packet types is $2^{|\mathcal{S}|} - 1$.

Similarly to the parent peers, every child peer $j \in \mathcal{D}_i$ forwards to the peer i a vector $\hat{\boldsymbol{R}}_{j\setminus i} = (\hat{R}_{j\setminus i}^t),\ \forall t \in \mathcal{T}$, where $\hat{R}_{j\setminus i}^t$ stands for the total innovative input flow rate of packets of type t that the peer j receives from its parent peers, except for the peer i

$$\hat{R}_{j\setminus i}^t = \sum_{u \in \mathcal{A}_j \setminus i} r_{uj}^t \tag{2}$$

Apart from this information, every child peer j also communicates to the peer i the identity s_j of the stream it wants to receive and the total input bandwidth

$$C_j^d = \sum_{u \in \mathcal{A}_j} b_{uj} \tag{3}$$

Finally, we assume that the peer i is aware of the local network statistics, *i.e.*, the channel capacity and loss rates of the input and output links ($b_{ki},\ \pi_{ki},\ \forall k \in \mathcal{A}_i$ and $b_{ij},\ \pi_{ij},\ \forall j \in \mathcal{D}_i$, respectively).

B. Problem formulation

We are now able to formulate the distributed rate allocation problem that is solved independently in every network peer. It consists in determining the optimal innovative[1] rates that a peer requests from its parents so that the average expected delay of a peer and its children peers is minimized.

Let us denote as $\boldsymbol{r} = (r_{ki}^t, r_{ij}^t),\ \forall k \in \mathcal{A}_i,\ \forall j \in \mathcal{D}_i,\ \forall t \in \mathcal{T}$, the vector of optimal innovative packet flow rates on the input and output links of peer i. Formally, the distributed delay optimization problem in the i-th peer node is stated as

$$\arg\min_{\boldsymbol{r}} \overline{\Delta}_i(\boldsymbol{r}) \quad \text{s.t. } \boldsymbol{r} \in \mathcal{R} \tag{4}$$

$\overline{\Delta}_i(\boldsymbol{r})$ is the average expected delay of the peer i and its children peers and is given by the following expression

$$\overline{\Delta}_i(\boldsymbol{r}) = \frac{1}{|\mathcal{D}_i| + 1}\big(\Delta_i(\boldsymbol{r}, s_i) + \sum_{j \in \mathcal{D}_i} \Delta_j(\boldsymbol{r}, s_j)\big) \tag{5}$$

The search space $\mathcal{R}$ is defined by a set of linear inequality constraints, which determine the set of feasible values of the innovative packet flow rates on the input and output links of the peer i

$$0 \le \sum_{t \in \mathcal{T}} r_{ki}^t \le b_{ki}(1 - \pi_{ki}),\ \forall k \in \mathcal{A}_i \tag{6}$$

$$0 \le \sum_{t \in \mathcal{T}} r_{ij}^t \le b_{ij}(1 - \pi_{ij}),\ \forall j \in \mathcal{D}_i \tag{7}$$

$$\sum_{t' \in \mathcal{T}_{t,s}} r_{ki}^{t'} \le \sum_{t' \in \mathcal{T}_{t,s}} R_k^{t'},\ \forall t \in \mathcal{T},\ \forall s \in \mathcal{S}_t,\ \forall k \in \mathcal{A}_i \tag{8}$$

$$\sum_{t' \in \mathcal{T}_{t,s}} r_{ij}^{t'} \le \sum_{t' \in \mathcal{T}_{t,s}} \sum_{k \in \mathcal{A}_i} r_{ki}^{t'},\ \forall t \in \mathcal{T},\ \forall s \in \mathcal{S}_t,\ \forall j \in \mathcal{D}_i \tag{9}$$

$$\sum_{t' \in \mathcal{T}_{t,s}} \sum_{k \in \mathcal{A}_i} r_{ki}^{t'} \le U_s,\ \forall t \in \mathcal{T},\ \forall s \in \mathcal{S}_t \tag{10}$$

$$\sum_{t' \in \mathcal{T}_{t,s}} r_{ij}^{t'} \le U_s - \sum_{t' \in \mathcal{T}_{t,s}} \hat{R}_{j\setminus i}^{t'},\ \forall t \in \mathcal{T},\ \forall s \in \mathcal{S}_t, \forall j \in \mathcal{D}_i \tag{11}$$

In the above equations, $\mathcal{T}_t$ represents the set of packet types that can be combined to generate packets of type t. The set $\mathcal{S}_t$ consists of the streams that participate in a particular

[1] Innovative are considered the packets that increase the rank of the set of packets that have been previously received by the peer.

combination of packets t. Finally, the set $\mathcal{T}_{t,s}$ includes all the packet types $t' \in \mathcal{T}$ that can be used to generate packets of type t and have as a component packets of the s-th stream.

The constraints appear in pairs and refer to the input and the output links of peer i respectively. Eqs. (6) and (7) are the link capacity constraints, which state that the sum of input and output innovative packet rates for all packet types cannot exceed the link capacity. Eqs. (8) and (9) give upper-bounds to the innovative packet flow rates with the available innovative packet rates at parent nodes. Finally, the last pair of constraints (Eqs. (10) and (11)) limits the innovative packet rate by the available innovative rate provided by the sources, *i.e.*, the peer cannot receive packets faster than they are injected in the network by the sources.

The average expected delay $\Delta_i(\boldsymbol{r}, s_i)$ experienced by the peer i for decoding one generation of packets of the requested stream s_i depends on the average number of packets that the peer i needs to collect for decoding. The latter is a function of the types and the innovative rates of packets that arrive at the peer. We assume that the time is slotted and that one packet is transmitted on each network link in every time slot. Thus, the time needed to receive one packet can be considered constant for a given download bandwidth. Therefore, we can estimate the average expected delay as a product of the time required to receive one packet and the average number of packets that the peer receives before it is able to decode

$$\Delta_i(\boldsymbol{r}, s_i) = d_i E[l] \tag{12}$$

where $d_i = \frac{1}{C_i^d}$ is considered to be the average time needed for the reception of one packet and $E[l]$ is the expected number of packets that the peer will receive before decoding.

The optimization problem stated in Eq. (4) is in general non-convex and its solution requires the computation of the average number of packets $E[l]$ that the peer and its children peers need to receive in order to decode the streams of their interest. In the following section, we introduce the concept of equivalent flows and present an efficient method for computing $E[l]$, that permits to transform the initial problem into a set of convex subproblems and to obtain a solution with low complexity.

III. Decoding analysis with equivalent flows

The equivalent rates are essentially the rates at which peer i collects innovative packets for each component stream when the requested stream is decoded from packets that are combinations of several streams. Let us denote as

$$p_i^t = \frac{\sum_{k \in \mathcal{A}_i} r_{ki}^t}{C_i^d}$$

the probability of receiving a useful packet of type t at peer i. Since this probability is a linear function of the input rates r_{ki}^t, instead of determining the equivalent rates in terms of packets/sec, we can directly derive the equivalent probabilities of receiving an innovative packet that increases the rank of a particular component stream. Let us denote as $q_{i,s}^t$ the equivalent probability to receive at peer i a packet that increases the rank of the set of packets for the s-th component stream when the decoding of a particular stream is performed from intra- and inter-session network coded packets of streams that participate in the combination t. The probability $P_{i,s}^t(l)$ to receive the N_s-th innovative packet for the s-th component stream of the combination t upon receiving exactly l packets at peer i is given by the negative binomial distribution that counts the number of packets received when the N_s-th useful packet arrives at peer i

$$P_{i,s}^t(l) = \binom{l-1}{N_s - 1} (q_{i,s}^t)^{N_s} (1 - q_{i,s}^t)^{l - N_s} \tag{13}$$

The average number of packets that need to be collected by the peer is given by the mean of the negative binomial distribution. In our case, it is simply the number of packets per generation N_s divided by the equivalent probability $q_{i,s}^t$ that the packet is useful

$$E[l] = \sum_{l=N_s}^{\infty} l P_{i,s}^t(l) = \frac{N_s}{q_{i,s}^t} \tag{14}$$

The equivalent probabilities for every component stream of the t-th combination can be expressed as

$$q_{i,s}^t(\boldsymbol{x}_i) = \sum_{t' \in \mathcal{T}_{t,s}} x_{i,s}^{t'}, \quad \forall s \in \mathcal{S}_t \tag{15}$$

where $\boldsymbol{x}_i = (x_{i,s}^{t'}),\ \forall s \in \mathcal{S}_t,\ \forall t' \in \mathcal{T}_{t,s}$ is a vector of probabilities. Every element $x_{i,s}^{t'}$ of this vector is the part of the probability $p_i^{t'}$, which contributes to the equivalent probability for the s-th stream.

The decoding of the requested stream is feasible as soon as the peer receives the last missing packet for the component stream with the non-full rank set of packets. This means that the performance in terms of average decoding delay is driven by the component stream that requires the largest average number of packets to be received for building a full rank system. This leads us to formulating the computation of the equivalent flow rates as a $\min\max$ optimization problem. The objective is to minimize the maximum average number of packets that are required for any component stream to be decoded given the actual rates of the various packet types that exist in the network. Formally, the optimization problem that permits to determine the equivalent rates for the combination t can be written as

$$\min_{\boldsymbol{x}_i} \max_{s \in \mathcal{S}_t} \frac{N_s}{q_{i,s}^t(\boldsymbol{x}_i)} \quad \text{s.t.} \sum_{s' \in \mathcal{S}_{t'}} x_{i,s'}^{t'} = p_i^{t'},\ \forall t' \in \mathcal{T}_t \tag{16}$$

Once we have computed the equivalent rates, we can estimate the average number of packets that need to be received at peer i to decode the desired stream from the combination t. This corresponds to the maximum average number of packets that we need to collect for every component stream in order to decode the requested stream from the t-th combination

$$E_t[l] = \max_{s \in \mathcal{S}_t} \frac{N_s}{q_{i,s}^t} \tag{17}$$

Recall that the peer can decode the requested stream from any combination t that contains this stream as long as it has gathered a full rank set of packets for every component stream of the specific combination. Furthermore, for a given set of innovative packet flow rates, there will be a particular combination of streams for which the peer will on average collect all the innovative packets faster than for all other combinations. To complete our analysis, we need to determine the average number of packets that the peer has to receive in order to be able to decode its stream considering all the possible combinations for decoding. Let us denote as $\mathcal{T}_{s'}$ the set of all possible packet combinations that have as a component stream s'. We can calculate the expected number of packets that the peer will receive before it is able to decode the stream s' as

$$E[l] = \min_{t \in \mathcal{T}_{s'}} E_t[l] = \min_{t \in \mathcal{T}_{s'}} \max_{s \in \mathcal{S}_t} \frac{N_s}{q_{i,s}^t} \quad (18)$$

where we consider the minimum over all possible combinations t that contain the requested stream s'. Specifically, once we have computed the equivalent packet flows for all valid combinations t (the ones that contain the requested stream s' including the case of decoding with only intra-session network coded packets), we can estimate the expected number of packets that the peer needs to collect for decoding from each combination t and then take the minimum of all expectations.

IV. Distributed rate allocation

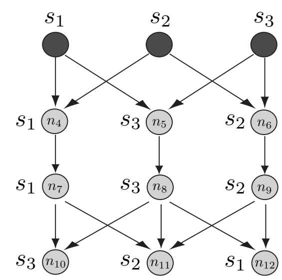
In this section we present the distributed rate allocation algorithm that allows to solve the initial rate allocation problem stated in Section II-B in a decentralized manner with the help of equivalent flow representations.

First, the peer obtains all the necessary information from its neighborhood following the communication protocol described in Section II-A. It then solves the rate allocation problem independently of the other peers and without any centralized control.

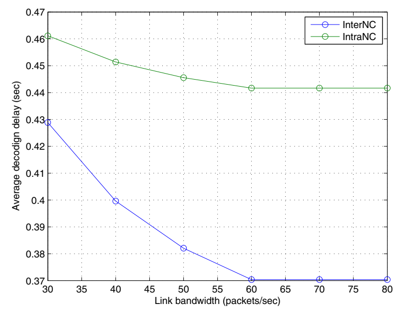
The solution to the delay minimization problem is obtained in the following way. The original problem is decomposed into a set of convex subproblems. The decomposition is based on the fact that, for a given rate allocation the peer i and its children peers $j, j \in \mathcal{D}_i$ decode most of the time from the same combination of packets t_i and $t_j, j \in \mathcal{D}_i$, respectively. Assuming that the peer i and its children peers $j,\ j \in \mathcal{D}_i$ decode from the specific packet combinations t_i and $t_j,\ j \in \mathcal{D}_i$ respectively, we can first determine the optimal rate allocation vector $\boldsymbol{r} = (r_{ki}^t, r_{ij}^t), \forall k \in \mathcal{A}_i, \forall j \in \mathcal{D}_i, \forall t \in \mathcal{T}$ for that particular tuple of packet combinations $(t_i, t_{j, j \in \mathcal{D}_i})$ by solving the following optimization problem

$$\begin{aligned} &\arg\min_{\boldsymbol{r},\boldsymbol{x}} \frac{1}{|\mathcal{D}_i|+1}\Big(d_i \max_{s \in \mathcal{S}_{t_i}} \frac{N_s}{q_{i,s}^{t_i}(\boldsymbol{x}_i)} + \sum_{j \in \mathcal{D}_i} d_j \max_{s \in \mathcal{S}_{t_j}} \frac{N_s}{q_{j,s}^{t_j}(\boldsymbol{x}_j)}\Big) \\ &\text{s.t. } \boldsymbol{r} \in \mathcal{R} \text{ and } \sum_{s' \in \mathcal{S}_{t'}} x_{n,s'}^{t'} = p_n^{t'},\ \forall t' \in \mathcal{T}_t,\ \forall n \in \{i \cup \mathcal{D}_i\} \end{aligned} \quad (19)$$

Recall that $p_i^t = \frac{\sum_{k \in \mathcal{A}_i} r_{ki}^t}{C_i^d}$ and $p_j^t = \frac{\hat{R}_{j \setminus i}^t + r_{ij}^t}{C_j^d}, j \in \mathcal{D}_i$. The subproblem of finding the optimal rate allocation for a specific tuple of packet combinations is convex. The optimal solution to the original problem in Eq. (4) can then be determined



(a)



(b)

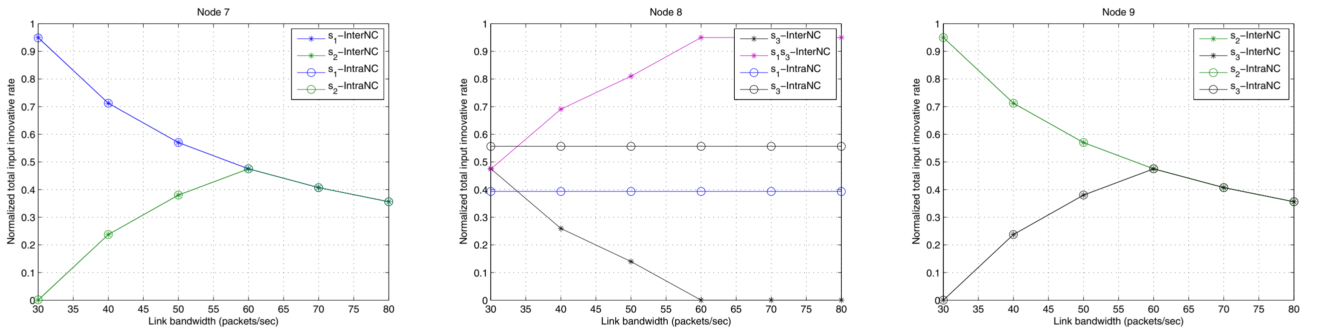
Fig. 1: (a) Toy P2P network topology where 3 streams are concurrently transmitted to the peer nodes. (b) Average decoding delay for the toy network topology depicted in Fig. 1a versus the bandwidth of the links connecting nodes n_4, n_7 and n_6, n_9.

by combining the results of the subproblems and choosing the rate allocation that yields the minimum average delay. In order to compensate for the myopic behavior of the peers and fully utilize the available network resources, the peer later maximizes the total throughput in terms of innovative packet rate while preserving the optimal rates obtained from the delay minimization step. This additional step boosts the performance of the optimization algorithm as it enables forwarding of packets that are potentially useful for other peers rather than the children peers of the peer that performs the rate optimization.

The algorithm runs periodically in every peer and independently from the other peers in the network. This allows to easily adapt the rate allocation to possible changes that may occur in the network.

V. Performance evaluation

In this section, we analyze the performance of our proposed distributed inter-session rate allocation algorithm for data transmission in overlay P2P networks. We consider the transmission of multiple concurrent data streams and evaluate the proposed scheme in terms of the average decoding delay for one generation of packets. We compare the performance of our scheme, henceforth denoted as “InterNC” (Inter-session Network Coding) to a baseline intra-session network coding rate allocation scheme “IntraNC” (Intra-session Network Coding). The latter is a modification of the proposed method where network coding across different sessions in the network nodes is not allowed. The convex optimization problem in Eq. (19)



(a) (b) (c)

Fig. 2: Normalized total input innovative packet rate for nodes (a) n_7, (b) n_8, (c) n_9 versus the bandwidth of links connecting nodes n_4, n_7 and n_6, n_9 for the topology depicted in Fig. 1a. The schemes under comparison are the distributed InterNC and the distributed IntraNC rate allocation algorithms.

is solved using the CVX Matlab-based modeling package [10].

We first evaluate the performance of the proposed distributed inter-session rate allocation algorithm for the network depicted in Fig. 1a. The network consists of 3 source nodes and 9 peer nodes. Next to each node we note the stream that it wants to receive. The packet loss rate is set to 5% on all links. The bandwidth of the links that originate from the sources, as well as of the link connecting nodes n_5 and n_8 is set to 30 packets/sec. The bandwidth of the links that originate from nodes n_7, n_8 and n_9 is set to 60 packets/sec. The generation size for all 3 sources is 10 packets. We should note that, even though we have chosen equal generation sizes for all streams, this is not imposed by our formulation and different generation sizes can be considered.

Fig. 1b presents the evolution of the average delay of the network clients with respect to the bandwidth of the links connecting nodes n_4, n_7 and n_6, n_9 for the schemes under comparison. We can observe that even for low link rates the proposed distributed InterNC rate allocation scheme performs better than the distributed IntraNC scheme. The gains come from the fact that with the proposed scheme nodes can combine packets from different sessions on bottleneck links, whereas with intra-session network coding only, the performance is limited by the presence of low rate links that cannot serve all the clients at the same time. As the link rates increase, higher gains in terms of delay can be noticed for our proposed InterNC scheme, as more packets are combined across different streams. On the contrary, the distributed IntraNC scheme fails to deal efficiently with the bottleneck created on the link between the nodes n_5 and n_8 and the slight improvement of the average decoding delay comes only from the increase of the rate at which packets are supplied to node n_{11}.

Our conclusions regarding the average decoding delay can be further supported by examining the innovative rate that is achieved by the schemes under comparison. Fig. 2 illustrates the normalized total innovative input packet rate of nodes n_7, n_8 and n_9. The normalization is done with respect to the total input bandwidth of the peer. s_j denotes a flow of intra-session network coded packets of stream s_j, whereas $s_i s_j$ represents the combined flow of inter-session network coded packets from streams s_i and s_j. The flows that are zero in the whole range of link bandwidths are omitted from the figures.

As we can notice from Fig. 2b the link between nodes n_5 and n_8 has to be shared by the flows s_1 and s_3 when only intra-session network coding is allowed as this is the only path from where nodes n_{10} and n_{12} can receive their requested flows. Thus, when the bandwidth of the links between nodes n_4, n_7 and n_6, n_9 increases, the average decoding delay of nodes n_{10} and n_{12} cannot be improved as they receive intra-session network coded packets at constant rates regardless of the bandwidth variations. The only reason for the slight improvement of the average delay that we observe in Fig. 1b is the additional supply of packets of stream s_2 to node n_{11} from node n_7 (Fig. 2a).

When inter-session network coding is allowed, the average performance of the network is enhanced mainly by the combination of flows s_1 and s_3 on the bottleneck link between nodes n_5 and n_6. As we can see in Fig. 2b the node n_8 allocates part of the input bandwidth to the combined flow $s_1 s_3$ whereas the rest is allocated to the intra-session network coded flow s_3. As the node n_9 starts to provide more intra-session network coded packets of flow s_3 to node n_{12} as a result of the increase in the bandwidth, the percentage of the combined flow on the bottleneck link increases and eventually the node n_8 requests only combined packets. At this point both nodes n_{10} and n_{12} manage to receive their requested streams at the rate of the bottleneck link since they receive the other component packets of the combined stream from nodes n_7 and n_9 respectively at the same rate and are able to decode the stream of their interest faster. Thus, we can see that the limitations imposed by the bottleneck link can be overcome by deploying inter-session network coding and utilizing the additional resources of the nodes for receiving packets that can help in decoding the combined streams.

We now test the performance of the proposed scheme for the clustered network depicted in Fig. 3. This network consists of three server nodes and 30 client nodes. The clients are

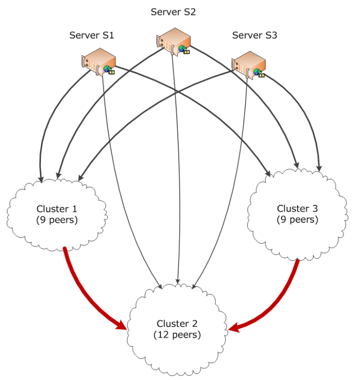


Fig. 3: Cluster network topology.

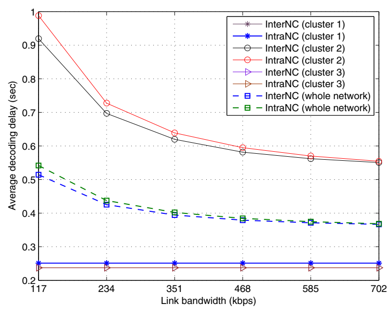


Fig. 4: Performance comparison of the InterNC and IntraNC algorithms regarding the average decoding delay as a function of the links' bandwidth for the examined cluster network.

organized in 3 clusters of 9, 12 and 9 nodes respectively. Each cluster is an irregular directed network generated from a regular network by removing and shifting randomly some of the links [11]. The pruning and shifting probabilities are set to 40% and 20% respectively. The loss rate is set to 5%. Every peer node is assigned one of the available streams. The selection of the stream is done randomly and with equal probabilities among the streams that can reach the peer. Clusters 1 and 3 are connected directly to the servers with links that have capacity 468 kbps each, whereas cluster 2 is connected to clusters 1 and 3 through links with a capacity that varies in the interval $[117, 702]$ kbps. Moreover, the cluster 2 receives some packets directly from the sources through low speed links that have a capacity of 117 kbps. Finally, the nodes within all the clusters are interconnected with high speed links of 1.6 Mbps. The packet size is fixed to 1500 bytes including the network coding header. We consider that the generation sizes for all data sources are equal to 10 packets. All the results are averages of 10 random realizations of the network.

Fig. 4 illustrates the average decoding delay per cluster as well as for the whole network depicted in Fig. 3 with respect to the bandwidth of the links that connect the cluster 2 to clusters 1 and 3. The schemes under comparison is the proposed distributed InterNC rate allocation algorithm and the baseline distributed IntraNC scheme. We can observe that, by allowing peers to combine different sessions, we can achieve lower decoding delays than those that we are able to have with the intra-session network coding scheme. As can be seen in Fig. 4 the gain is observed in cluster 2 that does not have sufficient resources to provide enough intra-session network coded packet to all the peers, contrarily to clusters 1 and 3 where all the peers are able to acquire all the packets directly from the sources. Thus, inter-session network coded packets are requested on the bottleneck links connecting cluster 2 to clusters 1 and 3 in order to serve more peers in the network, whereas the additional packets that are provided through the low capacity links that connect cluster 2 to the sources are used to decode faster the combined packets.

VI. Conclusions

We have proposed a novel distributed rate allocation algorithm for delivery of multiple concurrent data streams in overlay networks. The algorithm is based on inter-session network coding. The network peers decide locally on the optimal coding decisions and rates for each combination of packets that they request from their parent peers. The decisions are based on the minimization of the average expected delay of the peer and its children peers and require only a minimal communication overhead. We show that the initial non-convex rate allocation problem can be decomposed into a set of simpler convex problems with the help of the equivalent flow representation; the final rate allocation can be obtained by combining the results of the subproblems. The evaluation of the proposed algorithm demonstrates the benefits of utilizing inter-session network coding in terms of the achieved delays and efficient exploitation of network resources.

References

[1] C. Wu and B. Li, "rStream: Resilient and Optimal Peer-to-Peer Streaming with Rateless Codes," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 1, pp. 77–92, Jan. 2008.

[2] X. Zhang, J. Liu, B. Li, and Y.-S. P. Yum, "CoolStreaming/DONet: a Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming," in *Proc. IEEE INFOCOM*, vol. 3, March 2005, pp. 2102 – 2111.

[3] M. Ponec, S. Sengupta, M. Chen, J. Li, and P. Chou, "Optimizing Multi-Rate Peer-to-Peer Video Conferencing Applications," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 856–868, Oct. 2011.

[4] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. Chou, "Peer-to-Peer Streaming Capacity," *IEEE Trans. Information Theory*, vol. 57, no. 8, pp. 5072–5087, Aug. 2011.

[5] M. Wang and B. Li, "R^2: Random Push with Random Network Coding in Live Peer-to-Peer Streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, Dec. 2007.

[6] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized Distributed Video Delivery With Randomized Network Coding," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 776–787, Aug. 2011.

[7] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. of the 41st Allerton Conf. on Communication, Control and Computing*, Monticello, IL, USA, Oct. 2003.

[8] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I2NC: Intra- and Inter-Session Network Coding for Unicast Flows in Wireless Networks," in *IEEE INFOCOM*, Shanghai, April 2011.

[9] M. Heindlmaier, D. Lun, D. Traskov, and M. Médard, "Wireless Inter-Session Network Coding - An Approach Using Virtual Multicasts," in *Proc. of IEEE Int. Conf. on Communications*, Kyoto, June 2011.

[10] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Apr. 2011.

[11] N. Thomos and P. Frossard, "Network Coding of Rateless Video in Streaming overlays," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1834–1847, Dec. 2010.