

Scalability of Reliable Group Communication Using Overlays

François Baccelli, Augustin Chaintreau, Zhen Liu, Anton Riabov, Sambit Sahu

▶ **To cite this version:**

François Baccelli, Augustin Chaintreau, Zhen Liu, Anton Riabov, Sambit Sahu. Scalability of Reliable Group Communication Using Overlays. [Research Report] RR-4895, INRIA. 2003. inria-00071687

HAL Id: inria-00071687

https://inria.hal.science/inria-00071687

Submitted on 23 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Scalability of Reliable Group Communication Using Overlays

François Baccelli — Augustin Chaintreau — Zhen Liu — Anton Riabov — Sambit Sahu

N° 4895

Juillet 2003

THÈME 1



ISSN 0249-6399 ISRN INRIA/RR--4895--FR+ENG



Scalability of Reliable Group Communication Using Overlays

François Baccelli* , Augustin Chaintreau† , Zhen Liu‡ , Anton Riabov§ , Sambit Sahu¶

Thème 1 —Réseaux et systèmes
Projet TREC

Rapport de recherche n° 4895 —Juillet 2003 —40 pages

Abstract: This study provides some new insights into the scalability of reliable group communication mechanisms using overlays. These mechanisms use individual TCP connections for packet transfers between end-systems. End-systems store incoming packets and forward them to downstream nodes in the multicast tree using different unicast TCP connections. In this paper we assume that buffers in end-systems are large enough for the storage. It is shown that the throughput of the reliable overlay group communication scales in the sense that for all multicast tree sizes and topologies, the group throughput is strictly positive provided the saturation throughputs of all unicast connections are bounded away from 0. This is in contrast with the IP supported multicast paradigm where reliable protocols have vanishing throughput when the group size tends to infinity.

The scalability of packet delay and buffer occupancy is then investigated. In the absence of additional control, the occupancy of the buffer and the latency in the end-systems explode with time. It is then shown that a proactive rate throttle mechanism implemented at the source leads to finite packet latency and buffer occupancy in any end-system of the network provided certain moment conditions are satisfied by cross traffic in the routers.

Some of the theoretical scalability results are based on methods stemming from statistical physics which are called hydrodynamic limits. They are validated by a set of experiments on the Internet and by simulations based on the max-plus representation of TCP which allow the handling of very large overlay networks. The paper also discusses a few important practical matters pertaining to the shaping of the trees and the required control mechanisms.

Key-words: Multicast, Application layer, Network overlay, Optimization, TCP tandem, Lattice animal, Hydrodynamic limit.

* INRIA-ENS, 45 rue d'Ulm 75005, Paris, France, francois.baccelli@ens.fr
† INRIA-ENS, 45 rue d'Ulm 75005, Paris, France, augustin.chaintreau@ens.fr
‡ IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, zhenl@us.ibm.com
§ IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, riabov@us.ibm.com
¶ IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, sambits@us.ibm.com

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 —Télécopie : +33 1 39 63 53 30

Extensibilité de Mécanismes Fiables de Communication Multipoint Fondés sur des Réseaux Overlay

Résumé : Nous étudions l'extensibilité de mécanismes de communication de groupe multipoint utilisant des réseaux de type "overlay". Dans de tels réseaux, les paquets sont transmis aux utilisateurs du groupe via des connections TCP point à point distinctes. Les utilisateurs stockent les paquets arrivant d'un utilisateur en amont et les retransmettent aux utilisateurs en aval dans l'arbre de diffusion en utilisant plusieurs connections TCP simultanées. Nous faisons ici l'hypothèse que les mémoires tampons des utilisateurs sont suffisament grandes pour procéder à ce stockage. Nous montrons qu'alors le débit de la communication de groupe est strictement positif dès que le débit de saturation de chacune des connexions point à point est lui même supérieur à une constante positive. Ce résultat vaut pour n'importe quelle taille de groupe et n'importe quelle topologie d'arbre de diffusion. Ceci est à rapprocher des résultats connus sur le multipoint fiable supporté directement par IP et par un mécanisme de fenêtre global, où le débit de la communication de groupe tend vers zero pour des groupes dont la taille tend vers l'infini.

Dans un deuxième temps, nous étudions l'extensibilité du délai des paquets, ou de manière équivalente celle du remplissage des mémoires tampons des utilisateurs. Nous montrons que si aucun contrôle n'est effectué à la source, ces deux grandeurs explosent en temps long. Dans le cas où le taux de départ de la source est régulé, nous montrons que sous certaines conditions de moments du traffic transverse observé sur les routeurs, le remplissage de la mémoire est au contraire localement fini pour n'importe quelle taille de groupe,

Nous illustrons ces résultats par des expériences effectuées directement sur l'Internet, et par des simulations numériques basées sur une représentation max-plus du réseau overlay et de ses connections TCP, qui permet de simuler des réseaux de très grande taille. Nous justifions aussi ces résultats de manière mathématique au moyen de méthodes de physique statistique connues sous le nom de limites hydrodynamiques. Les questions pratiques posées par la formation des arbres de diffusion, et par le développement du protocole de contrôle sont aussi discutées.

Mots-clés : Communication multipoint, Réseau overlay, Optimisation, Connexions TCP en tandem, Animaux de Grille, Limite hydrodynamique.

1 Introduction

Reliable group communication has remained an important research problem for the last decade. Significant effort has been spent on the design and the evaluation of reliable multicast transport protocols, see for example [11, 6, 20] and the references therein. However, such IP supported reliable multicast schemes have been facing two major obstacles. First, there is no wide spread deployment of IP multicast in the Internet. Second, it has been shown in various studies [27, 8] that group throughput vanishes when the group size increases, thus suffering from scalability issues.

Recently an alternative approach that uses overlays of end-systems has been proposed to support group communications. In this approach, end-systems form an overlay by establishing point-to-point connections in between end-systems, where each node forwards data to downstream nodes in a store-and-forward way. The multicast distribution tree is formed at the end-system level. Such a paradigm is referred to as end-system multicast, or application-level multicast, or simply multicast using overlays. Various studies have been conducted with the primary focus on the protocol development for efficient overlay tree construction and maintenance, such as Narada [9], Yoid [13], ALMI [25], Host Multicast [32], NICE [5], Delauney graph [21]. Some other work in peer to peer network is also related to the tree construction in application level multicast, see e.g. Chord [30] and CAN [26].

Reliable multicast can also be implemented in overlay using point-to-point TCP connections. In Overcast [17], HTTP connections are used in between end-systems. In RMX [7], TCP sessions are directly used. The main advantage of such approaches is the ease of deployment. In addition, [7] argues that it is possible to better handle heterogeneity in receivers because of hop-by-hop congestion control and data recovery.

However there is a lack of understanding of the performance of TCP protocol when used in an overlay based group communication to provide reliable content delivery. Although studies in [7, 17] have advocated the usage of overlay networks of TCP connections, they do not address the scalability concerns, in terms of throughput, buffer requirements and latency of content delivery. In [31], the scalability issue is investigated for a model of overlay based reliable multicast. The authors considered a TCP-friendly congestion control mechanism with fixed window-size for the point-to-point reliable transfer. Some simulation results were presented to show the effect of the size of end-system buffers on the group communication throughput.

In our work, we provide a mathematical framework based on the max-plus representation of TCP to address the scalability of overlay group communication when TCP is used for providing reliable content delivery. We examine the scalability of three variables with respect to the group size and the end-system network connectivity: 1) the throughput obtained by the group communication, 2) the delay for packets to reach end-systems and 3) the buffer contents at the end-systems. For this we propose theoretical investigations, experimentations in the Internet, and simulations of large networks.

In order to address these questions, we first provide a framework to study the behavior of a group of TCP sources in a chain and then in a tree topology based on the max-plus algebra (see e.g. [4] and the references therein) under the assumption of infinite buffer space at each intermediate end-system.

Using this framework, we establish a first result that states that irrespective of the group size and the behavior of the underlying network connecting the end-systems in the overlay network, there exists a strictly positive group throughput, provided the saturation throughputs (defined in the paper) of all unicast connections are bounded away from 0. This contrasts with the known result established in the case of IP-supported multicast for reliable group communication about the non-scalability of such protocols. In addition, we establish the maximum possible throughput achievable for a set of receivers and the conditions that are required to achieve this maximum throughput.

We then examine whether there exist mechanisms that can achieve both a non-zero group throughput and a finite packet delay, as well as a finite buffer occupancy in each end-system of an overlay with a general tree topology. We propose and analyze a pro-active mechanism which throttles the sending rate of the source. We show that there exists a critical rate such that when the sending rate at the origin node is limited to this critical rate, one can guarantee (in some sense to be defined precisely in the paper) finite buffers and latency at all the nodes in the overlay tree. This shows that rate control combined with TCP congestion control mechanism provides a scalable approach in both throughput and buffer occupancy.

Using a prototype implementation of the TCP overlays, we conducted experiments in the Internet to validate these results. In addition to this, we designed a simulator taking advantage of the max-plus representation of TCP connections and allowing one to simulate the transmission of a large number of packets over overlay networks consisting of very large trees. Various simulation results are also presented.

Moreover, we find that in order to maximize the group's throughput, the design of the protocol and the construction of the distribution tree should take into account the *local saturation throughput* (see definition below) of the TCP connections between end-systems.

The paper is organized as follows. Section 2 defines the problems under consideration and presents the notation and the mathematical models. In Section 3, we prove the existence of positive throughput in a tandem of TCP sources with unconstrained buffers at end-systems. This result is then extended to any arbitrary tree configuration of overlay network. In Section 4 we introduce the rate-control based protocol allowing one to bound the buffer occupancy and the latency for any arbitrary group size. In Section 5, we discuss how the theoretical results obtained in the paper can be used for the design of new reliable group communication protocols using overlays. Section 6 summarizes the work.

2 Modeling Overlay Group Communication

2.1 Reliable Overlay Group Communication

At a high level of abstraction, an overlay network can be described as a directed communication graph where the nodes are the end-systems and an edge between any two nodes a and b represents the data forwarding network from node a to node b. An edge in the overlay network represents the path between the two nodes that it connects. While this path may traverse several routers in the physical network (see the models introduced in §2.3) on which a feedback control mechanism is enforced, this level of abstraction considers the path as a mere directed edge in the overlay network.

While it is not required, we will assume that the nodes are connected in a tree topology. The end-systems participate explicitly in forwarding data to other nodes in a store-and-forward way. As illustrated in Figure 1, after receiving data from its parent node in the overlay network, a node will replicate the data on each of its outgoing edges and forward it to each of its downstream nodes in the overlay network.

The topology is typically constructed accounting for forwarding capacity of each node and geographical distance between nodes. Let us examine how TCP may be used in this overlay network to support reliable content delivery. The obvious approach that does not require any changes to the TCP protocol is to use end-point abstraction for every edge, i.e. a TCP connection for every edge in the overlay network. In this model, a node after receiving data will store the data and forward it on a per-connection basis using established TCP connections for each of its downstream nodes in the overlay network. We shall describe a specific implementation of this data forwarding later in this section.

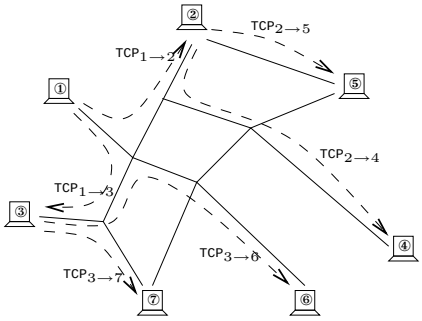


Figure 1: A binary tree overlay network

In such an overlay network, except for leaf nodes, all the other nodes, which store and forward packets, need to provision buffers for the packet forwarding purpose. One buffer is needed at the sender side for each of the TCP connections. Disc space is abundant in typical end-systems such as PCs or work stations. In such systems, the end-system buffers can be provisioned large enough to accommodate the TCP traffic. In the analysis presented in this paper we assume infinite buffer capacity at each end-system. This is in particular justified when each end-system keeps data in local storage system for its own use.

For the case where end-systems do not necessarily need to keep all data in local storage system, and only play the role of a relay system, we propose a proactive mechanism which consists in throttling the send rate of the source in such a way that the buffer occupancies as well as the packet delays are finite in all end-systems. In this case, we study in particular how much buffer space each end-system has to provide on average in order to play its relay role for the group communication.

2.2 Problem Definition and Methodology

We define the group throughput of an overlay network as the minimum sending rate across all its edges. Due to the reasons explained above, the group throughput in an overlay network with TCP sessions on each of the overlay edges will depend upon the network conditions of the underlying paths between the nodes.

We shall examine the behavior of TCP connections arranged in a tree topology to answer the scalability properties of overlay based reliable group communication. Specifically, our study will examine the following:

- how the group's throughput is related to the local saturation throughput of TCP connections at the overlay edges, where this local saturation throughput of TCP connection is defined as the throughput achieved with an infinite backlog of packets at the sender node of this TCP session;
- what conditions are needed to achieve a positive throughput irrespective of group size?
- whether it is feasible to provide any scaling behavior in terms of delay and buffer requirements?
- how one should construct the forwarding tree in order to maximize group throughput?

To investigate the above issues, we resort to both theoretical modeling, simulation and experimentation in the Internet using a prototype implementation. Using previous results on the modeling of TCP via the max-plus algebra [4], we extend the model to analyze the behavior of TCP connections in tandem. Next we apply the results from tandem TCP connections to examine the behavior of TCP sessions in a tree topology. This algebra ([3]) is particularly useful to describe synchronization constraints such as window flow control, the serialization associated with queueing or the fork at end-systems. In the present paper, it will primarily be used in order to reduce some of the questions of interest to longest path problems in certain infinite random graphs along the lines of what was done for other models in [1] and [23].

It should be stressed that most of the techniques of [3] cannot be used directly for the present study. A first reason stems from the fact that the fixed support assumptions of Chapters 7 and 8 of [3] do not hold here because of the varying window size. Another and more fundamental reason stems from the necessity to handle random graphs in a two dimensional infinite lattice in order to analyze the stationary regime of very large (here infinite) overlay networks, a case not covered in [3] either.

2.3 A Model for TCP Connections in Tandem

In this section we develop a model to study the performance metrics of reliable group communications using overlay. We shall first consider a special case of the overlay topology which is the chain topology. The general topology will be considered in the next section.

2.3.1 Assumptions and Notation

The overlay network consists of K nodes (end-systems), arranged in tandem, from 1 to K, as illustrated in Figure 2. The source is node 0. The source has an infinite number of packets to multicast. The m-th packet is available at time T_m. The sequence $\{T_m\}$ is (strictly) increasing or constant (e.g. $T_m = 0$ for all $m = 1, 2, \ldots$). We shall assume throughout the paper that the inter-arrival times $\{T_m - T_{m-1}\}_m$ are mixing. If a stationary sequence is mixing, it is ergodic; sequences of independent and identically distributed random variables are mixing (see e.g. [2] for more on these definitions). Under these conditions, the limit $\lambda \equiv \lim_{m\to\infty} m/T_m$ exists with probability 1 (w.p.1) thanks to the ergodic theorem. The parameter λ represents the packet arrival intensity at the source. When all the packets are available at time 0, namely $T_m \equiv 0$, we have $\lambda = \infty$. We shall refer to this case as the *saturated case*.

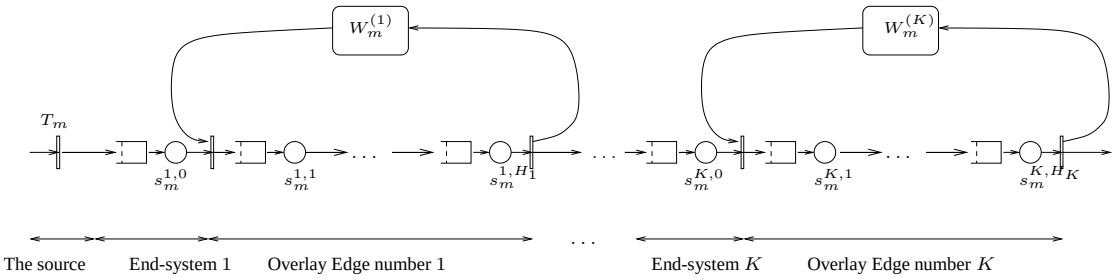


Figure 2: TCP connections in tandem

The (TCP) connection from node (end-system) k to node $k+1$ is referred to as overlay edge k. Underlying overlay edge k, there are H_k routers, denoted as routers (k, h) for $h = 1, \ldots, H_k$, which are modeled by single server queues. The TCP congestion control is characterized by the window size $W_m^{(k)}$, illustrated in Figure 2 as the arc from router $h = H_k$ to router $h = 1$ on this overlay edge k. The variable $W_m^{(k)}$ represents the window size when packet m is available at node k. This node is allowed to transmit packet m when packet $m - W_m^{(k)}$ is received by node $k+1$.

All these routers can have *cross traffic* (packets from other connections using the same router). The effect of such cross traffic is modeled by *aggregated service times* which represent the processing time of the packet of the reference TCP connection in the router plus the additional waiting time due to external packets intervening between two packets of the reference connection. Such a modeling approach was also used in [8]. We denote by $s_m^{(k,h)}$ the aggregated service time experienced by the packet number m going through the h-th router of the overlay edge k.

In addition to these routers, the node itself is also modeled as a single server queue. The corresponding service time, denoted by $s_m^{(k,0)}$ for packet m at node k, is the time to copy an incoming packet to an outgoing queue in the end-system. Typically, this time is negligible (at the order of ms

or less) compared to end-to-end round trip times. For simplicity of presentation, we shall assume that $s_m^{(k,0)} \equiv 0$.

2.3.2 Evolution Equations and Longest Paths in a Graph

We follow the TCP modeling approach proposed in [4]. For each TCP connection, we establish the evolution equations governing the packet arrival and departure times. The evolution of the TCP window size is governed by independent packet loss processes.

Let $x_m^{(k,h)}$ be the time when router (k,h) has finished forwarding packet m. Let $\vee$ denote max. Then for all k:

$$x_m^{(k,h)} = \left(x_{m-1}^{(k,h)} \vee x_m^{(k,h-1)}\right) + s_m^{(k,h)}, \qquad h > 1 \tag{1}$$

$$x_m^{(k,1)} = \left(x_{m-1}^{(k,1)} \vee x_m^{(k,0)} \vee x_{m-W_m^{(k)}}^{(k,H_k)}\right) + s_m^{(k,1)} \qquad h = 1 \tag{2}$$

$$x_m^{(k,0)} = \left(x_{m-1}^{(k,0)} \vee x_m^{(k-1,H_{k-1})}\right) + s_m^{(k,0)}, \qquad k > 1, h = 0 \tag{3}$$

$$x_m^{(1,0)} = \left(x_{m-1}^{(1,0)} \vee T_m\right) + s_m^{(1,0)}, \qquad k = 1, h = 0. \tag{4}$$

In words, these equations state that in order to serve packet m, packet $m-1$ should have departed from the same router; packet m should have arrived from the upstream router; and for the first router of the TCP connection (i.e. $h = 1$), the transmission should be allowed by the TCP congestion control.

The presence of losses will be taken into account via the evolution of the window size $(W_m^{(k)})_{m\in\mathbb{Z}}$, which will be governed by the AIMD rule of the congestion avoidance phase of RENO. The possible values of each window are $\{1, 2, \ldots, W_{\max}\}$. The window sequences will be assumed independent for different overlay edges. For each edge k, we consider a Markov chain made of the two variables $(W_m^{(k)}, r_m^{(k)})_{m\in\mathbb{Z}}$ with

$$(W_m^{(k)}, r_m^{(k)}) \in \left\{(w,r) \in \{1,2,\ldots,W_{\max}\}^2 \mid r \le w\right\},$$

where $W_{\max}$ is the maximum window size, $W_m^{(k)}$ is the current window size, and $r_m^{(k)}$ is the counter triggering window size increments. When the packet loss process is Markovian, the joint process of the loss and $(W_m^{(k)}, r_m^{(k)})$ is Markovian as well. In particular, when the losses are Bernoulli, $(W_m^{(k)}, r_m^{(k)})$ is a Markov chain with transitions given by:

From (w,r), with $r > 1$, the next state is:

$$\begin{cases} (w, r-1) & \text{with probability } 1 - p_k\,, \\ (\lfloor \frac{w}{2} \rfloor \vee 1, \lfloor \frac{w}{2} \rfloor \vee 1) & \text{with probability } p_k\,. \end{cases}$$

From $(w,1)$, the next state is

$$\begin{cases} ((w+1) \wedge W_{\max}, (w+1) \wedge W_{\max}) & \text{w. p. } 1-p_k \ , \\ (\lfloor \frac{w}{2} \rfloor \vee 1, \lfloor \frac{w}{2} \rfloor \vee 1) & \text{w. p. } p_k \ . \end{cases}$$

The parameter $0 < p_k < 1$ represents the packet loss probability along the routers of this edge of the overlay network. This Markov chain is irreducible and aperiodic over a finite state space, and thus converges to a steady state with coupling in finite time, and so does the joint process $(W_m^{(1)}, r_m^{(1)}, \ldots, W_m^{(K)}, r_m^{(K)})_{m \in \mathbb{Z}}$.

Note that other features of TCP such as timeout, retransmission, acknowledgment packet delays, etc., can also be taken into consideration in these equations in the way presented in [4].

As one can observe from Equations (1–4), only the maximum and plus operations are used. Thus, as in [4], the TCP connections in tandem can be represented as linear evolution equations in the max-plus algebra. These equations can be seen as a recursive way of computing the evolution of the packets in a large tandem (and in the same way in a large tree). They are the basis of the simulation tool used later in the paper.

Instead of using matrix algebraic calculations in this algebra, we adopt a more direct approach based on weighted random graph. The random graph describes the dependency relations between state variables $x_m^{(k,h)}$. It has

- the set of vertices $V = \{(m,k,h) | m \geq 1,\ 0 \leq k \leq K,\ 0 \leq h \leq H_k\}$; vertex (m,k,h) has weight $s_m^{(k,h)}$, where, by convention, we set $s_m^{(0,0)} = T_m - T_{m-1}$;
- the set of edges :

$$\begin{aligned} E = \quad & \{(m,k,h) \to (m-1,k,h) \mid m \geq 1\} \\ & \cup \{(m,k,h) \to (m,k,h-1) \mid m \geq 0,\ h \geq 1\} \\ & \cup \{(m,k,0) \to (m,k-1,H_{k-1}) \mid m \geq 0,\ k > 1\} \\ & \cup \{(m,1,0) \to (m,0,0) \mid m \geq 0\} \\ & \cup \{(m,k,1) \to (m - W_m^{(k)},k,H_k) \mid m \geq 0,\ k \geq 1\} \end{aligned}$$

  This graph represents the dependency structure when performing the recursive computation of the dates of events e.g. the computation of the variable with index (m,k,h) requires that of the variable $(m-1,k,h)$ etc.

Part of this graph is illustrated in Figure 3, where three types of edges are presented: horizontal edges (from packet m to $m-1$, for the same station); vertical edges (from station k to station $k-1$, for the same packet) and edges representing the window congestion control (that go backward of W_m packets, and from the first hop $(k,1)$ to the last hop (k,H_k) of a connection).

For a path π defined by a sequence of vertices in this graph, we denote by $\texttt{Wei}(\pi)$ the sum of weights of vertices of the path. It is then easy to check the following property using induction and the evolution equations (1–4).

$$\begin{gathered} x_m^{(k,h)} \text{ is given by the maximum of } \texttt{Wei}(\pi) \\ \text{over all possible paths } \pi \text{ from } (m,k,h) \text{ to } (1,0,0) \end{gathered}$$

Notation : For any value of m, k, h, m', k' and h', let :

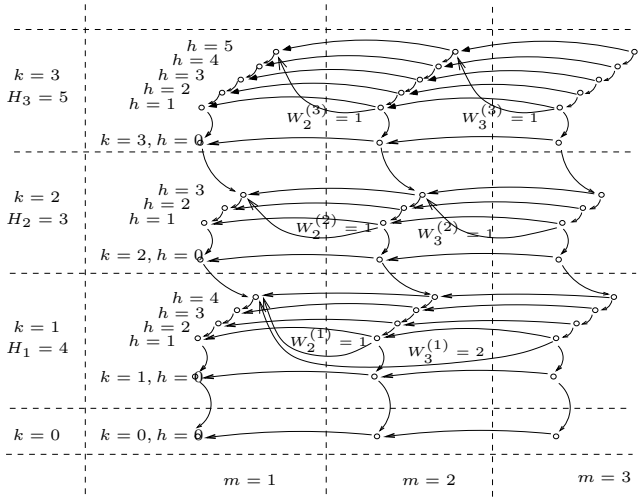


Figure 3: Random Graph to represent a tandem of TCP connections

- $P_{(m,k,h)\to(m',k',h')}$ denote the set of the paths from vertex (m,k,h) to vertex (m',k',h');
- $\texttt{Wei}_{((m,k,h)\to(m',k',h')} = \max_{\pi\in P_{(m,k,h)\to(m',k',h')}} \texttt{Wei}(\pi)$;
- $\texttt{Wei}_{(m,k)\to(m',k')} = \texttt{Wei}_{(m,k,0)\to(m',k',0)}$.

Note that, as $s_m^{(k,0)}$ is assumed to be zero for any k and m,

$$\texttt{Wei}_{(m,k,H_k)\to(m',k',0)} = \texttt{Wei}_{(m,k+1)\to(m',k')} \ .$$

3 Scalability of Throughput

In this section, we consider the throughput of the group. We first consider the case of the chain topology of the overlay network. We shall then investigate the case of arbitrary tree topology. Some experimental results will be presented at the end of this section.

3.1 Scalability of the Chain Topology

We first consider the case of the chain topology of the overlay network. We thus use the queueing network model analyzed in the previous section. We show that the reliable group communication with such an overlay exhibits a throughput equal to the minimum of the local saturation TCP throughputs of all overlay edges, where by local saturation TCP throughput, we mean the throughput achieved

by the TCP session when the sender has an infinite backlog of packets to transmit so that it is not constrained by its upstream node.

Let $D^{\lambda}_{m,k} = x_m^{(k,H_k)}$ be the time when the m-th packet has been transmitted in the k-th overlay edge. This quantity will be denoted by $D^{\infty}_{m,k}$ in the saturated case $\lambda = \infty$. The throughput of the group communication is defined as $\Theta^{\lambda}_{1,K} \equiv \lim_{m\to\infty} \frac{m}{D^{\lambda}_{m,K}}$, provided the limit exists, where we recall that λ denotes the arrival intensity of packets from the source. When this limit exists, it represents the long term average of the throughput seen at the output of overlay network K. When $\lambda = \infty$, we shall denote it by $\Theta^{\infty}_{1,K}$. The next theorem shows that under mild assumptions, the throughput limit exists.

Assumption 1 *The sequences of aggregated service times are independent for different routers and independent of the inter-arrival times as well as of the loss processes. For all* $1 \le k \le K$, $1 \le h \le H_k$*, the sequence* $\{s_m^{(k,h)}\}_m$*, is mixing.*

Note that this assumption is very general and allows in particular the aggregated service times to be long range dependent. It also contains as a special case the case where these aggregated service times are independent and identically distributed, as assumed in [31].

Theorem 1 *Under assumption 1, for all* $1 \le k \le K$, $\dfrac{m}{D^{\lambda}_{m,k}}$ *converges almost surely to a constant* $\Theta^{\lambda}_{1,k}$ *as* m *tends to infinity.*

Proof: We start by showing the following subadditivity property: for any $k \ge 1$ we define, for $a \ge 1$ and $0 \le b$:

$$\xi^{(k,H_k)}_{a,a+b} = \max\left\{\texttt{Wei}(\pi) \mid \pi : (a+b,k,H_k) \to \ldots \to (a,0,0)\right\}.$$

Then we have for any $k \ge 1$, $a \ge 1$ and $0 \le b \le c$:

$$\xi^{(k,H_k)}_{a,a+c} \le \xi^{(k,H_k)}_{a,a+b} + \xi^{(k,H_k)}_{a+b,a+c}. \tag{5}$$

To prove (5), consider a path π in the random graph from $(a+c,k,H_k)$ to $(a,0,0)$, and denote the sequence of the first coordinates of the vertices along path π by: $a+c = m_0, m_1, \ldots, m_L = a$.

We can divide this path into a first set of steps made in the domain $a+c \ge m \ge a+b+1$ (which creates path π_1) and the rest of the steps made in the domain $a+b \ge m \ge a$ (creating path π_2). We can then complete the first path π_1 by an appropriate suffix in order to create a path $\bar{\pi}_1$ from $(a+c,k,H_k)$ to $(a+b,0,0)$. The second path π_2 can also be completed by a prefix in order to form a path $\bar{\pi}_2$ from $(a+b,k,H_k)$ to $(a,0,0)$. Hence

$$\begin{aligned}\texttt{Wei}(\pi) &= \texttt{Wei}(\pi_1) + \texttt{Wei}(\pi_2) \le \texttt{Wei}(\bar{\pi}_1) + \texttt{Wei}(\bar{\pi}_2)\\ &\le \xi^{(k,H_k)}_{a+b,a+c} + \xi^{(k,H_k)}_{a,a+b}.\end{aligned}$$

Taking the maximum over all possible paths π proves (5).

As $x_m^{(k,H_k)} = \xi_{1,m}^{(k,H_k)}$, the result to be proved also reads: $\frac{1}{m}\xi_{1,m}^{(k,H_k)}$ converges a.s. to a finite deterministic constant, for $m \to \infty$.

We have seen in Section 2.3.2 that the sequence of successive window values

$$\{W_m^{(1)} \dots, W_m^{(K)}\}_{m \geq 1}$$

couples with a steady state for an almost surely finite packet number $M_{(\texttt{cpl})}^{(W)}$.

As all these stationary sequences (stationary windows values, aggregated service times in routers, and inter-emission of packets at the source) are supposed independent. The first one (the stationary window values) forms a mixing sequence because it is the product of aperiodic, irreducible and positive recurrent Markov chains (such a chain is mixing as easily deduced from Kolmogorov's asymptotic theorem for Markov chains, and the product of independent mixing processes is mixing); the other ones are mixing too by assumption. Thus, jointly, these sequences are mixing too and hence ergodic.

For the system taken in steady state (that we denote by $\tilde{}$), $(\tilde{\xi}_{a,a+b}^{K,H_K})_{a \leq a+b}$ is an ergodic process, and it is subadditive as a consequence from the first part of the proof. Kingman's subadditive ergodic theorem then states that $\lim_{b\to\infty} \frac{1}{b}\tilde{\xi}_{a,a+b}^{K,H_K}$ a.s. converge to a constant. The result is then proven once we have observed that the sequence $(\xi_{1,m}^{K,H_K})_{m\geq 1}$ couples a.s. in finite time with its stationary version. □

Of particular interest is the local saturation throughput of TCP connections at the overlay edges. Let θ_k be the local saturation throughput of the overlay edge k which is defined as the throughput obtained when it is directly fed by a source with an infinite backlog. In other words, θ_k is the value of $\Theta_{1,k}^{\infty}$ when the overlay edges $1, \cdots, k-1$ all have zero aggregate service times on their routers. In the next section we shall establish a relation between these local saturation TCP throughput and the reliable group communication throughput.

Lemma 1 *Under Assumption 1, for all* $1 \leq k \leq K$, $\Theta_{1,k}^{\lambda} = \min(\Theta_{1,k-1}^{\lambda}, \theta_k)$*, i.e., the throughput of the first* k *nodes is the minimum of the throughput of the first* $k-1$ *nodes and the local saturation throughput of overlay edge* k*, where by convention* $\Theta_{1,0}^{\lambda} = \lambda$*.*

Proof: For $k = 1$, the assertion that we need to prove is $\Theta_{1,1}^{\lambda} = \min(\lambda, \theta_1)$. Clearly, if $\lambda \geq \theta_1$, then the queueing station $(1, 0)$ will eventually be saturated so that the TCP connection overlay edge 1 will behave as if it was directly fed with a source with an infinite backlog. Therefore, $\Theta_{1,1}^{\lambda} = \theta_1 = \min(\lambda, \theta_1)$.

If $\lambda < \theta_1$, then the queueing network composed of the source and the routers of the overlay edge 1 will be stable and, thanks to the convergence in variation of the Markov chain of the TCP window size $(W_m^{(1)}, r_m^{(1)})$, the output point process will converge with coupling to a stationary and ergodic point process (see e.g. [4]). Therefore, the throughput of this overlay will be λ too. This completes the proof of the case $k = 1$.

Assume the assertion holds for some $k \geq 1$. Consider the case of $k+1$. Then, the overlay network composed of the source and the overlay nodes $1, \cdots, k$ will act as the source for overlay edge $k+1$. The same argument as in the induction base can be used to show that $\Theta_{1,k+1}^{\lambda} = \min(\Theta_{1,k}^{\lambda}, \theta_{k+1})$. □

As a direct corollary of Lemma 1, we have

Theorem 2 *Under Assumption 1, for all* $1 \leq k \leq K$, $\Theta^{\lambda}_{1,k} = \min(\lambda, \theta_1, \cdots, \theta_k)$*, i.e., the throughput of the first* k *nodes is the minimum of the arrival intensity and of the local saturation throughput of the overlay edges* $1, \cdots, k$*. In particular, if* $\lambda = \infty$*, we have* $\Theta_{1,k} = \min(\theta_1, \cdots, \theta_k)$.

Therefore, when the local saturation TCP throughputs are strictly positive, the reliable group communication using overlay network is scalable in the sense that its throughput is lower bounded by the minimum of the local saturation throughputs. In particular, if all saturation throughputs are bounded away from 0, then even an infinite diffusion tandem leads to a positive group communication throughput.

3.2 Tree Topologies with Uncongested Access Links

We now consider arbitrary tree topologies, still under the assumption that the overlay buffers are unbounded. With general tree topology with K overlay edges, the throughput of the group communication is still defined as the minimum throughput observed at the output of end-systems: $\Theta^{\lambda}_{1,K} \equiv \lim_{m\to\infty} \min_{1\leq k\leq K} \frac{m}{D^{\lambda}_{m,k}} = \min_{1\leq k\leq K} \lim_{m\to\infty} \frac{m}{D^{\lambda}_{m,k}}$.

Suppose Assumptions 1 and 2 hold with:

Assumption 2 *The aggregated service times in any router of an overlay edge originating from a node are independent of the number of TCP connections originating from this node.*

A few comments on Assumption 2 are necessary. This assumption pertains to the access links of the end-systems. It assumes that none of these links is actually congested due to the presence of the multiple TCP connections originating simultaneously from the end-system nodes, which might not be true if the out degree of nodes is too large in the overlay tree (in the core of the Internet, there are simultaneously a big number of other TCP sessions anyway. So each individual session added by the multicast tree has little effect on the router behavior).

By the same arguments as above applied to all paths of the tree, we obtain:

Theorem 3 *Under Assumptions 1 and 2, for any arbitrary tree rooted at the source node,* $\Theta^{\lambda}_{1,K} = \min(\lambda, \theta_1, \cdots, \theta_K)$*, and in particular, we have* $\Theta^{\infty}_{1,K} = \min(\theta_1, \cdots, \theta_K)$*, with* θ_k *the local saturation throughput of overlay edge* k.

Thus, in contrast to the results of [8] which established that in the presence of random perturbations, the throughput of IP supported multicast goes to 0 when the size of the group goes to infinity, we have a scalability result for the throughput of overlay multicast trees under Assumption 2.

The main reason for the different fate of throughput in IP supported reliable multicast as considered in [8] and in overlay multicast can be intuitively explained as follows. The end-to-end control of IP supported reliable multicast makes it such that each node will be permanently *randomly* delayed due to its waiting for the acks of the the latest of its offspring nodes, whereas in overlay multicast, each line of offspring of a node can actually progress at its own and proper speed and a key decoupling takes place which allows each TCP connection to get the long term average throughput it would get in the absence of the other parts of the tree.

3.3 Tree Topology with Possibly Congested Access Links

Assumption 2 allowed the reduction of overlay trees to overlay tandems by assuming that the transfers of the multicast tree not belonging to some *reference path* had no impact on the throughput of the various overlay edges along this path.

However, if the out degree of some node of the tree is large, then the access link from this node may become the actual bottleneck due to the large number of simultaneous transfers originating from this node. Hence the throughput of the transfer of the reference overlay edge originating from this node may in fact very well be significantly affected by the other transfers originating from this node.

This "first-mile link" effect can be incorporated in our model. The extra traffic created by the transfers not located on the reference path can be represented by an increase of the aggregated service times on the reference path (we remind that aggregated service times represent the effect of cross traffic on the reference TCP transfer – see e.g. [8]).

We now show that whenever the out degree of each node is bounded from above by some constant integer M (2 in the case of a binary tree), then the main scalability results of the last subsections are still valid (though with different constants) provided some natural assumptions listed below are satisfied. In what follows, we denote by $f(k)$ the index of the end-system that is the source of overlay edge k.

Assumption 3 Locality assumption*: in the reference path, the non-reference transfers originating from end-system $f(k)$ affect the aggregated service times of the reference transfer of overlay edge k only.*

This assumption is quite natural should the nodes of a given multicast application be sparse enough for being all located on different LANs or geographical areas.

Assumption 4 Fairness assumption*: let $s_m^{(k,h)}$ (resp. $\overline{s}_m^{(k,h)}$) denote the aggregated service time of packet m of the reference transfer on hop h of overlay edge k when the out degree of end-system k is equal to 1 (resp. M). The fairness assumption states that $\overline{s}_m^{(k,h)} \leq M s_m^{(k,h)}$.*

The terminology of the last assumption stems from the fact that if for all m and h, $\overline{s}_m^{(k,h)} = M s_m^{(k,h)}$, then the average throughput of the reference connection is exactly divided by M when moving from 1 to M transfers stemming from node $f(k)$, which is the usual fairness assumption made on TCP bandwidth sharing in the presence of multiple transfers with the same RTTs.

Notice that the situation where $\overline{s}_m^{(k,h)} = M s_m^{(k,h)}$ for all m and h corresponds to a worst case scenario since increasing the number of simultaneous transfers from 1 to M in end-system $f(k)$

- should probably only affect the aggregated service times of the very first hops of overlay edge k rather than all;
- could only lead to a multiplication of the aggregated service times of overlay edge k by M in case there is no other type of cross traffic than these other simultaneous transfers.

So even in this worst case scenario, under Assumptions 3 and 4, the throughput obtained by each reference transfer is at most divided by M when taking into account the effect of *all* other branches

of the tree. So under these two assumptions, the conclusions of the other sections are still valid with a worst case scenario obtained by dividing all earlier throughputs by M.

Above, we assumed that the RTTs of all the TCP connections originating from a node to its downstream nodes were approximately the same. In case of heterogeneous RTTs, if one assumes a bandwidth sharing inversely proportional to RTT (one of the cases considered in e.g. [19]), it is then easy to get a similar result via bounding techniques, though with different constants, at least whenever all RTTs are bounded.

3.4 Experimental Results

In Tables 1, 2 and 3 we present results of our measurements of throughput and buffer utilization. In each table, the leftmost column contains the symbolic names assigned to hosts used in the experiments. The indentation in this column describes the structure of the overlay multicast tree, with the first indentation level corresponding to the root of the tree, the second to its children etc. For each non-root node, we list the characteristics of the incoming link to that node (so that each line actually describes a link). In each configuration we repeated measurements 10 times, and took average, minimum and maximum of measured parameters.

In each table, the second column shows the local saturation throughput of the incoming link in kilobytes per second, measured shortly after or before the multicast diffusion. This local saturation throughput is that obtained when the local node always has packets to transmit downstream. On each local node, all parallel transfers were started simultaneously so as to take into account the bandwidth sharing on last-mile links as described in §3.3. In these experiments, buffer size was not restricted.

The last two columns show effective throughput and buffer utilization measurements, as observed during the global overlay multicast. By effective throughput, we mean the throughput observed on overlay edge during the mulicasting. We report the maximum number of entries used in the buffer located on the upstream node of the link. Each buffer entry corresponds to one 100-byte block. In Tables 1 and 3 we send 20,000 blocks, and in Table 2 we only send 5,000 blocks. Buffer utilization is measured as a proportion of the maximum number of blocks used in the buffer to the total number of blocks sent during experiment. Notice that buffer utilization at the root node is high, since data is generated at the root node very quickly, and almost all blocks are immediately buffered. When following a path in the tree, one observes the diminishing effective throughput phenomenon studied in Lemma 1.

One immediately observes from these tables that for all these three configurations, the assertion of Theorem 3 is valid, namely, the group throughput is equal to the minimum of the local saturation TCP throughput. One can also observe that the buffer occupancy is quite high in many buffers. This is due to the fact that the local saturation TCP throughputs are quite heterogeneous.

4 Scalable Buffers via Rate Control

This section explores the scalability of buffer occupancy and of latency. Namely, we examine the conditions under which the multicast communication can achieve finite local latency and finite buffer occupancy after the transmission of a large number of packets on an overlay tree of arbitrary size.

Node	Link Throughput (KB/s)			Effective Throughput (KB/s)			Buffer Utilization (%)		
	min	avg	max	min	avg	max	min	avg	max
b7									
asterix-1	201	235	254	147	155	165	98	98	99
ace	356	372	403	147	155	165	0	0	1
edge	231	235	244	147	155	164	0	0	1
asterix-2	186	204	224	146	154	164	3	4	5
ananda-1	341	397	507	147	155	165	0	0	0
umn-1	864	885	900	147	155	164	0	0	1
baobab	103	113	124	113	116	119	31	36	44
fermi-1	31	32	34	22	36	58	60	69	74
berk-1	121	209	309	22	36	58	1	1	1
pisa-1	21	25	28	17	19	21	82	83	83
ucsb-1	721	769	821	17	19	21	1	1	1
cmu-1	667	671	678	17	19	21	1	1	1
berk-2	107	387	555	219	367	558	95	96	99
ucsb-2	65	118	173	135	159	177	27	46	66
cmu-4	538	625	673	134	158	176	0	1	1
ananda-2	1044	1159	1366	134	158	176	0	0	0
dogmatix	219	372	561	134	150	164	0	10	27
umn-2	872	877	888	134	150	164	0	0	0
b8	91	133	165	128	154	186	49	59	69
asterix-3	258	276	308	128	136	146	10	17	27
berk-3	94	161	214	116	125	133	3	4	4
pisa-2	346	483	560	127	135	146	3	3	3
cmu-2	884	905	939	128	154	185	0	1	1
fermi-2	660	690	721	128	154	185	0	0	0

Table 1: Configuration 1.

Node	Link Throughput (KB/s)			Effective Throughput (KB/s)			Buffer Utilization (%)		
	min	avg	max	min	avg	max	min	avg	max
ananda-1									
ucsb-1	87	89	90	81	87	89	95	99	100
umn-1	77	232	452	56	82	88	1	4	31
berk-2	25	34	44	27	35	41	32	50	60
asterix-2	37	55	96	43	61	88	2	25	42
edge-2	104	120	163	43	60	87	1	4	29
cmu-2	195	378	549	80	85	88	1	1	1
dogmatix	581	988	1575	80	85	87	0	1	1
umn-2	120	331	461	69	83	87	0	2	19
fermi-2	143	164	219	80	85	88	1	1	2
asterix-3	256	295	326	80	85	88	1	1	2
ananda-2	341	471	618	80	85	87	1	1	2
edge	126	162	189	162	204	249	87	96	100
baobab	2	13	25	1	10	25	61	79	89
pisa-1	10	24	57	1	9	17	10	20	49
edge-3	99	121	200	1	9	17	2	2	4
cmu-1	14	28	47	1	10	19	12	20	49
ace	581	642	729	1	10	19	1	1	1
berk-1	63	96	130	104	144	216	4	27	73
pisa-2	224	467	555	94	141	214	1	6	31
fermi-1	219	253	271	92	138	209	1	3	6
cmu-3	498	533	549	93	139	213	1	3	6
asterix-1	26	51	84	17	47	202	6	61	90
ucsb-2	83	85	88	17	35	83	2	13	66
berk-3	109	177	263	17	42	147	1	8	51

Table 2: Configuration 2.

Node	Link Throughput (KB/s)			Effective Throughput (KB/s)			Buffer Utilization (%)		
	min	avg	max	min	avg	max	min	avg	max
ace									
berk-3	112	296	472	189	269	342	95	96	99
dogmatix	72	140	206	71	134	176	26	54	76
fermi-1	500	554	599	71	133	176	2	2	2
edge-2	115	129	150	71	116	147	5	16	31
asterix-2	376	392	409	71	133	175	0	1	1
cmu-3	1843	1940	1993	71	133	176	2	2	2
berk-2	92	248	457	71	127	174	0	7	24
umn-2	88	171	311	60	154	246	29	48	68
geranium	19	20	22	32	36	42	42	74	87
fermi-2	59	67	72	32	36	42	0	3	5
asterix-3	26	34	44	30	33	35	9	17	34
pisa-2	30	31	33	47	56	69	25	55	76
ananda-2	329	443	571	47	56	69	1	1	1
cmu-2	760	868	948	944	966	1002	95	95	99
baobab	122	124	124	96	118	124	88	91	98
cmu-1	39	46	56	30	49	74	38	63	79
umn-1	460	622	797	30	49	74	1	1	1
ananda-1	1039	1357	1860	30	49	74	1	1	1
ucsb-1	19	25	29	19	25	30	80	85	91
berk-1	141	701	1263	19	25	30	0	1	2
pisa-1	585	657	723	348	614	723	29	43	81
edge-3	100	105	109	104	142	181	55	74	81
ucsb-2	69	71	72	71	73	74	30	47	58
asterix-1	90	107	132	62	88	135	80	86	91
edge	119	136	159	62	85	114	0	10	46

Table 3: Configuration 3.

We will use the sojourn time of a packet m in the k-th overlay edge: $V^{\lambda}_{m,k} = D^{\lambda}_{m,k} - D^{\lambda}_{m,k-1}$.

4.1 The Necessity of a Rate Control at the Source

The first result of this section shows the necessity to control the rate at which the source sends data.

Theorem 4 *If the intensity of packet arrival date $(T_m)_{m\in\mathbb{Z}}$, denoted by λ is larger than $\Theta^{\infty}_{1,K}$ (defined in §3.1), then there exists at least one station $1 \leq k \leq K$, for which the sojourn time of packet m converges to infinity a.s.*

Proof: The result reduces to the study of an overlay edge with reference saturation throughput $\theta^{\infty}_k \leq \lambda$. We have to distinguish between the case $\lambda > \theta^{\infty}_k$. The result follows from the ergodic theorem. The techniques are similar to those of Chapters 7 of [3]. □

4.2 Scalability of Latency with Rate Control

We now consider the case where the source throttles packet emission at rate λ with $\lambda < \Theta^{\infty}_{1,K}$ and in such a way that the inter-emission times at the source, which will be denoted by $\tau_m = T_{m+1} - T_m$, form a stationary and mixing sequence. Then under the assumptions of the last sections concerning windows and service times, one can construct the stationary regime of the first K overlays as follows.

Let $R_{m,k}$ denote the time that elapses between the emission of packet m by the source until this packet leaves overlay k, namely $R_{m,k} = D^{\lambda}_{m,k} - T_m$, where $D^{\lambda}_{m,k}$ is the departure time of packet m from overlay edge k, which is obtained from the max-plus equations (1–4) for the boundary conditions T_m at the source described above.

The stationary version of the $R_{m,k}$ variable can be viewed as that obtained when taking into account the emission of all packets $n \leq m$, where n ranges down to $-\infty$, and the stationary version of the window process (which can be continued to all positive and negative indices using classical time reversal arguments). It is easy to check that this variable, which will be denoted by $\tilde{R}_{m,k}$, can also be represented using our longest path approach via the formula

$$\tilde{R}_{m,k} = \sup_{n \leq m} \{\mathtt{Wei}_{(m,k+1)\to(n,1)} - \sum_{i=n}^{m-1} \tau_i\}. \tag{6}$$

For more on this type of representations, see Appendix III and [1]. This has to be compared to the transient version of the $R^k_{[m',m]}$ variable when taking only into account all packets between m' and m (with of course $m' \leq m$) and when departing from an empty system which is given by: $R^k_{[m',m]} = \sup_{m' \leq n \leq m}\{\mathtt{Wei}_{(m,k+1)\to(n,1)} - \sum_{i=n}^{m-1} \tau_i\}$. It is clear from this representations that $R^k_{[m',m]} \leq \widetilde{R}_{m,k}$ for all m' (within this setting, the stationary regime is the worst case scenario when compared to the transient starting from an empty system). Of course, this stationary regime allows one to define that of the sojourn time of packet m in overlay k via the formula $\widetilde{V}_{m,k} = \widetilde{R}_{m,k} - \widetilde{R}_{m,k-1}$ with the convention $\widetilde{R}_{m,0} = 0$.

In what follows, both in the simulation results and the mathematical derivations, the stochastic assumptions will be that the packet inter-emission times at the source are i.i.d. and independent of the aggregated service times. The aggregated service time will also be assumed independent for different routers and i.i.d. for each given router.

We are considering both

- The homogeneous case where all overlay edges have the same number of routers and the same aggregated service time law; we will denote by θ the local saturation throughput of an overlay edge.
- The non homogeneous case where we will assume that the laws of the aggregated service times to be bounded from above by a variable $\bar{s}$, with respect to the stochastic order (see e.g. [2], Ch 4): for all k, h and m, $s_m^{(k,h)} \leq_{\mathtt{st}} \bar{s}$. In addition, we assume that the number of routers in an overlay edge H_k are all bounded by some constant $\bar{H}$. In this case, we will consider the *homogeneous upper bound* system where $H_k = \bar{H}$ for any value of k and the aggregated service times in nodes are independent and distributed like $\bar{s}$ (except for $k \geq 1$ and $h = 0$ where $s_m^{(k,0)} = 0$). Here θ denotes the local saturation throughput of such an homogeneous upper bound overlay network.

Under the condition $\lambda < \theta$, the throttling mechanism is hence such that all finite trees admit a stationary regime in the sense that the stationary sequence $\{\tilde{R}_{m,k}\}$ is finite (see Appendix III). Hence, under this condition, for all multicast trees of depth K, the buffer occupancy of any end-system and the packet sojourn time through any overlay edge converge in distribution to finite random variables when the number of transmitted packet goes to infinity.

The main scalability question concerns what happens when one then lets K go to infinity. Do the stationary sojourn time through an overlay edge of depth K and the buffer occupancy in an end-system of depth K converge to a finite limit when K goes to infinity ?

The mathematics for approaching these questions of buffer occupancy and packet latency in very large networks require the extension of the hydrodynamic limits proved in [1] for infinite tandems of GI/GI/1 queues to infinite tandems and infinite trees of TCP connections over edges composed themselves of several routers. We will start with simulation results and back them by mathematical justifications.

4.3 Simulation Results

All the simulation results of the paper are based on a direct exploitation of the evolution equations (1)-(4) of §2.3.2. Only the homogeneous case is considered.

Figure 4 studies the stationary mean buffer occupancy in an end-system located at level k of an overlay network composed of an arbitrary tree. The throttling of the source is assumed to be realized via a deterministic scheme: it sends a packet every λ^{-1} seconds with $\lambda < \theta$.

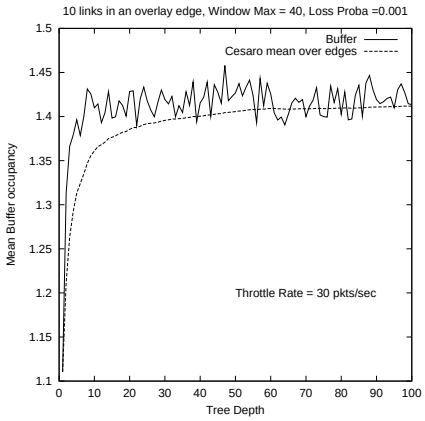


Figure 4: Convergence of the mean buffer occupancy at infinity.

As one can check, the mean stationary buffer occupancy grows with k and stabilizes to some asymptotic value $\overline{b}$, which can be intuitively thought of as the mean stationary buffer of an end-system being at level ∞. This convergence illustrates the key scalability result alluded to above. Combined with Little's law, this extends to a similar limiting result for the "delay at infinity", $\overline{d}$ which is again defined as the limit in k of the stationary mean delay through an overlay edge located at level k, when k goes to infinity.

Figure 5 (left) studies the sensitivity of the $\overline{b}$ function w.r.t. the throttling rate λ of the source. Four different curves are plotted that give $\overline{b}$ as a function of λ for all $\lambda < \theta$. The only difference between these four curves is the distribution function of the aggregated services representing the influence of cross traffic. The lowest curve is that with exponential aggregated service times. The upper curves feature various Pareto distributions with increasing variability.

As one can check the mean buffer occupancy at infinity is quite sensitive to the variability of cross traffic. The influence of an increased variability of the aggregated service distribution functions is

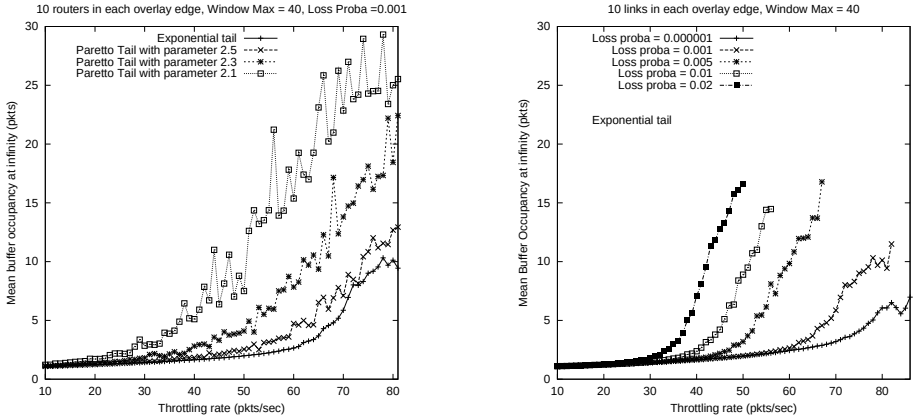


Figure 5: The mean buffer occupancy at infinity as a function of the throttling rate : for different laws of service time (left), and for various packet loss probabilities (right).

well illustrated by the comparison of the exponential case to any of the Pareto cases and also by the comparison of the various Pareto cases.

Figure 5 (right) studies the sensitivity of the $\overline{b}$ function w.r.t. the packet loss probability.

4.4 Mathematical Comments

The general aim of this section consists in outlining the main steps of a mathematical justification of the scalability results observed by simulation in the last subsection. The line of thoughts is in the continuation of that of [1], [22] and [23]. And the detailed proof of the different results can be found in Appendix.

The random variable $\bar{s}$ is assumed to satisfy the condition:

$$\int_0^{+\infty} P(\bar{s} \geq u)^{1/2} du < \infty \ . \tag{7}$$

As shown in [22], this condition is implied by moment conditions (for instance, if s has its moment of order 3 finite, then (7) holds; refined conditions that are weaker than the finiteness of moment of order 3 but stronger than the finiteness of moment of order 2 can also be found in this reference). This condition is satisfied in all simulated examples of this paper.

Theorem 5 *Under this assumption, for all $x \geq 0$, the a.s. limit $\gamma(x) = \lim \frac{D^{\infty}(\lfloor xk \rfloor, k)}{k}$ exists and is finite for all rational numbers x. The γ function is deterministic and nondecreasing.*

This function is called the hydrodynamic limit of the saturated system. It describes the asymptotic growth rate of the maximal weight path in direction x. The proof of this result is based on subadditivity and on the notion of *greedy lattice animal* (see the references in [22]). All details can be found in appendix II.

We are now in a position to state the main mathematical result backing the scalability of latency.

Theorem 6 *Under the last set of assumptions, if the γ function is concave, and $\lim_{x\to\infty}\gamma'(x) = \theta^{-1}$ then*

$$\frac{1}{K}\sum_{k=1,..,K}\widetilde{V}_{m,k} = \frac{1}{K}\widetilde{R}_{m,K} \to \overline{d} < \infty \text{ as } K \to \infty,$$

where the last convergence takes place both a.s. and in expectation. In addition, $\overline{d}$ is given by the Legendre transform of γ at point $\frac{1}{\lambda}$:

$$\overline{d} = \sup_{x>0}\left(\gamma(x) - \frac{1}{\lambda}x\right). \tag{8}$$

The proof is similar to those used for analogue results in [1], [22] and [23], and can be found in appendix III. This result should be interpreted as follows: when the depth of the overlay tree grows large, the sum of the delays on a path originating from the source and ending in some end-system (or equivalently the overall latency up to this end-system), grows linearly with the level of the end-system, with an average increment of $\overline{d}$ seconds per overlay in the limit, where $\overline{d}$ is some finite constant. The computation of the constant $\overline{d}$ requires the knowledge of the hydrodynamic limit $\gamma(x)$ associated with the random graph of the saturated problem. To the best of our knowledge, the explicit form of this function is only known in the particular case with constant window $W_m^{(k)} \equiv 1$, with $H_k = 1$, and with $\overline{s}$ exponential, where it was studied in the context of first passage percolation (see [1] and the references therein). Fortunately, the exact value of $\overline{d}$ is not needed in order to derive the qualitative scaling result of the last theorem, namely the finiteness of $\overline{d}$.

Figure 6 gives an example of the γ function. For this case as for all other simulated cases, the conditions allowing one to compute $\overline{d}$ from γ and in particular its concavity are clearly satisfied (up to the statistical noise).

Figure 7 plots two evaluations of $\overline{d}$ as a function of the throttling rate λ : The first one gives $\overline{d}$ defined analytically via (8) whereas the second one evaluates $\overline{d}$ by simulation as the average stationary sojourn time at infinity. Note that these two evaluations of $\overline{d}$ are based on very different mathematical objects: on one side the Legendre transform of the hydrodynamic function of the saturated system, and on the other side something equivalent to a mere discrete event simulation of the rate throttled system. The match is nevertheless very good, as long as the throttling rate is not taken too close to θ. A similar match was observed in all simulations.

The results of Theorem 6 extend to buffer contents. We have :

$$\lim_{k\to\infty}\tfrac{1}{k}\textstyle\sum_{l=1}^{k} E[\widetilde{V}_{m,l}] = \overline{d}. \tag{9}$$

Let $\widetilde{B}_{m,k}$ denote the stationary buffer occupancy in overlay k, which by definition includes the packets buffered in the k-th node itself and those in transit in the path from node k to node $k+1$. From Little's law, for all k, $E[\widetilde{B}_{m,k}] = \lambda E[\widetilde{V}_{m,k}]$, where λ denotes the rate of the stationary input into overlay k. So, from (9), the limit $\lim_{k\to\infty}\frac{1}{k}\sum_{l=1}^{k} E[\widetilde{B}_{m,l}]$ exists and is equal to a finite constant (equal to $\overline{d}\lambda$). This shows that under the throttling strategy described in §V.C, the buffer occupancy scales in the following sense: when the number of overlays grows large, the sum of

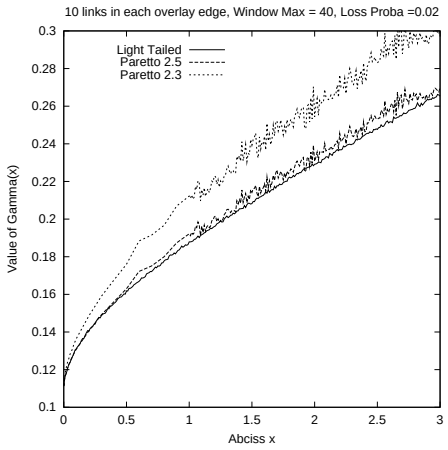


Figure 6: An example of hydrodynamic function for the saturated system.

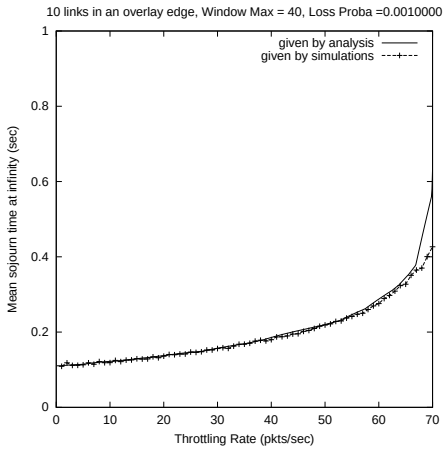


Figure 7: Mean sojourn time at infinity by two different methods.

the mean stationary buffer contents grows linearly with the number of overlays, with an average increment of $\overline{d}\lambda$ packets per overlay in the limit.

4.5 Experimental Results

In Tables 4, 5 and 6 we show effect of transmission rate control at the source node on buffer utilization. This experiment is identical to that described in § 3, except that we have introduced a 10-millisecond delay between sending individual 100-byte blocks at the source node.

This experiment is performed on the same configurations, as in Tables 1, 2 and 3. Numbers for link throughput are repeated from these tables. To collect the measurements, we ran unsyn-

Node	Link Throughput (KB/s)			Fixed Rate (KB/s)			Buffer Utilization (%)		
	min	avg	max	min	avg	max	min	avg	max
b7									
asterix-1	201	235	254	10	10	10	0	1	3
ace	356	372	403	10	10	10	0	0	0
edge	231	235	244	10	10	10	0	25	74
asterix-2	186	204	224	10	10	10	0	0	0
ananda-1	341	397	507	10	10	10	0	0	0
umn-1	864	885	900	10	10	10	0	0	0
baobab	103	113	124	10	10	10	0	1	4
fermi-1	31	32	34	10	10	10	0	1	3
berk-1	121	209	309	10	10	10	1	1	1
pisa-1	21	25	28	10	10	10	1	2	3
ucsb-1	721	769	821	10	10	10	1	1	1
cmu-1	667	671	678	10	10	10	1	1	1
berk-2	107	387	555	10	10	10	0	0	0
ucsb-2	65	118	173	10	10	10	0	0	0
cmu-4	538	625	673	10	10	10	0	0	0
ananda-2	1044	1159	1366	10	10	10	0	0	0
dogmatix	219	372	561	10	10	10	0	0	0
umn-2	872	877	888	10	10	10	0	0	0
b8	91	133	165	10	10	10	0	2	6
asterix-3	258	276	308	10	10	10	0	1	3
berk-3	94	161	214	10	10	10	0	0	1
pisa-2	346	483	560	10	10	10	0	1	1
cmu-2	884	905	939	10	10	10	0	0	1
fermi-2	660	690	721	10	10	10	0	0	0

Table 4: Configuration 1, fixed rate.

Node	Link Throughput (KB/s)			Fixed Rate (KB/s)			Buffer Utilization (%)		
	min	avg	max	min	avg	max	min	avg	max
ananda-1									
ucsb-1	87	89	90	5	5	5	0	0	0
umn-1	77	232	452	5	5	5	0	0	0
berk-2	25	34	44	5	5	5	0	1	8
asterix-2	37	55	96	5	5	5	0	0	0
edge-2	104	120	163	5	5	5	0	0	0
cmu-2	195	378	549	5	5	5	0	0	0
dogmatix	581	988	1575	5	5	5	0	0	0
umn-2	120	331	461	5	5	5	0	0	0
fermi-2	143	164	219	5	5	5	0	0	0
asterix-3	256	295	326	5	5	5	0	0	0
ananda-2	341	471	618	5	5	5	0	0	0
edge	126	162	189	5	5	5	0	0	0
baobab	2	13	25	5	5	5	0	28	85
pisa-1	10	24	57	5	5	5	0	13	27
edge-3	99	121	200	5	5	5	1	2	3
cmu-1	14	28	47	5	5	5	1	8	28
ace	581	642	729	5	5	5	1	1	1
berk-1	63	96	130	5	5	5	0	0	0
pisa-2	224	467	555	5	5	5	0	0	0
fermi-1	219	253	271	5	5	5	1	1	2
cmu-3	498	533	549	5	5	5	1	1	2
asterix-1	26	51	84	5	5	5	0	0	0
ucsb-2	83	85	88	5	5	5	0	1	1
berk-3	109	177	263	5	5	5	1	1	1

Table 5: Configuration 2, fixed rate.

chronized transfers, overlay multicast and overlay multicast with transfer rate control in sequence, one experiment after another without delays, until 10 measurements were taken in each experiment. The average time of one experiment ranged from 2 to 5 minutes. By performing measurements immediately one after another, we tried to minimize the effects of network fluctuation as much as possible.

Node	Link Throughput (KB/s)			Fixed Rate (KB/s)			Buffer Utilization (%)		
	min	avg	max	min	avg	max	min	avg	max
ace									
berk-3	112	296	472	10	10	10	0	0	0
dogmatix	72	140	206	10	10	10	0	0	0
fermi-1	500	554	599	10	10	10	0	0	1
edge-2	115	129	150	10	10	10	0	0	0
asterix-2	376	392	409	10	10	10	0	0	0
cmu-3	1843	1940	1993	10	10	10	0	0	1
berk-2	92	248	457	10	10	10	0	0	0
umn-2	88	171	311	10	10	10	0	0	0
geranium	19	20	22	10	10	10	0	0	1
fermi-2	59	67	72	10	10	10	0	0	0
asterix-3	26	34	44	10	10	10	0	0	0
pisa-2	30	31	33	10	10	10	0	0	1
ananda-2	329	443	571	10	10	10	0	0	1
cmu-2	760	868	948	10	10	10	0	0	0
baobab	122	124	124	10	10	10	0	0	0
cmu-1	39	46	56	10	10	10	0	0	1
umn-1	460	622	797	10	10	10	0	0	1
ananda-1	1039	1357	1860	10	10	10	0	1	1
ucsb-1	19	25	29	10	10	10	0	1	2
berk-1	141	701	1263	10	10	10	0	0	0
pisa-1	585	657	723	10	10	10	0	0	0
edge-3	100	105	109	10	10	10	0	0	0
ucsb-2	69	71	72	10	10	10	0	0	0
asterix-1	90	107	132	10	10	10	0	0	0
edge	119	136	159	10	10	10	0	0	0

Table 6: Configuration 3, fixed rate.

It is clear to see from these tables that for all these three configurations, the rate control mechanism is very effective. All the overlay edges now experience the same throughput. Moreover, the buffer occupancy is strikingly low, and this irrespective of the fact that the local TCP throughputs are quite heterogeneous.

5 Implication on Overlay Protocol Design

In this section, we focus on the implications that the scalability results of the last sections have on the protocol for building overlay trees. Theorems 3 and 6 establish that in order to have acceptable buffer occupancy in each end-system and latency through each overlay edge of a large overlay network, the sending rate of the source has to be limited to some value that is strictly less than the overlay group throughput, which was shown to coincide with the minimum of the local maximal throughput of all overlay edges.

The immediate implication of the above results on the overlay tree construction is that the forwarding paths should be chosen such that the resulting overlay tree has the local maximal throughput of its bottleneck overlay edge maximized. Note that in an overlay network, every node has a logical path or a forwarding edge to every other node. Thus the problem consists in choosing $n-1$ logical edges out of these $n(n-1)$ edges such that

- the chosen $n-1$ edges form a spanning tree;
- the bottleneck overlay edge in the resulting spanning tree has a local maximal throughput as large as possible.

Thus the protocol for designing overlay based reliable group communication has to (i) be aware of the rates on the logical path between any two nodes, (ii) efficiently select those paths that lead to maximizing the group throughput (iii) effectively determine the bottleneck rate to adapt the sending rate of the source. While we do not attempt to provide a detailed solution for developing the complete protocol, we provide insight into these three aspects below.

5.1 Optimal Tree Construction Algorithm

Consider a complete graph $G = (V, E)$. Nodes in the graph correspond to end-systems and (optionally) servers, which are used to build an overlay network. Assume that nodes are numbered from 1 to n, where node 1 is the root, from which data is transmitted. Each pair of nodes $i, j \in V$ is connected via an overlay edge (a route in the Internet) with local saturation throughput θ_{ij}. Although each node can send copies of information to several other nodes simultaneously, it makes sense to assume that the total throughput of each node i for outgoing transmissions is limited by a constant c_i (which is typically determined by the access link connecting node i to the Internet).

We define the throughput of a path P in graph G to be the minimum of θ_{ij} over all links $(i, j) \in P$. From the results of the previous sections, the problem is to find a tree from root with maximum group throughput, where group throughput is by definition the minimum of all path throughputs in the tree.

We consider this problem of overlay tree construction in two cases. In the first case, we ignore the throughput limitation at the access link that was alluded to above. This case refers to the situation when TCP throughput is dominated by the RTT and the loss rate on the path other than the access link. The second model accounts for the bottleneck at the access link. As we shall see, the first case is tractable and it is possible to design an optimal solution for it. The second model results in a minimum degree spanning tree construction which is NP hard.

5.1.1 Model I: Access Link not the Bottleneck

Under the assumption that access link is not the bottleneck, the maximal local throughput θ_{ij} (which we recall to be the TCP throughput that a saturated source located in node i would experience) can be estimated from measurements of the RTT r_{ij} on the edge and the loss probability p_{ij} on the edge using the square root formula for persistent flows. As described above, the construction of overlay tree is a decision process for choosing $n - 1$ edges out of $n(n - 1)$ logical edges. The following algorithm allows one to construct a tree with optimal group throughput:

- Sort all $n(n - 1)/2$ edges in increasing (local maximal) throughput order (assume for sake of simplicity that all throughputs are different, so that the order is total);
- Discard edges starting with those with the smallest throughput until the set of remaining edges on the n nodes makes a connected graph; let $n+1 \leq K(K-1)/2$ be the number of discarded edges when connectedness is lost for the first time;
- Build a spanning tree rooted in the source using the $K(K - 1)/2 - n$ remaining edges of the sorted list.

The resulting spanning tree, say T is optimal as easily shown by contradiction: assume there exists a spanning tree rooted in the source node and that has a better group throughput than T. Then this tree uses none of the $n+1$-st edges of the sorted list. There should then exist a spanning tree from the root to all other nodes and using the $K(K-1)/2-n-1$ last edges of the list, which contradicts the stopping rule used for the definition of n.

5.2 Model II: Accounting for Bottleneck at Access Link

In this model, we account for the case that access links could possibly become the bottleneck due to limited available bandwidth. In practice, this case refers to the situation where forwarding nodes are typically connected to the Internet via DSL/Cable and modem links. The decision problem under this setting is a generalization of the minimum degree spanning tree (which, in turn, is a generalization of Hamiltonian path), and therefore the problem is provably NP-hard. We provide a heuristic that we show to be within $1/2$ of the optimal solution.

5.2.1 Solution Strategy

Suppose we fix target group throughput θ. Next, we can remove from our network G the links that have throughput less than θ, since these links can not participate in any feasible solution. Let us call the new graph $G'_\theta = (V, E'_\theta)$, where $E'_\theta = \{(i,j) \in E : \theta_{ij} \geq \theta\}$. Naturally, while G is a complete graph, G'_θ is not necessarily complete. With θ fixed, the constraints on node throughput for each node i can be treated as degree constraints, allowing the solution to have at most $\lfloor c_i/\theta \rfloor$ links per node. If we can construct a spanning tree T in graph G'_θ, such that T satisfies the degree constraints, T can be used as an overlay routing with throughput θ. We can further use binary search to find the smallest value of θ, for which such a tree can be constructed.

Unfortunately, it is known that the problem of finding a spanning tree T, satisfying degree constraints, in general graphs (or proving that no such tree can be constructed) cannot be solved exactly in polynomial time. Therefore we adopt an approximation algorithm with polynomial running time, proposed in [14] for finding a spanning tree of minimum degree with additive error of at most one.

We are going to show that our algorithm may not be able to construct a tree if degree constraints on one or more nodes are exactly the same as they are in the optimal solution. However, if all degree constraints are such that they allow one more link, than is used in the optimal solution, then the tree will be constructed by our algorithm. In other words, solution will be found for any fixed throughput θ satisfying

$$\frac{c_i}{\theta} \geq \frac{c_i}{\theta^*} + 1 \text{ or, equivalently, } \theta \leq \frac{\theta^* c_i}{\theta^* + c_i}, \tag{10}$$

for all nodes i, where θ^* is the best achievable throughput. Note that any feasible throughput θ and for all nodes i it hold that $\theta \leq c_i$, otherwise node i can not be reached by broadcast with required throughput θ. We can conclude that a feasible solution will be found by our algorithm for any $\theta \leq \theta^*/2$, as $\theta^*/2$ satisfies (10) for any i since : $\frac{\theta^*}{2} = \frac{\theta^* c_i}{2c_i} \leq \frac{\theta^* c_i}{\theta^* + c_i}$.

5.2.2 Fixed Throughput Routing

In this section we describe our generalization of approximation algorithm for minimum degree spanning tree [14]. Our goal is to learn, given a fixed throughput value, whether there exists a routing (i.e. a spanning tree) that allows one to achieve this throughput, and if it exists, to give the routing tree. The described problem is NP-hard, and therefore the solution will be approximate: our algorithm will violate some of degree constraints when constructing the tree. As it is shown in the previous section, if constraint violation can be bounded, objective value can be modified to satisfy the constraints, it is possible to bound required difference in the objective value.

For a given target value of throughput $\theta = \tilde{\theta}$, graph $G'_\theta = (V, E'_\theta)$, and bounds $\{c_i\}_{i \in V}$, our algorithm constructs in polynomial time a spanning tree T in G'_θ, such that degree constraints in T are violated by at most 1 each, provided that there exists a spanning tree satisfying all degree constraints implied by throughput $\tilde{\theta}$.

Let us choose $\tilde{\theta}$ to be the target value of θ, and compute degree constraints d_i for each node i based on this target value:$d_i = \lfloor c_i/\tilde{\theta} \rfloor$.

If for one of the nodes i, the degree limit d_i is 0, the algorithm must report failure, since node i can not be reached and a feasible routing does not exist. Therefore, in the rest of our analysis we will assume that $d_i \geq 1 \quad \forall i$.

The algorithm starts by constructing an arbitrary spanning tree in $G'_{\tilde{\theta}}$, using any simple algorithm; depth first search is a good choice, for example. Then, it computes a set $B \subset V$ of all nodes with maximum degree constraint violation, and tries to reduce cardinality of B by performing series of improvements. For example, if degree constraints are violated by 3, 5 and 7 extra edges, the algorithm will form B of all nodes of the tree that have 7 edges more than it is allowed.

We define improvement as following. Suppose maximum degree violation in our tree is k. Then, if adding an edge connecting two nodes with degree violation less than $k-1$ to the tree, and breaking the loop by removing one edge, incident to one of the nodes with violation k, from the tree, reduces degree violation of one of the nodes in B from k to $k-1$, we say that this operation is an improvement.

The algorithm performs improvements until no improvements are possible, or until B is empty. When B is empty, we build a new set B of nodes with violation $k-1$, and repeat the procedure, until there are no violating nodes or until no improvements are possible.

To prove correctness of the algorithm we need to show that improvements become impossible only when $k \leq 1$. Suppose this is not the case. Then, the algorithm terminates, and nodes in nonempty set B have degree constraint violation k, $k > 1$.

Let C be the set of all nodes that violate degree constraints by at least $k-1$. Notice that $B \subseteq C$, and therefore C is not empty.

Removing all nodes of C from the graph decomposes the graph into at least $K = \sum_{i \in B} d_i + |B|(k-1) + |C \setminus B|(k-2)$ disconnected components, since by definition of B there are total $\sum_{i \in B} d_i + |B|k$ edges in the current spanning tree, going from the nodes in set B, and at least $\sum_{i \in B} d_i + |B|(k-1)$ of these edges are connected to nodes outside of B. If there were links in the graph between nodes on different tree branches connected to nodes in B, those links would allow the algorithm to make an improvement step, and therefore no such links exist, i.e. components are indeed disconnected.

In the optimal (zero-violation) solution, the set of nodes B must have at least K edges going to nodes in $V \setminus B$, at least one for each component, since there are no direct edges between the K components in the graph. Since the optimal solution does not violate degree constraints, but the aggregate degree of nodes in B in the optimal solution is at least $\sum_{i \in B} d_i + |B|(k-1)$, we conclude that $k - 1 \leq 0$, or equivalently $k \leq 1$.

5.3 Decentralized Algorithm using Voronoi Tesselations

In Section 5.1 we provided an optimal tree construction algorithm. It is however not really practical due to its requirement of global knowledge of the network conditions (TCP throughput in particular) between every pair of node. However, scalable tree construction algorithms can be designed for suboptimal solutions. We briefly describe here a suboptimal by decentralized and scalable version of that of Section 5.1 using Voronoi tesselations. It assume that each node can estimate its distance to each other (this distance can be based on either minimal number of hops or minimal cumulated RTTs on the route between the two nodes). Let $M > 1$ be some integer which is a parameter of the algorithm.

Step 1 The source randomly selects a set of first level nodes with cardinality M. This can be done via an independent and decentralized sampling where each node gets elected to the first level with a probability M/K with K the total number of nodes. Ideally, all nodes in this first level set should be far away from reach other (one can keep in mind the idea of one node per continent or per autonomous system). This can be obtained by departing from this random collection of first level nodes with cardinality M and using then a random evolution from this configuration that replaces a node that would be too close to some others (say in terms of number of hops and RTT) by another and the throughput between each pair of nodes (including the source) are large. Let S denote the set composed of the source and the first level nodes. An optimal spanning tree rooted in the source is then built on S using the centralized procedure of the last section.

Step 2 Each remaining (non elected) node pings the nodes of S and selects the closest. Let $\mathcal{K}_k$ denote the set containing the k-th node of S and all nodes that are closer from this node than from any other node in S. The family $\mathcal{K}_k$ forms a partition of the set of remaining nodes. In parallel and for all k,

- apply the same election procedure as that of step 1 but within the subset $\mathcal{K}_k$: this defines a set S_k of second level nodes, with expected cardinality M within each $\mathcal{K}_k$, and a locally optimal spanning tree on set S_k which is rooted on the k-th node of S.
- the nodes of $\mathcal{K}_k$ ping those of S_k, which defines sets $\mathcal{K}_{kj}$ that form a partition of $\mathcal{K}_k$.

The procedure goes on recursively until there are no remaining nodes left.

5.4 Rate Control Mechanisms

The proposed rate control mechanisms seem to be a very good alternative to the back pressure mechanism. They not only exhibit scalable throughput, but also scalable buffer occupancy and

packet delays. The experimental results confirm this. From practical standpoint, several issues need to be considered.

The first one is the rate estimation. As we only need to know what is the smallest local saturation TCP throughput, the edges only need to measure the RTT and packet loss probability and report them back to the source. The source can then determine the critical threshold $\Theta_{1,K}$.

The second one is the rate adaptation. It is well known that the network conditions fluctuate quite a lot. In order to achieve a scalable throughput and a scalable buffer occupancy, one needs to be pessimistic and to consider a worst case scenario by adopting a low rate. A more appealing approach would be to adapt the send rate of the source dynamically. This rate adaptation can be carried out in accordance with the throughput estimation as discussed above.

6 Conclusions

We have presented a new framework for the study of the scalability of overlay based reliable group communication using TCP. In the case of unconstrained overlay buffers with rate control, we have established the scalability of such a paradigm in both the obtained group throughput and the buffer required for arbitrary large group. Experimental results obtained with a prototype validate the theoretical ones.

One of the main scientific contributions of the present paper is the general link that it establishes between the scalability of reliable overlay multicast and the properties of the type of hydrodynamic limits encountered in certain models of statistical physics such as percolation and particle systems. This link has several direct and important implications. For instance, as it was seen in Section 4, one of the key questions of overlay multicast, which is that of the behavior of the buffer contents in end-systems when the size of the multicast group grows large, can actually be obtained by computing the Legendre transform of some hydrodynamic shape as encountered in first passage percolation [16]. In addition, the analysis gives some moment conditions on the cross traffic encountered by a long lived TCP flow in routers that guarantee the actual scalability of buffer contents.

Our results on rate control at the source node and the conditions required to maximize group throughput provide useful insights into the design of scalable reliable group communication protocols using overlays. A first general observation is that in order to maximize the group's throughput, the design of the protocol and the construction of the distribution tree should take into account the local saturation throughput of the TCP connections between end-systems. Another general observation is that rate control combined with TCP congestion control mechanism provides a scalable approach in both throughput and buffer occupancy. Such a combination of rate throttling and congestion control should be considered in the design of efficient and effective reliable overlay multicast schemes.

There are a number of issues that remain to be addressed. The first one that comes to our mind is the scalability issue of any tree topology when a *back pressure* is implemented in each node (i.e. when the buffer of a node is full, this node stops the communication coming from the upstream node). No rate control at the source needs then to be implemented but the throughput can be degradated, furthermore the topology of the tree used between overlay nodes should have a direct impact on the throughput achieved to everybody. This case will be the subject of a companion paper.

Proof of the Latency Scalability Result

These sections describe preliminary results, intermediary lemmas, and give the proofs of the mathematical results contained in Section 4.4.

We treat here the homogeneous case where $H_k = H$ for all k, the law of the window process $(W_m^k)_{m\in\mathbb{Z}}$ is the same for any overlay edge k, and each aggregated service time $s_m^{(k,h)}$ has the same law which is that of the random variable s (with the exception of aggregated service times in overlay nodes of type $s_m^{(k,0)}$ which are set to 0). In addition we assume that s verifies Condition (7).

Notation reminder: For any value of m, k, h, m', k' and h', we denote by

- $P_{(m,k,h)\to(m',k',h')}$ the set made with the paths in the graph from vertex (m,k,h) to vertex (m',k',h');
- $\texttt{Wei}_{((m,k,h)\to(m',k',h')}$ the quantity $\max_{\pi\in P_{(m,k,h)\to(m',k',h')}} \texttt{Wei}(\pi)$;
- $\texttt{Wei}_{(m,k)\to(m',k')}$ the quantity $\texttt{Wei}_{(m,k,0)\to(m',k',0)}$.

Since $s_m^{(k,0)}$ is supposed null for any k and m,

$$\texttt{Wei}_{(m,k,H_k)\to(m',k',0)} = \texttt{Wei}_{(m,k+1)\to(m',k')} \ .$$

I Preliminary Results on the Random Graph

A Lattice Animals and their Links with Paths in the Random Graph

Two vertices (m,k,h) and (m',k',h') are said *adjacent* if:

- either $(k,h) = (k',h')$ and $|m-m'| = 1$,
- or $m = m'$, $k = k'$ and $|h-h'| = 1$,
- or $m = m'$, $k = k'+1$, $h = 0$, and $h' = H_{k'}$,
- or $m = m'$, $k' = k+1$, $h = H_k$, and $h' = 0$.

A *lattice animal* in the random graph is a connected set of adjacent vertices: this means that for all pairs of vertices (m,k,h) and (m',k',h') taken in this set, one can progress from one to the other while staying in the set, using only pairs of adjacent vertex.

The *size* of a lattice animal, denoted by $|\xi|$ is the number of vertices contained in it, and its *weight* denoted $\texttt{Wei}(\xi)$ is the sum of the weights of its vertices. This definition allows us to have lattice animals defined on our graph equivalent to classical lattice animals defined in $\mathbb{Z}^2$.

In particular, the weight of such lattice animals containing a fixed vertex (m,k,h) is asymptotically related to their size by the following result, shown in [22] :

Theorem 7 *Under Condition (7) on s, we have*

$$\lim_{N\to\infty} \sup \frac{1}{N} \left(\max_{\{|\xi|=N \text{ and } (m,k,h)\in\xi\}} \texttt{Wei}(\xi) \right) \leq \texttt{Cste} < \infty \ .$$

Lattice animals and paths in our random graph can be related by the following result :

Lemma 2 *Given a path $\pi \in P_{(m,k,h)\to(m',k',h')}$, we can show that there exists a lattice animal which contains π and whose size is bounded by $(2H+W_{\max}-1)(m-m')+(H+1)(k-k')+(h-h')$.*

Proof: We define $\psi(m,k,h) = Hm + (H+1)k + h$.
One can check that for any edge $(m_i,k_i,h_i) \to (m_{i+1},k_{i+1},h_{i+1})$ in our random graph,

$$\psi(m_{i+1},k_{i+1},h_{i+1}) < \psi(m_i,k_i,h_i) \ ,$$

so that the length of π is bounded by

$$|\pi| \leq H(m-m') + (H+1)(k-k') + (h-h') \ .$$

The subset made with the vertices included in the path π is not in general a lattice animal as a pair of successive vertex in the path may not be adjacent when the edge used in the path is of the form $(m_i,k_i,1) \to (m_i - W_{m_i}^{(k_i)}, k_i, H_{k_i})$. However, there cannot be more than $(m-m')$ edges of this type included in path π, and for each one of them, we can add the $H_{k_i}-1$ vertices

$$(m_i,k_i,2),\ldots,(m_i,k_i,H_{k_i})$$

and then the $W_{m_i}^{(k_i)}$ vertices

$$(m_i-1,k_i,H_{k_i}),\ldots,(m_i-W_{m_i}^{(k_i)}-1,k_i,H_{k_i})$$

(if they are not included already in the path) to the vertices of the path, in order to obtain a connected subset of adjacent vertices.

As a consequence we can complete the subset made of vertices appearing in π with at most $(m-m')(H+W_{\max}-1)$ vertices to obtain a lattice animal ξ, which proves our lemma. □

B Concentration of Measure

Concentration of measure allows one to bound $P\left(|Z-\mathbb{E}[Z]|>u\right)$ when Z is a maximum of sums of independent and bounded random variables. We will apply this to a graph with truncated weights.

Truncation : For any k,h and m, we denote by symbol $\check{}$ the truncated version of the service time :

$$\check{s}_m^{(k,h)} = \min\left(s_m^{(k,h)}, (\max(|k|(H+1)+h,|m|))^{\frac{1}{4}}\right)$$

In particular the following result on lattice animals, already proven p.18 in [23], bounds the distance between the initial random graph and its truncated version.

Lemma 3 *for any (m,k,h) we have almost surely :*

$$\frac{1}{N}\left(\max_{\{|\xi|=N \text{ and } (m,k,h)\in\xi\}} \mathtt{Wei}(\xi) - \check{\mathtt{Wei}}(\xi)\right) \to 0.$$

Consider paths $(m,k,h) \to (m',k',h')$ in the graph; each one contains at most $H(m-m')+(H+1)(k-k')+(h-h')$ vertices (a number that we can bound from above by $R=(H+1)(|m|+|m'|+|k|+|k'|+1)$). For a vertex (m_i,k_i,h_i) of the path, we have $m \geq m_i \geq m'$ and $k \geq k_i \geq k'$, hence

$$\begin{aligned} 0 \leq \check{s}_{m_i}^{(k_i,h_i)} &\leq \max\left(|k|(H+1)+H, |m|, |k'|(H+1)+H, |m|\right)^{\frac{1}{4}} \\ &\leq R^{\frac{1}{4}} . \end{aligned}$$

We can then apply Lemma 5.6 in [23] and prove

Lemma 4 *For any k,h,m,k',h',m' we have*

$$P(|\check{\mathtt{Wei}}_{(m,k,h)\to(m',k',h')} - \mathbb{E}[\check{\mathtt{Wei}}_{(m,k,h)\to(m',k',h')}]| \geq u)$$

$$\leq \exp\left(-\frac{u^2}{16(H+1)^{\frac{3}{2}}(|m|+|m'|+|k|+|k'|+1)^{\frac{3}{2}}}+64\right).$$

II Hydrodynamic Limit in the Saturated Case

In the saturated case ($\lambda=\infty$), all packets $m=1,2,\ldots$ are immediately available to be transmitted, or equivalently in our graph, vertices of the line $k=0$ all have null weights. We can then write :

$$D^{\infty}_{m,k} = \mathtt{Wei}_{(m,k,H_k)\to(1,1,0)} = \mathtt{Wei}_{(m,k+1)\to(1,1)}.$$

In this section, we study the growth rate of this delay when the packet index growths proportionally to the overlay edge index, i.e. the growth rate of $D^{\infty}_{\lfloor xk\rfloor,k}$ when k grows large. The following result shows that under Condition (7) on s, this delay grows as a linear function of k.

Theorem 8 *Assume Condition (7) holds; then for any rational number $x \geq 0$:*

$$\lim_{k\to\infty} \frac{D^{\infty}_{\lfloor xk\rfloor,k}}{k} \text{ is a finite constant} = \gamma(x), \quad \text{a.s. and in } \mathbb{L}_1. \tag{11}$$

In addition, for any m and k, we have

$$\mathbb{E}[D^{\infty}_{m,k}] \leq k\gamma(\frac{m}{k}) . \tag{12}$$

The function γ is nondecreasing.

Proof: We adapt the proof of Theorem 6.3 in [15]:

- `Step-1` : Let us first prove that for all integers p, q :

  $$\frac{D^{\infty}_{pk,qk}}{k} \text{ converges to a constant} = \eta(p,q) \text{ a.s. and in } \mathbb{L}_1.$$

  First we define the following sequence

  $$\begin{array}{lll} X_{k,k'} & = \mathtt{Wei}_{(pk',qk'+1)\to(pk+1,qk+1)} & \text{for } k < k', \\ & = 0 & \text{for } k \geq k', \end{array}$$

  It is superadditive as $X_{k,k'} + X_{k',k''}$ can be seen as the maximal weight of a path from $(pk'', qk''+1, 0)$ to $(pk'+1, qk'+1, 0)$ added to the maximal weight of a path from $(pk', qk'+1, 0)$ to $(pk+1, qk+1, 0)$. These vertices can then be completed into a path from $(pk'', qk''+1, 0)$ to $(pk+1, qk+1, 0)$.

  Any path in the random graph from $(pk, qk+1, 0)$ to $(1, 1, 0)$ can be included in a lattice animal ξ with size

  $$|\xi| \leq (H + W_{\max} - 1)(pk - 1) + (H + 1)qk + H \leq O(k) \ .$$

  As a consequence from Theorem 7, Condition (7) then implies

  $$\limsup_{k\infty} \frac{X_{0,k}}{k} \leq \mathtt{Cste} < \infty.$$

  It is easy to check that for all k, the sequence $X_{n,n+k}$ is stationary and mixing in n, thanks to our independence assumptions. Kingman's superadditive theorem can be applied and $\frac{X_{0,k}}{k}$ converges a.s. and in $\mathbb{L}_1$ to a deterministic limit, that we denote by $\eta(p,q)$. The result is proven once we have observed $X_{0,k} = D^{\infty}_{kp,kq}$.

  The expectations $\mathbb{E}[X_{0,2^k}]/2^k$ is non-increasing with k, hence we obtain for $p = m, q = k$: $\mathbb{E}[D^{\infty}_{m,k}] = \mathbb{E}[X_{0,1}] \leq \eta(m,k)$

- `Step-2`: For $x = \frac{p}{q} \in \mathbb{Q}$, we have:

  $$D^{\infty}_{p\lfloor k/q \rfloor, q\lfloor k/q \rfloor} \leq D^{\infty}_{\lfloor (p/q)k \rfloor, k} \leq D^{\infty}_{p\lfloor (k/q)+1 \rfloor, q\lfloor (k/q)+1 \rfloor} \ ,$$

  hence with $\gamma(x) = \frac{1}{q}\eta(p,q)$, limit (11) extends to all rational numbers.

  Consequently, the previous inequality in expectation becomes:

  $$\mathbb{E}[D^{\infty}_{m,k}] \leq \eta(m,k) = k\gamma(\frac{m}{k}).$$

□

The following lemma is a direct consequence from the property we assumed on gamma that will be used in the next proof:

Lemma 5 *Assume that the γ function is concave; its right derivative can then be defined; it is non-negative, non-increasing, hence admitting a limit in infinity. We suppose in addition that*

$$\lim_{x\to\infty} \gamma'(x) = \theta^{-1}.$$

Then for $\theta^{-1} < \frac{1}{\lambda} - \mu < \frac{1}{\lambda}$, there exists a ζ such that for $x \geq \zeta$,

$$\gamma(x) - \frac{1}{\lambda}x \leq -\mu x.$$

N.B.: In a tandem of single server queues, the previous properties assumed on function γ can be directly deduced from its definition. We have no proof at this time to show that they remain true in general for a tandem of TCP connections. We have checked empirically by simulation that in all the cases we have studied, the function γ satisfies these properties.

III Stationary Regime with Rate Control & Proof of Theorem 6

Let us now consider the case of a general emission point process from the source, with rate λ, where the packets $m = 1, 2, \ldots$ are available to be transmitted in the first end-system at times $T_1, T_2, \ldots$. Then

$$\begin{aligned} D^{\lambda}_{m,k} &= \mathtt{Wei}_{(m,k,H_k)\to(1,0,0)} = \mathtt{Wei}_{(m,k+1)\to(1,0)} \\ &= \max_{1\leq m'\leq m} \{\mathtt{Wei}_{(m,k+1)\to(m',1)} + T_{m'}\}. \end{aligned}$$

Denoting by $\tau_m = T_{m+1} - T_m$ the inter-arrival sequence, we get:

$$D^{\lambda}_{m,k} = T_m + \max_{1\leq m'\leq m} \{\mathtt{Wei}_{(m,k+1)\to(m',1)} - \sum_{m''=m'}^{m-1} \tau_{m''}\}.$$

Building the stationary regime Let us now assume that the communication starts with an empty system for packet $m = -M$; we then have:

$$R_{m,k}(M) = \max_{-M\leq m'\leq m} \{\mathtt{Wei}_{(m,k+1)\to(m',1)} - \sum_{m''=m'}^{m} \tau_{m''}\}.$$

As $-M$ goes to $-\infty$, the $\max$ ranges over an increasing domain, so that $R_{m,k}(M)$ increases to

$$\tilde{R}_{m,k} = \sup_{m'\leq m} \{\mathtt{Wei}_{(m,k+1)\to(m',1)} - \sum_{m''=m'}^{m} \tau_{m''}\}. \tag{13}$$

It is easy to check that whenever $\lambda < \theta$, then the random variable $\tilde{R}_{m,k}$ is a.s. finite for all m and k, and also that its law is not depending any more on m. The process $\{\tilde{R}_{m,k}\}_m$ is the stationary latency process to overlay edge k.

Rewriting the result of Theorem 6 We have to prove that almost surely :

$$\lim_{k\to\infty}\frac{\tilde{R}_{m,k}}{k}=\lim_{k\to\infty}\frac{\tilde{R}_{0,k}}{k}=\sup_{x\geq 0}\{\gamma(x)-\frac{1}{\lambda}x\}.$$

Proof: From Equation (13) giving $\tilde{R}_{0,k}$, this is equivalent to :

$$\frac{1}{k}\sup_{m\geq 0}\{\mathtt{Wei}_{(0,k+1)\to(-m,1)}-\sum_{m'=0}^{m}\tau_{-m'}\}\to\sup_{x\geq 0}\{\gamma(x)-\frac{1}{\lambda}x\}. \tag{14}$$

To prove the last equation, we adapt the argument of Theorem 5.2 given in [23].
As a consequence from Lemma 5, since

$$\lim_{x\to\infty}\gamma'(x)<\frac{1}{\lambda}-\mu<\frac{1}{\lambda},$$

we can find ζ such that :

$$\gamma(x)-\frac{1}{\lambda}x\leq-\mu x \text{ for } x\geq\zeta,$$

and thus

$$\sup_{x\geq 0}\{\gamma(x)-\frac{1}{\lambda}x\}=\sup_{0\leq x\leq\zeta}\{\gamma(x)-\frac{1}{\lambda}x\}.$$

The result of the theorem is then the consequence of the following two a.s. limits :

$$\frac{1}{k}\max_{0\leq m\leq\zeta k}\left\{\mathtt{Wei}_{(0,k+1)\to(-m,1)}-\sum_{m'=0}^{m}\tau_{-m'}\right\}\to\sup_{0\leq x\leq\zeta}\{\gamma(x)-\frac{1}{\lambda}x\} \tag{15}$$

and

$$\left[\frac{1}{k}\sup_{m\geq\zeta k}\left\{\mathtt{Wei}_{(0,k+1)\to(-m,1)}-\sum_{m'=0}^{m}\tau_{-m'}\right\}\right]_{+}\to 0. \tag{16}$$

- `Step-1` : Let us first prove Equation (15).

  As a consequence from Lemmas 2 and 3 :

  $$\frac{1}{k}\max_{0\leq m\leq\zeta k}\left|\mathtt{Wei}_{(0,k+1)\to(-m,1)}-\check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)}\right|\to 0 \text{ a.s.}$$

  As proved by Lemma 5.8 in [23] we have :

  $$\frac{1}{k}\max_{0\leq m\leq\zeta k}\left|\sum_{m'=0}^{m}\tau_{-m'}-\frac{m}{\lambda}\right|\to 0 \text{ a.s.}$$

  Hence the following result is sufficient to prove (15) :

  $$\frac{1}{k}\max_{0\leq m\leq\zeta k}\left\{\check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)}-\frac{m}{\lambda}\right\}\to\sup_{0\leq x\leq\zeta}\{\gamma(x)-\frac{1}{\lambda}x\} \tag{17}$$

Let us first prove that this is true if $\mathtt{W\check{e}i}$ is replaced by its expectation : Equation (12) shown in Theorem 8 gives :

$$\begin{aligned}\frac{\mathbb{E}[\mathtt{W\check{e}i}_{(0,k+1)\to(-m,1)}] - \frac{m}{\lambda}}{k} &\leq \frac{\mathbb{E}[\mathtt{W\check{e}i}_{(m+1,k+1)\to(1,1)}]}{k} - \frac{m}{k}\frac{1}{\lambda} \\ &\leq \frac{\mathbb{E}[\mathtt{Wei}_{(m+1,k+1)\to(1,1)}]}{k} - \frac{m}{k}\frac{1}{\lambda} \\ &\leq \frac{\mathbb{E}[D^{\infty}_{m+1,k}]}{k} - \frac{m}{k}\frac{1}{\lambda} \\ &\leq \gamma\left(\frac{m+1}{k}\right) - \frac{m}{k}\frac{1}{\lambda}.\end{aligned}$$

Hence :

$$\lim_{k\to\infty}\sup\max_{0\leq m\leq \zeta k}\frac{\mathbb{E}[\mathtt{W\check{e}i}_{(0,k+1)\to(-m,1)}] - \frac{m}{\lambda}}{k} \leq \sup_{0\leq x\leq\zeta}\{\gamma(x) - \frac{1}{\lambda}x\}.$$

For the other bound, we use :

$$|\frac{\mathbb{E}[\mathtt{Wei}_{(0,k+1)\to(-\lfloor xk\rfloor,1)}]}{k} - \gamma(x)| = |\frac{\mathbb{E}[D^{\infty}_{\lfloor xk\rfloor+1,k}]}{k} - \gamma(x)| \to 0 \text{ a.s.}$$

which implies by Lemma 3 and dominated convergence that :

$$|\frac{\mathbb{E}[\mathtt{W\check{e}i}_{(0,k+1)\to(-\lfloor xk\rfloor,1)}]}{k} - \gamma(x)| \to 0 \text{ a.s.}$$

Hence

$$\lim_{k\to\infty}\inf\max_{0\leq m\leq \zeta k}\frac{\mathbb{E}[\mathtt{W\check{e}i}_{(0,k+1)\to(-m,1)}] - \frac{m}{\lambda}}{k} \geq \sup_{0\leq x\leq\zeta}\{\gamma(x) - \frac{1}{\lambda}x\}$$

finishing the proof of :

$$\lim_{k\to\infty}\max_{0\leq m\leq \zeta k}\frac{\mathbb{E}[\mathtt{W\check{e}i}_{(0,k+1)\to(-m,1)}] - \frac{m}{\lambda}}{k} = \sup_{0\leq x\leq\zeta}\{\gamma(x) - \frac{1}{\lambda}x\}.$$

To conclude the proof of (17), for all $\varepsilon > 0$, we have :

$$\begin{aligned}&\sum_{m=0}^{\lfloor\zeta k\rfloor} P\left(|\mathtt{W\check{e}i}_{(0,k+1)\to(-m,1)} - \mathbb{E}[\mathtt{W\check{e}i}_{(0,k+1)\to(-m,1)}]| > \varepsilon k\right) \\ &\qquad \leq \textstyle\sum_{m=0}^{\lfloor\zeta k\rfloor}\exp\left(-\frac{(\varepsilon K)^2}{16(H+1)^{\frac{3}{2}}(m+k+2)^{\frac{3}{2}}} + 64\right) \\ &\qquad \leq (\zeta k+1)\exp\left(-\frac{\varepsilon^2 K^{\frac{1}{2}}}{16(H+1)^{\frac{3}{2}}(1+\zeta)^{\frac{3}{2}}} + 64\right).\end{aligned}$$

The RHS for $k = 1, 2, \ldots$ are summable. By the Borel Cantelli lemma, this implies that for k large enough,

$$\max_{0 \le m \le \zeta k} \frac{\mathtt{Wei}_{(0,k+1)\to(-\lfloor xk \rfloor,1)} - \frac{m}{\lambda}}{k}$$

differs of its expectation by at most ε. The a.s. limit of the expectation hence implies the a.s. limit of Equation (17).

- `Step-2` : To prove Equation (16), let us first choose a ν such that $\frac{1}{\lambda} - \mu < \nu < \frac{1}{\lambda}$. Then $\sum_{m'=0}^{m} \tau_{-m'} - \nu m$ is a.s. positive for a sufficiently large m, and the following result is sufficient to conclude :

$$\left[\frac{1}{k} \sup_{m \ge \zeta k} \left\{ \mathtt{Wei}_{(0,k+1)\to(-m,1)} - \nu m \right\} \right]_+ \to 0 \text{ a.s.} \tag{18}$$

From (12), we have with $\delta = (\nu - (\frac{1}{\lambda} - \mu))/2 > 0$:

$$\begin{aligned} \mathbb{E}[\check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)}] &\le \mathbb{E}[\mathtt{Wei}_{(0,k+1)\to(-m,1)}] = \mathbb{E}[D^{\infty}_{m+1,k}] \\ &\le k\gamma(\frac{m+1}{k}) \le k\left(\frac{1}{\lambda}\frac{m+1}{k} + \mu\frac{m+1}{k}\right) \\ &\le \nu(m+1) - 2\delta(m+1). \end{aligned}$$

This implies in particular :

$$\begin{aligned} \mathtt{Wei}_{(0,k+1)\to(-m,1)} - \nu m &\le \mathtt{Wei}_{(0,k+1)\to(-m,1)} - \check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)} - \delta m \\ &\quad + \check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)} - \mathbb{E}[\check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)}] - \delta m \\ &\quad + (\nu - 2\delta). \end{aligned} \tag{19}$$

First line of RHS in (19) : As a consequence from Lemmas 2 and 3,

$$\frac{1}{m} \max_{K < \frac{m}{\zeta}} \left(\mathtt{Wei}_{(0,k+1)\to(-m,1)} - \check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)} \right)$$

vanishes a.s. for $m \to \infty$. Hence there is only a finite number of m such that :

$$\max_{K < \frac{m}{\zeta}} \left(\mathtt{Wei}_{(0,k+1)\to(-m,1)} - \check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)} \right) \ge \delta m.$$

This proves that for k sufficiently large : for all $m > \zeta k$,

$$\left(\mathtt{Wei}_{(0,k+1)\to(-m,1)} - \check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)} \right) < \delta m.$$

Second line of RHS in (19) : As a consequence from Lemma 4,

$$\begin{aligned} &P(\check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)} - \mathbb{E}[\check{\mathtt{Wei}}_{(0,k+1)\to(-m,1)}] \ge \delta m) \\ &\le \exp\left(-\frac{(\delta m)^2}{16(H+1)^{\frac{3}{2}}(m+k+2)^{\frac{3}{2}}} + 64 \right) \end{aligned}$$

so that the probability of the event

$$\left\{ \frac{\max\limits_{K<\frac{m}{\zeta}} \left(\check{\mathtt{We}}\mathtt{i}_{(0,k+1)\to(-m,1)} - \mathbb{E}[\check{\mathtt{We}}\mathtt{i}_{(0,k+1)\to(-m,1)}] \right)}{m} \geq \delta \right\}$$

is less than

$$\left(1+\frac{m}{\zeta}\right) \exp\left(-\frac{\delta^2}{16(H+1)^{\frac{3}{2}}(3+\frac{1}{zeta})^{\frac{3}{2}}} m^{\frac{1}{2}} + 64 \right).$$

This is summable in m, so that a.s. this occurs for a finite number of m. Hence, a.s. for k large enough :

$$\frac{1}{k} \sup_{m \geq \zeta k} \left\{ \check{\mathtt{We}}\mathtt{i}_{(0,k+1)\to(-m,1)} - \mathbb{E}[\check{\mathtt{We}}\mathtt{i}_{(0,k+1)\to(-m,1)}] - \delta m \right\}$$

is negative. This implies Equation (18) and completes the proof of the theorem.

□

References

[1] F. Baccelli, A. Borovkov and J. Mairesse, *Asymptotic results on infinite tandem queueing networks*, Probability and Related Fields 118, p.365-405, October 2000.

[2] F. Baccelli and P. Brémaud, *Elements of Queuing Theory*, Springer Verlag (2nd edition 2002).

[3] F. Baccelli, G. Cohen, G. Olsder, J.P. Quadrat *Synchronization and Linearity*, Wiley, 1992.

[4] F. Baccelli and D. Hong, *TCP is Max-Plus Linear and what it tells us on its throughput*, ACM Sigcomm 2000, p.219-230.

[5] S. Banerjee, B. Bhattacharjee and C. Kommareddy, *Scalable Application Layer Multicast*, in Proceedings of ACM Sigcomm 2002.

[6] C.Bormann, J.Ott, H.-C. Gehrcke, T.Kerschat and N. Seifert, *MTP-2: Towards Achieving the S.E.R.O. Properties for Multicast Transport*, International Conference on Computer Communications and Networks (ICCCN 94), 1994

[7] Y. Chawathe, S. McCanne, and E. A. Brewer, *RMX: Reliable Multicast for Heterogeneous Networks*, in Proceedings of IEEE Infocom, 2000.

[8] A. Chaintreau, F. Baccelli and C. Diot, *Impact of TCP-like Congestion Control on the Throughput of Multicast Group*, IEEE/ACM Transactions on Networking vol.10, p.500-512, August 2002.

[9] Y.-H. Chu, S. G. Rao, and H. Zhang, *A Case for End System Multicast*, in Proceedings of ACM SIGMETRICS, June 2000.

[10] Yang Chu, Sanjay Rao, Srinivasan Seshan, Hui Zhang, *Enabling conferencing applications on the Internet using an overlay multicast architecture*, in Proceedings of ACM Sigcomm, August 2001.

[11] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, *A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing*, in IEEE/ACM Transactions on Networking, December 1997, Volume 5, Number 6, pp. 784-803.

[12] P. Francis, *Yallcast: Extending the Internet multicast architecture*, in http://www.yallcast.com (September 1999).

[13] P. Francis, *Yoid: Extending the Internet Multicast Architecture*, http://www.icir.org/yoid/docs/yoidArch.ps.gz (April 2000).

[14] M. Fürer, B. Raghavachari, *Approximating the minimum degree spanning tree to within one from the optimal degree*, in Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms, Orlando, Florida, United States, pp. 317–324, ACM Press (1992).

[15] P. Glynn and W. Whitt, *Departures from many queues in series*, Annals Appl. Prob., 1(4):546–572, 1991.

[16] G. Grimmett, *Percolation*, Springer Verlag 1999.

[17] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole, *Overcast: Reliable Multicasting with an Overlay Network*, in Proceedings of the 4th Symposium on Operating Systems Design and Implementation, Oct. 2000.

[18] Kirk L. Johnson, John F. Carr, Mark S. Day, and M. Frans Kaashoek, *The measured performance of content distribution networks*, in Proceedings of the 5th WCW (2000).

[19] T.V. Lakshman and U. Madhow, *The performance of TCP/IP for networks with high bandwidth-delay products and random loss*, in IEEE/ACM Transactions on Networking, 5-3, pp. 336-350 (1997).

[20] B.N. Levine and J.J. Garcia-Luna-Aceves, *A Comparison of Reliable Multicast Protocols*, ACM Multimedia Systems, August 1998.

[21] J. Liebeherr, M. Nahas, *Application-layer Multicast with Delaunay Triangulations*, To appear in JSAC, special issue on multicast, 2003.

[22] J. Martin, *Linear Growths for Greedy Lattice Animals*, Stochastic Processes and their Applications, vol. 98, no. 1, pp. 43-66 (2002)

[23] J. Martin, *Large Tandem Queueing Networks With Blocking, Queueing Systems, Theory and Applications*, vol. 41, pp. 45-72 (2002).

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, *Modeling TCP Throughput: a Simple Model and its Empirical Validation*, in Proceedings of ACM SIGCOMM, August 1998.

[25] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, *ALMI: An Application Level Multicast Infrastructure*, in 3rd Usenix Symposium on Internet Technologies and systems (USITS), March 2001.

[26] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A scalable content-addressable network* in Proceedings of ACM SIGCOMM (August 2001).

[27] E. M. Schooler, *Why Multicast Protocols (Don't) Scale: An Analysis of Multipoint Algorithms for Scalable Group Communication*, Ph.D. Dissertation, Computer Science Department, 256-80 California Institute of Technology, Sept. 2000.

[28] S. Shi and J. S. Turner, *Multicast Routing and Bandwidth Dimensioning in Overlay Networks*, IEEE JSAC (2002).

[29] S. Shi and J. Turner, *Placing Servers in Overlay Net-works*, Technical Report WUCS-02-05, Washington University, 2002.

[30] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. *Chord: A scalable peer-to-peer lookup service for Internet applications*, in Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications, 2001, pp.149–160, Diego, California, United States.

[31] G. Urvoy-Keller and E. W. Biersack, *A Multicast Congestion Control Model for Overlay Networks and its Performance*, in NGC 2002, October 2002.

[32] B. Zhang, S. Jamin, L. Zhang, *Host Multicast: A Framework for Delivering Multicast To End Users*, in Proceedings of IEEE Infocom (2002).



Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399