DISTRIBUTED MEDIA RATE ALLOCATION IN OVERLAY NETWORKS

Dan Jurca and Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute
CH-1015 Lausanne, Switzerland

ABSTRACT

We address the problem of distributed path selection and rate allocation for media streaming in overlay networks. Under the assumption that each node has only a local view of the network, we propose a distributed algorithm for joint path selection, and rate allocation, in order to minimize the end-to-end media distortion. The distributed algorithm performs iteratively, by greedy rate allocation for all incoming media flows on the outgoing links at each intermediate node. Our algorithm is shown to converge to the optimal rate allocation solution in a very small number of iterations, and to outperform heuristic distributed rate allocation mechanisms for a number of random network topologies.

1. INTRODUCTION

As the internet is still far from providing any widely deployed guarantee of service solution, efficient media streaming strategies have to be devised to get the best out of the network infrastructure. Lately, multipath streaming emerged as an interesting solution to overcome some of the lossy internet path limitations [1, 4]. It allows for an increase in streaming bandwidth, by balancing the load over multiple network paths between the media server and the client. It also provides means to limit packet loss effects, when combined with error resilient streaming strategies and scalable encoding capabilities of the latest encoding standards. Multipath streaming can be applied in the context of CDNs, overlay networks, or wireless and peer-to-peer scenarios, where the client has easy access to the media sources, through various network paths.

Most of the scientific work dedicated to multipath streaming focuses on the streaming process itself (media scheduling aspects), but generally not towards finding which paths should ideally be used for the streaming application, given an available network topology between a server and a client. Most of these works rely on classic routing algorithms that find the best path (or set of paths) given some established network metrics. While this may be optimal in terms of network utilization, it is certainly suboptimal from the point of view of the media streaming application [6].

Our prior work [2] provides a general framework for the analysis of joint path and rate allocation in multipath streaming, driven by media-specific metrics. We consider a network model composed of multiple flows, and a streaming server that can adapt the media source rate to the transmission conditions (by scalable coding, or transcoding, for example). Given the knowledge of the network parameters, the server computes the optimal set of paths for transmission, along their optimal rate allocation. In this paper we release the assumption on network status knowledge at the server, and we propose a distributed path selection and rate allocation algorithm. Previous works in distributed video streaming [3, 7, 5] deal with resource allocation and scheduling on multiple, a priori chosen streaming paths, with the final goal of minimizing the overall distortion perceived by the media clients. None of these works address the problem of an optimal choice of streaming paths on an available network graph, and their corresponding rate allocation.

The rest of this paper is organized as follows. In Section 2, we identify the streaming scenario under consideration and briefly present the optimization problem. Section 3 presents our distributed approach towards solving the optimization problem. We discuss the performance of our algorithm in Section 4 and present the simulation results in Section 5. Finally, we conclude in Section 6.

2. PRELIMINARIES

2.1. Video Model

The end-to-end distortion, as perceived by the media client, can generally be computed as the sum of the source distortion, and the channel distortion. The source distortion D_S is mostly driven by the encoding or streaming rate R, and the media sequence content, whose characteristics influence the performance of the encoder. The channel distortion D_L is dependent on the loss probability π of video information, and the sequence characteristics. It is roughly proportional to the number of video entities (e.g., frames) that cannot be decoded. The end-to-end distortion can thus be written as:

$$D = D_S + D_L = \alpha \cdot R^\xi + \beta \cdot \pi,$$

where $\alpha, \beta \in \Re^+$ and $\xi \in [-1, 0]$ are parameters that depend on the video sequence. This distortion model is a
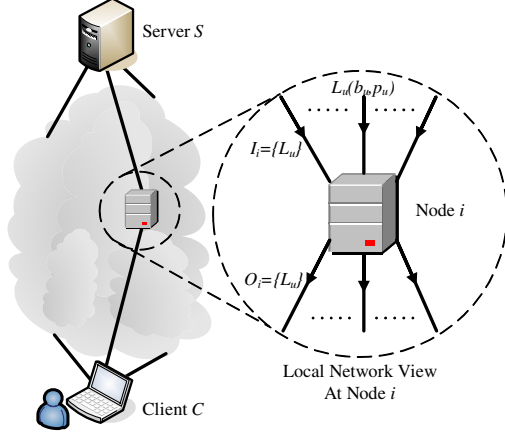
Fig. 1. Overlay Network Scenario and Network View at Node N_i.

simple and general approximation, suitable for most common streaming strategies where the number of packet per frame is independent of the encoding rate. A validation of this model through video experiments can be found in [2].

2.2. Network Model

We consider that the media streaming application uses a large scale overlay network, modelled as a fully connected directed acyclic graph $G(V, E)$ between the streaming server S and the client C (Figure 1). V is the set of nodes in the network, and E is the set of links. Each link $L_u \in E$ has two associated positive metrics:

- the available bandwidth $b_u > 0$ expressed in some appropriate unit (e.g., kbps), and,

- the average packet loss probability $p_u \in [0, 1]$, related to an iid packet loss process, assumed to be independent of the streaming rate.

Finally, we define P_i, $1 \leq i \leq M$, as an end-to-end path between S and C in G, with parameters b_i and p_i being the end-to-end bandwidth and loss probability respectively, and M the total number of distinct paths. Flow $\mathcal{F}_i$ transmitted on path P_i has a streaming rate $r_i \leq b_i = \min_{L_u \in P_i}(b_u)$, and is affected by the loss probability $p_i = 1 - \prod_{L_u \in P_i}(1 - p_u)$.

2.3. Distributed Optimization Problem

When no single node $N_i \in V$ (including S), is aware of the entire network topology G, we want to find the optimal path selection and flow rate allocation that minimizes the overall distortion D, perceived by the client.

Under the assumption that the streaming rate can be controlled (e.g. scalable encoding, transcoding or packet filtering), the server S adapts to the available rate of the

paths/flows used for transmission: $R = \sum_{i=1}^{n} r_i$. In the same time the overall loss process that affects the streaming application is computed as $\pi = \dfrac{\sum_i p_i \cdot r_i}{\sum_i r_i}$.

Our distributed joint routing and rate allocation problem translates into finding the optimal rate allocation vector $\vec{R^*} = [r_1, ...r_M]^*$ that minimizes the distortion metric D:

$$\vec{R^*} = \arg \min_{\vec{R}}(\alpha \cdot R^\xi + \beta \cdot \pi) \qquad (1)$$

The goal of the following sections is to provide a new algorithm that solves the optimization problem in a distributed way, when each node has only a local view of the network.

3. DISTRIBUTED RATE ALLOCATION

In our previous work [2], we show that the optimal rate allocation, once the parameters of all paths P_i are known by S, can be achieved by a greedy path selection algorithm that starts with the paths affected by the smallest end-to-end loss process. In the same time, we show that the rate of joint bottleneck links between multiple network paths should be split among media flows in a similar greedy fashion, and that once a path P_i is chosen for transmission, it is optimal from the media application perspective, to stream at rate $r_i = b_i$.

The distributed algorithm presented in this paper iteratively computes the optimal rate allocation based only on the local network view at individual nodes. In each round of the algorithm, each node computes a greedy rate allocation for the incoming flows, on the outgoing links. Media flows requested by C, are reserved by S, only if they can be accommodated by all intermediate nodes. Each round, the algorithm adds media flows to a partially optimal flow allocation, finally converging to the optimal path selection and rate allocation solution.

In order to obtain exact bounds on the performance of our distributed algorithm, we assume that any two nodes connected in G can exchange control messages through a reliable bidirectional control channel, and that all nodes are synchronized.

The control scheme can be described as follows. Consider any node $N_i \in V$. Let I_i and O_i be the sets of incoming and respectively outgoing links to/from N_i. We assume that N_i knows or estimates b_u and p_u, $\forall L_u \in O_i$. Let $P_i(b_i, p_i, f_i, d_i)$ be a path connecting S and N_i, where b_i and p_i represent the path parameters, and $f_i, d_i = \{0, 1\}$ are decision flags used by our algorithm. f_i is a path reservation flag that can only be set/reset by C, respectively S, while d_i is a decision flag that can be updated by any intermediate node in path P_i. Let $\mathcal{P}_i^u = \{P_i\}$ be the set of distinct paths between S and N_i, arriving on $L_u \in I_i{}^1$, and

---

[1]Note that two distinct paths P_i and P_j may not necessarily be fully disjoint. They may share one or more network links.

Algorithm 1 Distributed Rate Allocation

server S:	node N_i:
upon receive $Recv^u, \forall L_u \in O_S$:	upon receive $Resv^u, \forall L_u \in O_i$:
1. compute $\mathcal{P}_C$;	1. compute $\mathcal{P}_C^u$ from $Resv^u \setminus \mathcal{P}^*$;
2. compute $\mathcal{R}$;	2. $\forall P_i(b_i, p_i, 1, 1) \in \mathcal{P}_C^u$:
3. if $\mathcal{R} = \emptyset$, return $\mathcal{P}^*$.	set $d_i = 0$ if $b_u < b_i, L_u \in P_i$,
4. else $\mathcal{P}^* = \mathcal{P}^* \cup \mathcal{R}$;	according to a greedy allocation;
compute G'_S and $\mathcal{P}_S^u$;	3. send $Resv^u, \forall L_u \in I_i$.
send $Path^u, \forall L_u \in O_S$.	
node N_i:	client C:
upon receive $Path^u, \forall L_u \in I_i$:	upon receive $Path^u, \forall L_u \in I_C$:
1. compute G'_i;	1. compute $\mathcal{P}_C$;
2. compute $\mathcal{P}_i$ on G'_i;	2. compute $\vec{R}^*$;
3. compute $\mathcal{P}_k^u, \forall L_u \in O_i$;	3. $\forall P_i, r_i^* \neq 0$ set $f_i = d_i = 1$;
4. send $Path^u, \forall L_u \in O_i$.	4. send $Resv^u, \forall L_u \in I_C$.

---

let $\mathcal{P}_i = \bigcup_{u:L_u \in I_i} \mathcal{P}_i^u$. We assume that the set $\mathcal{P}_i$ is ordered according to the increasing value of the path loss probabilities p_i. Initially, $\mathcal{P}_S = \emptyset$.

We define $\oplus$ as the path extension operator between path $P_i \in \mathcal{P}_i$ and link $L_u \in O_i$. We compute the parameters of the extended path $P_k = P_i \oplus L_u$ as $b_k = \min(b_i, b_u)$, and $p_k = 1 - (1 - p_i)(1 - p_u)$. $\mathcal{P}_k^u = \mathcal{P}_i \oplus L_u$ is the complete set of distinct path extensions $\forall P_i \in \mathcal{P}_i$ and $L_u \in O_i$, where b_u is greedily allocated to paths P_i is ascending order of their p_i. Paths P_k with $b_k = 0$ are omitted.

Let $\mathcal{P}^*$ be the set of successfully reserved paths between S and C, and let $\mathcal{R}$ be the paths reserved by the algorithm during one round. $\mathcal{R} = \{P_C\}, \forall P_C(b_C, p_C, 1, 1) \in \mathcal{P}_C$, until the first $P_C(b_C, p_C, 1, 0)$. Initially $\mathcal{P}^* = \mathcal{R} = \emptyset$.

We denote G_i as the partial network topology view available at N_i (Figure 1), and $G'_i = G_i \setminus \mathcal{P}^*$, the residual network view at N_i after allocating bandwidth for the reserved paths in $\mathcal{P}^*$.

We finally define two control messages, exchanged between nodes during the execution of the algorithm, on the control channel. $Path^u = \{\mathcal{P}_k^u \bigcup \mathcal{P}^*\}$ is the path discovery message initiated by S, and forwarded by each N_i, on the control channels associated with $L_u \in O_i$, according to `Algorithm 1`. $Resv^u = \{\mathcal{P}_i^u \bigcup \mathcal{P}^*\}$ is the path reservation message initiated by C upon the reception of $Path$ messages, and forwarded by N_i towards S, on the backward control channels $\forall L_u \in I_i$.

`Algorithm 1` finally formalizes the execution of our distributed rate allocation algorithm. It starts at the server, at Step 4, with $\mathcal{P}^* = \mathcal{R} = \emptyset$, and returns the optimal set of transmission paths, along with their rate allocation in a finite number of rounds. The resulting set of paths $\mathcal{P}^*$ will ensure the optimal video distortion at the receiving client.

4. ANALYSIS AND DISCUSSION

Now we analyze the performance of `Algorithm 1`, by stating some of its most important properties. For the sake of clarity, we assume that during one run of the algorithm,

the network remains stable.

Property 1. If the paths requested by C do not share any bottleneck joint link L_u, `Algorithm 1` converges in one round.

Let b_i be the rate of requested path P_i. According to the path extension operation, $b_i \leq b_u, \forall L_u \in P_i$. As long as requested paths do not share a bottleneck link, we also have $\sum_{i:L_u \in P_i} b_i \leq b_u$, hence all flows on paths P_i can be reserved at intermediate nodes. If $\sum_{i:L_u \in P_i} b_i > b_u$, then link L_u is a bottleneck link for flows on paths P_i.

Property 2. During each round, `Algorithm 1` reserves in the residual network G', at least the smallest loss probability flow P_i.

Since at each intermediate node N_i, the rate allocation for the requested flows is performed greedily, the flow with the minimum loss probability always has priority. From the path extension operation we have $b_i \leq b_u, \forall L_u \in P_i$, hence P_i will be reserved at each intermediate node N_i on the path.

Property 3. `Algorithm 1` terminates in maximum n rounds, where n is the number of requested flows.

This result is a consequence of Property 2. At each round, the algorithm reserves at least one flow, and the available rate of the paths in the residual network decreases. Hence, on subsequent rounds of the algorithm, C will not be able to request an infinite number of flows.

Finally, we discuss the convergence speed of `Algorithm 1`. Observe that one round of the algorithm requires one message exchange between S and C, on all available paths. Hence, the time required by one round is in the order of the RTT of the slowest paths in the network. The computations at intermediate nodes and at S and C are trivial and their duration can be omitted. Since the optimal number of used paths is usually small [2], we expect our algorithm to converge extremely fast.

5. SIMULATIONS

We evaluate the performance of the distributed rate allocation algorithm in various random network scenarios. We generate two types of network topologies: (i) *Wireless* network graphs, with low bandwidth and high error probability for the intermediate links; and (ii) *Hybrid* network scenarios, where the server is connected to the wired infrastructure (high rates, low loss probabilities), and the client can access the internet via multiple wireless links.

For both scenarios, we generate 500 random graphs, with 10 nodes each. Any two nodes are directly connected with a probability γ. The parameters for each link are randomly chosen according to a normal distribution, in the interval $[R_{min}, R_{max}]$, for the bandwidth, and respectively $[p_{min}, p_{max}]$ for the loss probability. The parameters for the wired and wireless links are presented in Table 1.
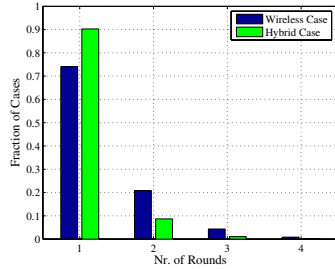
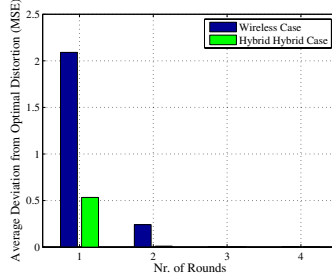**Fig. 2**. Number of Rounds needed for Convergence of Algorithm 1.

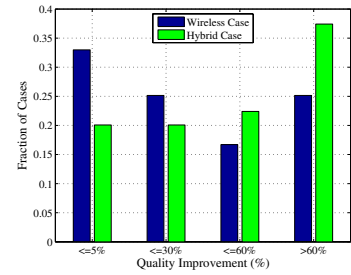**Fig. 3**. Convergence of the Distributed Algorithm, in terms of Distortion.

Fig. 4. Quality Improvement vs. Heuristic Rate Allocation.

Table 1. Parameters for Random Graph Generation

Parameter	Wired Links	Wireless Links
Connectivity Probability γ	0.4	0.6
R_{min}	$10^6 bps$	$10^5 bps$
R_{max}	$3 \cdot 10^6 bps$	$7 \cdot 10^5 bps$
p_{min}	10^{-4}	10^{-3}
p_{max}	$5 \cdot 10^{-3}$	$4 \cdot 10^{-2}$

First we analyze the number of rounds in which the distributed algorithm converges to the optimal rate allocation given by [2]. The results for both network scenarios are presented in Figure 2. We observe that the great majority of tested scenario need less than three iterations in order to reach the optimal rate allocation. This shows that our algorithm performs very fast and needs only a small number of control messages.

Next, we test the convergence of the distributed algorithm in the two network scenarios, in terms of video distortion, compared to the final optimal distortion value. For each protocol round, we compute the average distortion difference for the corresponding rate allocation. The results are presented in Figure 3. We observe, that even from the first protocol round, the obtained partial solution is very close to the optimal one, and that starting with the second round, virtually there is no difference between the obtained results and the optimal one. Again, this emphasizes the speed of convergence of our proposed distributed algorithm.

Finally, we compare the results obtained by the proposed algorithm, to the ones obtained by a simpler distributed heuristic which forwards the incoming network flow at each intermediate node on the best outgoing link in terms of loss probability. We compute the fraction of simulation scenarios for which our results bring an improvement of $[0 - 5\%]$, $[5 - 30\%]$, $[30 - 60\%]$ and above 60% compared to the heuristic ones. These results are presented in Figure 4. We observe that, for both network scenarios, our algorithm obtains significantly better results in more that 70% of the cases. This motivates the extra control overhead introduced by our algorithm, needed in order to obtain the final optimal rate allocation.

6. CONCLUSIONS

We present an optimal distributed path selection and rate allocation algorithm, for media streaming in overlay networks. By performing a distributed rate allocation operation at each intermediate node, we release the assumption of centralized network knowledge at one individual node. Simulation results show that our algorithm converges extremely fast to the optimal solution, hence requires a very small execution time and a low number of control messages. In the same time, compared to other simpler distributed heuristics, our results in terms of distortion are significantly better in the great majority of cases. In future work we intend to extend the analysis of our distributed rate allocation algorithm in typical network topologies, possibly with dynamic resources.

7. REFERENCES

[1] L. Golubchik, J. Lui, T. Tung, A. Chow, and W. Lee. Multipath continuous media streaming: What are the benefits? *ACM Journal of Performance Evaluation*, 49(1-4):429–449, Sept 2002.

[2] D. Jurca and P. Frossard. Media-specific rate allocation in heterogeneous wireless networks. In *Proceedings of the International Packet Video Workshop*, 2006.

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

[4] Y. Li, S. Mao, and S. S. Panwar. The case for multimedia transport over wireless ad hoc networks. In *Proceedings of IEEE/ACM BroadNets*, October 2004.

[5] T. Nguyen and A. Zakhor. Multiple sender distributed video streaming. *IEEE Transactions on Multimedia*, 6(2):315–326, April 2004.

[6] S. Savage, A. Collins, and E. Hoffman. The end-to-end effects of internet path selection. In *Proceedings of ACM SIGCOMM*, 1999.

[7] X. Zhu and B. Girod. Distributed rate allocation for multistream video transmission over ad-hoc networks. In *Proceedings of IEEE ICIP*, 2005.