Scalability of Reliable Group Communication Using Overlays

François Baccelli† Augustin Chaintreau† Zhen Liu‡ Anton Riabov†† Sambit Sahu‡

† INRIA & ENS
45, rue d'Ulm
75005 Paris FRANCE
francois.baccelli@ens.fr,
augustin.chaintreau@ens.fr

‡ IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights
NY 10598, USA
zhenl@us.ibm.com,
sambits@us.ibm.com

†† Columbia University, IEOR Dept
540 W 120 st, New York
NY 10027, USA
avr11@columbia.edu

***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 using different unicast TCP connections. In this paper we assume that buffers in end-systems are large enough for the transfers. 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 under natural conditions. 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 explodes with time. It is then shown that 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.**

I. Introduction

Reliable group communication has remained an important research problem for the last decade. Significant effort has been spent on the design and evaluation of reliable multicast transport protocols, see for example [11], [7], [18] 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 [24], [9] 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 [10], Yoid [12], ALMI [23], Host Multicast [28], NICE [6], Delauney graph [19], and [26], [25].

Reliable multicast can also be implemented in overlay using point-to-point TCP connections. In Overcast [16], HTTP connections are used in between end-systems. In RMX [8], TCP sessions are directly used. The main advantage of such approaches is the ease of deployment. In addition, [8] 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 [8], [16] 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 [27], the authors investigated this scalability issues while considering a TCP-friendly congestion control mechanism with fixed window-size for the point-to-point reliable transfer. 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. Using theoretical investigations, experimentations in the Internet, and simulations of large networks, we examine the scalability of three key features of such overlays with respect to the size of the group: throughput of the group communication; delay of packets to reach end-systems; buffer requirements at the end-systems.

We first examine the dependency of the throughput on group size and on the network connectivity. For this, we 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. [5] and the references therein) under the assumption of infinite buffer space at each intermediate end-system. Each TCP connection is represented by a set of FIFO routers in series where some marking scheme is used for controlling the sources. 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. 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 the following more difficult question: does there exist mechanisms that can achieve both a scalable (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) bounded 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. We also show that the mean delay spent in each end system, or equivalently the mean buffer occupancy in each end system, can be evaluated by computing the Legendre transform of some hydrodynamic limit associated with the saturated source case.

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 found 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 maximum throughput* (see definition below) of the TCP connections between end systems.

The paper is organized as follows. Section II defines the problems under consideration and presents the notation and the mathematical models. In Section III, 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 IV we introduce the rate-control based protocol allowing one to bound the buffer occupancy and the latency for any arbitrary group size. In Section V, 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 VI summarizes the work.

II. Modeling Overlay Group Communication

A. 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 §II-C) on which a feedback control mechanism is enforced, this level of abstraction considers the path as a direct edge in the overlay network. While it is not required, we assume that the nodes are connected in a tree topology. As illustrated in Figure 1, after receiving data from its parent node in the overlay network, a node 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 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 store it and forward it on a per-connection basis using established TCP connections for each of its downstream nodes in the overlay tree.

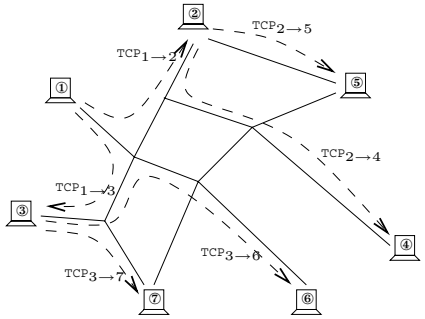

Fig. 1. A binary tree overlay network

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 in order to play its relay role for the group communication.

B. 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 depends upon the network conditions of the underlying paths between the nodes and also on the buffer capacity at each

node. Some understanding on this dependency is needed in order to determine the conditions under which overlay based reliable group communication can scale in group size. Also this understanding would help us design appropriate overlay topology to maximize the group throughput.

Specifically, our study examines the following:

- how the group throughput is related to the local maximum throughput of TCP connections at the overlay edges, where this local maximum throughput of TCP connection is defined as the throughput achieved with an infinite source 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 [5], 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 max-plus algebra ([4]) 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 is primarily 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 [21].

It should be stressed that most of the techniques of [4] cannot be used directly for the present study. First the fixed support assumptions of Chapters 7 and 8 of [4] do not hold here because of the varying window size. Second, a 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 [4] either.

C. A Model for TCP Connections in Tandem

We shall first consider a special case of the overlay topology which is the chain topology. The general topology is considered in the next section.

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, denoted as node 0, 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 $m = 1, 2, \dots$). We shall assume throughout the paper that the inter-arrival times $\{T_m - T_{m-1}\}_m$ form a stationary and ergodic sequence[1] so that the limit $\lambda \equiv \lim_{m\to\infty} m/T_m$ exists with probability 1. This constant denotes the arrival intensity of the packets at the source. If all the packets are available at time 0 (i.e. $T_m \equiv 0$), we have $\lambda = \infty$. We refer to this case as the saturated case.

The (TCP) connection from node 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 $h = 1, \dots, H_k$, which are modeled by single server queues. The TCP congestion control is characterized by the variable $W_m^{(k)}$, representing the window size as seen by packet m at node k. This node is allowed to transmit packet m when packet $m - W_m^{(k)}$ is received by node $k + 1$. We assume that the routers use a marking scheme for letting the TCP connection adapt to local variations of traffic. The case of a loss system is not considered here (for loss based congestion control mechanisms, an additional resequencing mechanism has to be implemented and represented at the output of each TCP connection at the occurrence of a loss in order to fulfill the increasing packet sequence number requirement).

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 (say from node k to node $k + 1$) in the router plus the additional waiting time due to cross traffic packets interleaved between two packets of the reference connection. Such a modeling approach was also used in [9]. We denote by $s_m^{(k,h)}$ the aggregated service time experienced by the m_th packet going through the h-th router of the overlay edge k.

The node (end-system) itself is modeled as a single server queue whose service time (denoted by $s_m^{(k,0)}$) can take into account the time to copy an incoming packet to an outgoing queue inside the end-system. Typically, this time is negligible compared to end-to-end round trip times ; in the rest of the paper we assume $s_m^{(k,0)} \equiv 0$.

2) Evolution Equations and Longest Paths in a Graph: We follow the TCP modeling approach proposed in [5]. For each TCP connection, we establish the evolution equations governing the packet departure times. TCP window size's evolution is governed by independent packet marking 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:

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

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

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

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

In words, these equations state that in order to serve packet m, the previous 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, the transmission should be allowed by the TCP congestion control.

[1] This level of generality is required as even if one takes a renewal process as input of some overlay network, the output process of the overlay is stationary and ergodic but not a renewal sequence. For more on the matter, see e.g. [2] and the comments following Assumption 1 below.

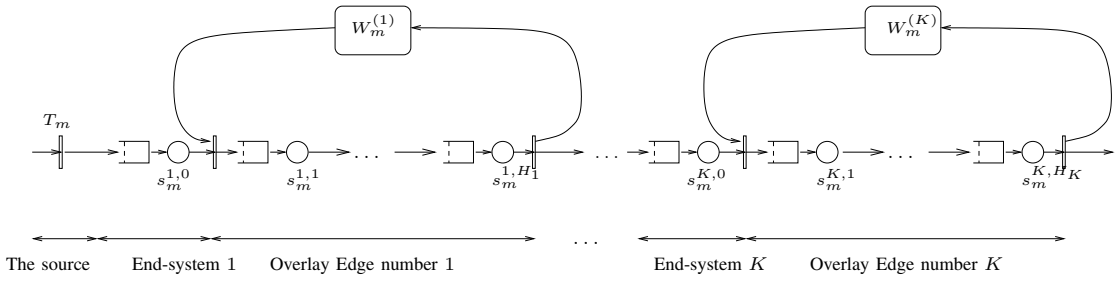

Fig. 2. TCP connections in tandem

The presence of packet marking is taken into account via the evolution of the window size $(W_m^{(k)})_{m\in\mathbb{Z}}$, which is governed by the AIMD rule of the congestion avoidance phase of RENO. The window sequences is assumed independent for different overlay edges. For each edge k, we consider a Markov chain made of $(W_m^{(k)}, r_m^{(k)})_{m\in\mathbb{Z}}$ with

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

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 marking process is Markovian, the joint process of the packet marking and $(W_m^{(k)}, r_m^{(k)})$ is Markovian as well. In particular, when the packet markings are Bernoulli, transition of $(W_m^{(k)}, r_m^{(k)})$ are 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 marking 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 $(W_m^{(1)}, r_m^{(1)}, \dots, W_m^{(K)}, r_m^{(K)})_{m\in\mathbb{Z}}$.

Note that other features of TCP such as timeout, retransmission, acknowledgment packet delays, can also be taken into consideration in these equations in the way presented in [5].

As one can observe from Equations (1–4), only the operators maximum and plus are used. Thus, as in [5], 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 the max-plus 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 $V = \{(m,k,h) | m \ge 1,\ 0 \le k \le K,\ 0 \le h \le H_k\}$, and 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 :

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

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.

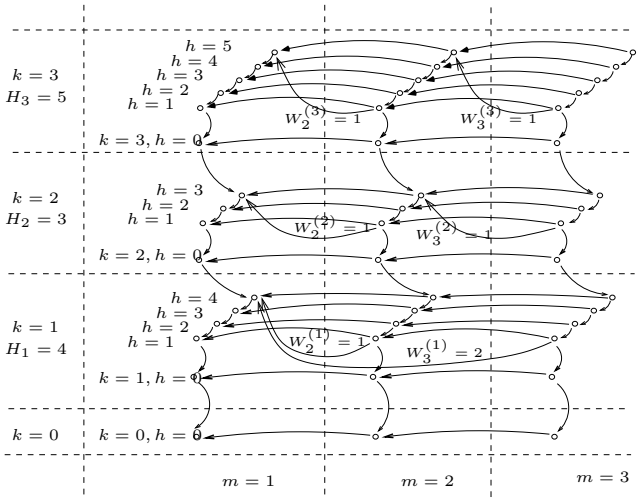

Fig. 3. Random Graph to represent a tandem of TCP connections

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 $k-1$, with the same packets) 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 the 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).

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

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

- $\texttt{Wei}_{((m,k,h)\to(m',k',h')} = \max_{\pi:(m,k,h)\to(m',k',h')} \texttt{Wei}(\pi)$;
- $\texttt{Wei}_{(m,k)\to(m',k')} = \texttt{Wei}_{(m,k,0)\to(m',k',0)}$.

III. Scalability of Throughput

In this section, we first study the throughput in a chain topology of the overlay network. We then investigate the throughput of an arbitrary tree topology. Some experimental results are then presented at the end of this section.

A. Scalability of the Chain Topology

We 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 local maximum TCP throughput of overlay edges, where by local maximum TCP throughput we mean the throughput achieved by the TCP session whose sender is not constrained by any upstream node. This result is not so surprising but it requires a proof.

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 is denoted $D^{\infty}_{m,k}$ 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 at 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 $\Theta^{\infty}_{1,K}$. Theorem 1 below shows that under very mild assumptions, the throughput limit exists.

Assumption 1: The sequences of aggregated service times, $\{(s_m^{(k,h)})_{1\le k\le K, 1\le h\le H_k}\}_m$, are jointly stationary and ergodic.

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 i.i.d., as assumed in [27].

Theorem 1: Under assumption 1, for all $1 \le k \le K$, $\lim_{m\to\infty} \frac{m}{D^{\lambda}_{m,k}}$ *converges almost surely to a constant* $\Theta^{\lambda}_{1,k}$.

Proof: We introduce for any $a \ge 1$ and $b \ge 0$:

$$\xi_{a,a+b}^{(k,H_k)} = \max\{\mathtt{Wei}(\pi) \mid \pi : (a+b,k,H_k) \to \ldots \to (a,0,0)\}.$$

For the system taken in steady state, this function of the interval $[a, a+b]$ is subadditive (this follows from the writing of this values as path in the graph), and it forms an ergodic process. Hence by Kingman's theorem, we have the above limit for this case. The system that we consider is coupling with the steady state in a finite number of packets, as seen in II-C.2, proving the above limit. All details are in [3]. □

Of particular interest is the local maximum throughput of TCP connections at the overlay edges. Let θ_k be the local maximum 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 $\Theta^{\infty}_{1,k}$ when the overlay edges $1, \cdots, k-1$ all have zero aggregate service times on their routers.

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

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

If $\lambda < \theta_1$, then the queueing network composed of the source and the routers of the overlay edge 1 is 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 converges with coupling to a stationary and ergodic point process (see e.g. [5]). Therefore, the throughput of this overlay is λ too. This completes the proof of the case $k = 1$.

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

As a direct corollary of Lemma 1, we have

Theorem 2: Under Assumption 1, for all $1 \le k \le K$, $\Theta^{\lambda}_{1,k} = \min(\lambda, \theta_1, \cdots, \theta_k)$, *i.e., the throughput of the first* k *nodes is the minimum of the arrival intensity and of the local maximum 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 (as it is assumed here) the local maximum 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 maximum throughputs.

B. 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, the throughput of the group communication is still defined as the minimum throughput observed at the end-systems: $\Theta^{\lambda}_{1,K} \equiv \lim_{m\to\infty} \min_{1\le k\le K} \frac{m}{D^{\lambda}_{m,k}} = \min_{1\le k\le K} \lim_{m\to\infty} \frac{m}{D^{\lambda}_{m,k}}$.

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.

Then, 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* $\Theta^{\infty}_{1,K} = \min(\theta_1, \cdots, \theta_K)$.

A few comments on Assumption 2 are necessary. The main restrictive hypothesis of 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's behavior. Hence Assumption 2 is primarily an assumption on the access links.

Thus, in contrast to the results of [9] 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 [9] 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 is 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.

C. Tree Topology with Possibly Congested Access Links

Assumption 2 allowed the reduction of overlay trees to tandem of overlays 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. [9]).

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.

Assumption 3: Locality assumption: the non-reference transfers originating from end-system 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 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 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 node k

- should probably only affect the aggregated service times of the very first hops of overlay edge k rather than all;
- can at most multiply the aggregated service times by M; indeed the resulting increase of cross traffic for packet m on hop h is at most $(M-1)s_m^{(k,h)}$ where $s_m^{(k,h)}$ is the size of the m-th packet of the reference flow divided by the speed of link h of overlay k. We have then $\overline{s}_m^{(k,h)} \leq s_m^{(k,h)} + (M-1)s_m^{(k,h)} = M s_m^{(k,h)}$.

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. [17]), it is then easy to get a similar result via bounding techniques, though with different constants, at least whenever all RTTs are bounded.

D. Experimental Results

Results of our measurements of throughput and buffer utilization are shown in Figure 4. 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). We repeated measurements 10 times, and took average, minimum and maximum of measured parameters.

The second column shows the local throughput of the incoming link in kilobytes per second, measured shortly after or before the multicast diffusion. We measure local throughput by sending packets on all downstream links at the maximum rate, without waiting for incoming transmission. 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 §III-C.

The last two columns show throughput and buffer utilization measurements, as observed during the global overlay multicast. In this experiment, buffer size was not restricted. 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. 20,000 blocks were sent. 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.

Node	Link Throughput (KB/s)			Tree 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

Fig. 4. Throughput and Buffer Utilization in an experimental Overlay Tree.

One immediately observes from this table that the assertion of Theorem 3 is valid, namely, the group throughput is equal to the minimum of the local maximum 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 TCP throughputs are quite heterogeneous.

IV. Scalable Buffers via Rate Control

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

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

A. 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 §III-A), then there exists at least one station $1 \le k \le K$, for which the sojourn time of packet m converges to infinity in probability when m goes to infinity.

Proof: The result reduces to the study of an overlay edge with reference throughput $\theta^{\infty}_k \le \lambda$. We have to distinguish between the case $\lambda > \theta^{\infty}_k$ and the critical case $\lambda = \theta^{\infty}_k$. For the case $\lambda > \theta^{\infty}_k$, the result follows from the ergodic theorem and one can actually show using techniques similar to those of Chapters 7 of [4] that in this case, the sojourn time of packet m converges to infinity almost surely. The proof in the critical case is based on the central limit theorem. □

B. 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 is denoted by $\tau_m = T_{m+1} - T_m$, form a stationary and ergodic sequence. Then under the stationary ergodic 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 $n \le m$, where n ranges down to $-\infty$. It is easy to check that this variable, which is denoted by $\tilde{R}_{m,k}$, can also be represented using our longest path approach via

$$\tilde{R}_{m,k} = \sup_{n\le m} \{ \mathtt{Wei}_{(m,k+1)\to(n,1)} - \sum_{i=n}^{m-1} \tau_i \}. \qquad (5)$$

For more on this type of representations, see [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 $m' \le m$) and when departing from an empty system which is given by: $R^k_{[m',m]} = \sup_{m'\le n\le m} \{ \mathtt{Wei}_{(m,k+1)\to(n,1)} - \sum_{i=n}^{m-1} \tau_i \}$. It is clear from this representations that $R^k_{[m',m]} \le \tilde{R}_{m,k}$ 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 $\tilde{V}_{m,k} = \tilde{R}_{m,k} - \tilde{R}_{m,k-1}$ with the convention $\tilde{R}_{m,0} = 0$.

In what follows, both in the simulation results and the mathematical derivations, the stochastic assumptions is that the packet inter-emission times at the source are i.i.d. and independent of the aggregated service times. The aggregated service time is also 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 denote by θ the local maximum throughput of an overlay edge.
- The non homogeneous case where the laws of the aggregated service times are assumed to be all 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, $s^{(k,h)}_m \le_{\mathtt{st}} \bar{s}$. 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 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 with same law $\bar{s}$ (except for $k \ge 1$ and $h = 0$ where $s^{(k,0)}_m = 0$). Here θ denotes the local maximum 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. 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 tandem of GI/GI/1 queues to infinite tandem and infinite trees of TCP connections over edges composed themselves of several routers. We start with simulation results and back them by mathematical justifications.

C. Simulation Results

All the simulation results of the paper are based on a direct exploitation of the evolution equations of §II-C.2. Only the homogeneous case is considered.

Figure 5 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 $\lambda < \theta$.

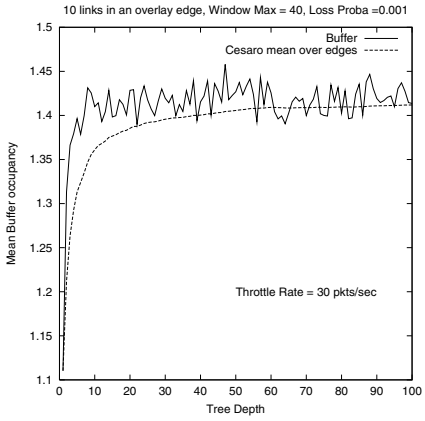

Fig. 5. Convergence of the mean buffer occupancy at infinity.

As one can check, the mean 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 6 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.

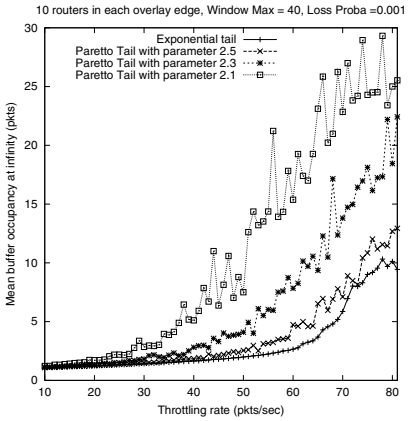

Fig. 6. The mean buffer occupancy at infinity as a function of the throttling rate for different laws of service time.

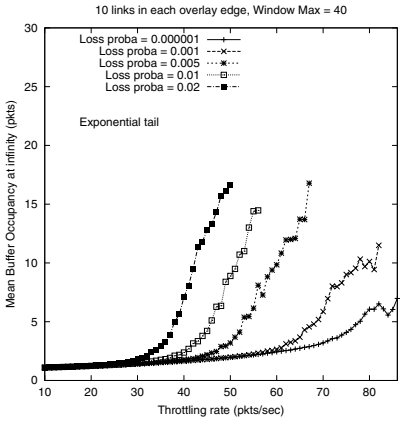

Fig. 7. The mean buffer occupancy at infinity as a function of the throttling rate for various packet marking probabilities.

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 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 7 studies the sensitivity of the $\overline{b}$ function w.r.t. the packet marking probability.

D. 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], [20] and [21]. The complete proof of the results can be found in [3].

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

$$\int_0^{+\infty} P(\bar{s} \geq u)^{1/2} du < \infty \ . \tag{6}$$

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 nondecreasing.
This function is called the hydrodynamic limit of the saturated system. The proof of this result is based on subadditivity and on the notion of *greedy lattice animal* (see the references in [20]). All details can be found in our technical report [3].

We are now in a position to state the main mathematical result, backing of 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

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

where the last convergence takes place both a.s. and in expectation. In addition, $\overline{d}$ is given by the following formula:

$$\overline{d} = \sup_{x>0} \left(\gamma(x) - \frac{1}{\lambda} x \right). \tag{7}$$

The proof is similar to those used for analogue results in [1], [20] and [21], and can be found in detail for this case in the technical report [3]. 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 $\overline{d}$.

The concavity of γ and the fact that γ' tends to θ hold in the case $W_m^k \equiv 1$ (see [1], [14]). We conjecture that this holds true for the more general setting with varying window considered here. Figure 8 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 9 plots two evaluations of $\overline{d}$ as a function of the throttling rate λ : The first one gives $\overline{d}$ defined analytically via (7) whereas the second one evaluates $\overline{d}$ by simulation as the average stationary sojourn time at infinity. The match is very good, as long as the throttling rate is not taken too close to θ.

The results of Theorem 6 extend to latency. We have :

$$\lim_{k\to\infty} \tfrac{1}{k} \textstyle\sum_{l=1}^{k} E[\widetilde{V}_{m,l}] = \overline{d}. \tag{8}$$

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 (8), 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 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.

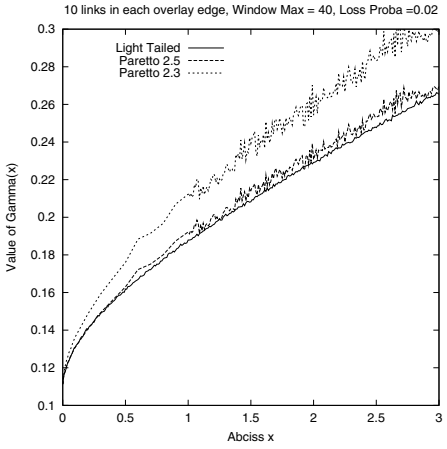

Fig. 8. An example of hydrodynamic function for the saturated system.

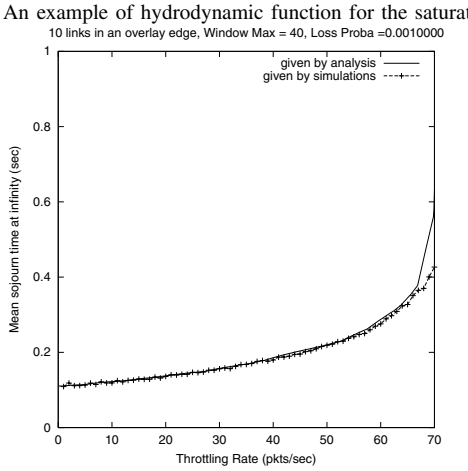

Fig. 9. Mean sojourn time at infinity by two different methods.

E. Experimental Results

In Figure 10 we show the effect of transmission rate control at the source node on buffer utilization. This experiment is identical to the overlay multicast experiment, described in § III, except that we have introduced a 10-millisecond delay between sending individual 100-byte blocks at the source node. This corresponds to fixed transmission rate of approximately 9.75 kilobytes per second.

This experiment is performed on the same configurations, as in Figure 4. Figures for local maximal throughputs are repeated from this table. To collect the measurements, we ran unsynchronized transfers, overlay multicast and overlay multicast with transfer rate control in sequence, one experiment after another without delays, until 10 measurements in each

experiment were taken. An 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
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

Fig. 10. Throughput and Buffer Utilization in an experimental Overlay Tree with controlled rate at the source.

It is clear from this table 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 in spite of the fact that the local TCP throughputs are quite heterogeneous.

V. 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.

A. 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 maximum local 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 throughput 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 a bottleneck 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.

1) Model I: Access Link not the Bottleneck: Under the assumption that the 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 marking probability p_{ij} on the edge using the square root formula for persistent flows (see [22]). As described above, the construction of overlay tree consists in 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 throughput 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.

2) Model II: Accounting for Bottleneck at Access Link: 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 is guaranteed to achieve at least $1/2$ of the optimal throughput, if either $\min c_i / \max c_i \geq 1/2$ or download bandwidth is at most twice upload bandwidth for each node.

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 cannot 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 $\lfloor c_i/\theta \rfloor$ outgoing 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 [13] for finding a spanning tree of minimum degree with additive error of at most one.

Now we describe our generalization of approximation algorithm for minimum degree spanning tree [13]. 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 for each node, 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: $d_i = \lfloor c_i/\tilde{\theta} \rfloor + 1$. If for one of the nodes i, the degree limit d_i is 0, the algorithm must report failure, since node i cannot 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 the cardinality of B by performing a 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. An improvement may also involve series of edge exchanges, which do not modify the degree of any nodes with violation $k-1$, and decrease the degree of one of the nodes with violation k.

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. The details of proof of the correctness and the optimality result are described in [3].

B. 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 maximum TCP throughput, the edges only need to measure the RTT and packet marking 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.

VI. Conclusions

We have presented a mathematical 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 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 IV, 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 [15]. 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 maximum throughput of the TCP connections between end systems. Such a consideration seems to be neglected in protocols and algorithms proposed in the literature for the group communication using overlays which primarily focus on the network distances 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 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). In this approach, no rate control may be needed at the source. However the throughput may degrade, and the structure of the overlay tree may have a direct impact on the throughput. This case will be the subject of a companion paper.

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. Bremaud, *Elements of Queuing Theory*, Springer Verlag (2nd edition 2002).

[3] F. Baccelli, A. Chaintreau, Z. Liu, A. Riabov, S. Sambit *Scalability of Reliable Group Communication Using Overlays*, INRIA Research Report 4895 (July 2003), available at : `http://www.inria.fr/rrrt/rr-4895.html`.

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

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

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

[7] 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

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

[9] 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.

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

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

[13] 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).

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

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

[16] 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.

[17] 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).

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

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

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

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

[22] 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.

[23] 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.

[24] 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.

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

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

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

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