A Congestion Control Model for Multicast Overlay Networks and its Performance

Guillaume Urvoy-Keller
Institut Eurecom
2229, route des Crêtes
06904 Sophia-Antipolis, France
urvoy@eurecom.fr

Ernst W. Biersack
Institut Eurecom
2229, route des Crêtes
06904 Sophia-Antipolis, France
erbi@eurecom.fr

ABSTRACT

We propose a new TCP-friendly Multicast Congestion Control (MCC) model for overlay networks and study its performance in terms of throughput. We assume that the multicast distribution tree is built at the application level and that each overlay network node has a limited buffer for packet storage. The overlay MCC decomposes the feedback loop between receivers and sender into a chain of control loops, one for each branch of the multicast distribution tree.

To calculate the throughput of a multicast session, we use a linear recurrence method based on the max-plus algebra. Preliminary numerical results demonstrate that the throughput performance of overlay MCC is almost independent of the number of receivers, while for the end-to-end MCC, the throughput decreases logarithmically with the number of receivers. This result is a clear indication that overlay networks for multicast distribution offer a major performance advantage over native end-to-end multicast distribution.

Keywords

Congestion Control, Multicast, Overlay Network, TCP

1. INTRODUCTION

Research in multicast congestion control has focused mainly on layered transmission schemes where each receiver chooses the number of layers to subscribe to [6, 2] and on the issue of TCP friendliness [8, 10]. Source-based single-rate multicast congestion control [9, 7] and in particular its performance has received less attention [4, 3]. Recently Chaintreau [3] has proven that there is a serious performance and scalability problem with source-based single-rate end-to-end multicast congestion control: Even for the case when the bottleneck bandwidths and round-trip times to the different receivers are, on average, the same and the transmission delays to the different receivers are only subject to light variations, the *throughput of a multicast session decreases like the inverse of the logarithm of the number of multicast receivers.*

Given the limited scalability of end-to-end (E2E) MCC, overlay networks for multicast data distribution offer an interesting alternative. Overlay multicast distribution operates as follows: Inside the network a certain number of overlay nodes are deployed. The data between adjacent overlay nodes is transmitted in a store-and-forward like manner using TCP: An overlay node receives data, buffers the data, and forwards the data on its multiple outgoing links towards the neighboring downstream overlay nodes. A set of overlay nodes, when organized in a tree like fashion, provides a multicast forwarding service that uses only the underlying unicast routing capabilities of the Internet (support of native multicast routing is not required). Such an overlay distribution network greatly simplifies the multicast congestion control problem: Instead of performing an end-to-end congestion control between the source and the receivers, overlay MCC uses unicast congestion control between adjacent overlay nodes. In fact, since the overlay nodes use TCP to transmit the data the overlay MCC will be TCP-friendly.

It seems intuitively evident that overlay MCC should result in a higher throughput since it decouples the rate of transmission over the different branches of the distribution tree. It is the objective of this paper to quantify the performance improvement of overlay MCC and to evaluate the additional storage requirement in an overlay node.

The article is organized as follows: In section 2, we represent the E2E MCC model used by Chaintreau [3] together with our Overlay MCC model. In section 3, we introduce the max-plus algebra, which was used in [3] to model E2E MCC and which provides the tool to analyze the Overlay MCC. We use a linear recurrence method based on the max-plus algebra to calculate the throughput of both the E2E and Overlay MCC model.

In section 4, we present preliminary numerical results that demonstrate that the Overlay MCC model does not significantly suffer a throughput degradation for increasing number of receivers. While these results are obtained for moderate size networks, they are noteworthy because they correspond to situations where the E2E MCC model is already affected by the number of receivers [3]. Furthermore, as opposed to [3], we consider a heavy-tailed service law rather than a light-tailed (exponential) one. This choice enables us to infer the behavior of the Overlay MCC model for a larger number of receivers. The results are promising since in our scenarios, the Overlay MCC model shows no signif-

icant throughput degradation for increasing number of receivers.

Finally, we conclude in section 5.

2. E2E AND OVERLAY MODELS

We will first present the E2E MCC model. Our objective is not to model the details of TCP's congestion control (at least not in a first stage) but to obtain meaningful results for a fixed-size window control that models a TCP connection in steady state. We will then explain the Overlay MCC model.

As a network topology, we assume a homogeneous tree where all receivers are at the leaf nodes and have same distance from the source. This assumption enables us to use a simple model where single window size is used for all receivers when E2E MCC is considered, which agrees with the Golestani's conclusion that the window size should be proportional to the distance from the source [4]. For sake of simplicity, we assume an unlimited bandwidth on the return path; delays of acknowledgments are negligible.

2.1 E2E MCC

The E2E MCC model is depicted in Figure 1. All nodes are multicast capable legacy nodes (routers). There is a global window parameter W^e and packet n is sent by the source only if copies of packet $n - W^e$ have been received by all the receivers (equivalently, this means that all the acknowledgments for packet $n - W^e$ have been received by the source since we assume an unlimited bandwidth on the reverse path).

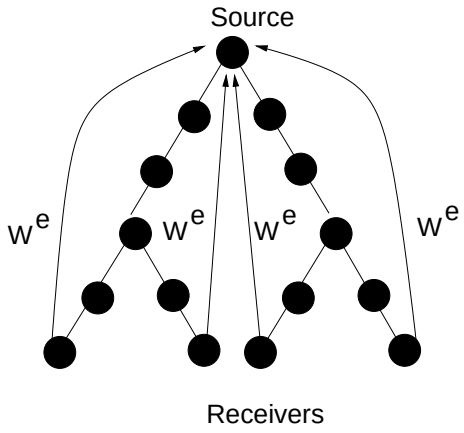

Figure 1: E2E MCC

2.2 Overlay MCC

The details of the overlay MCC model are presented in figure 2. Overlay nodes are placed whenever the distribution tree forks off (i.e. the source is also an overlay node). All the overlay nodes with exception of the source are *inside* the network. The functionality of the overlay node can be realized using a separate machine installed adjunct to the router to which it is connected for receiving and sending data packets. Intermediate nodes between overlay nodes are legacy nodes (routers), which can be ordinary routers that are not multicast capable. The unicast connection between two adjacent overlay nodes uses a fixed-size window control with parameter W_c^o. This window control is a *congestion* control mechanism as it will slow down the sending overlay node when the network path between the two overlay nodes gets congested.

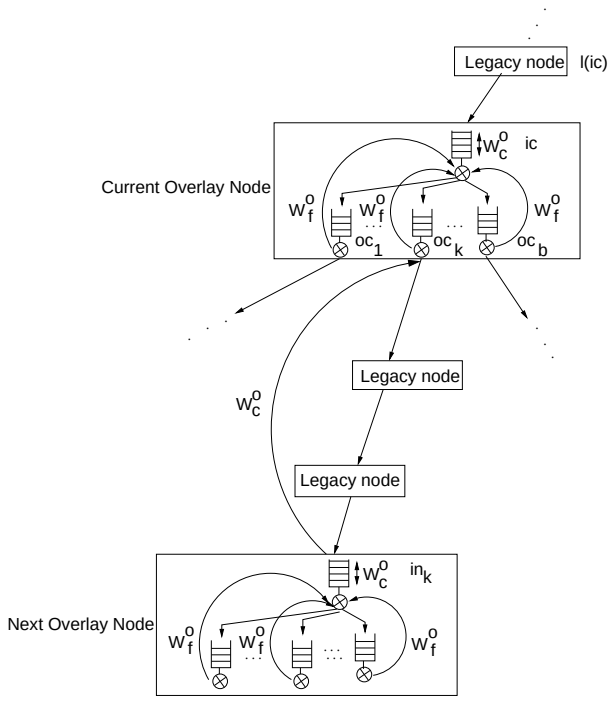

Figure 2: Overlay MCC model

To decouple the outgoing branches of an overlay node, there is a buffer (virtual or real) per output. However, even if the objective is to break the *tight* coupling induced by the E2E MCC model, we want to preserve a loose coupling in the sense that if a branch or a part of the distribution tree gets too congested for some time (i.e. falls too much back in forwarding packets), the source should become aware of it. A backpressure mechanism is thus mandatory. For this purpose, we apply a fixed window (W_f^o) control between the output buffers of an overlay node and its input buffer: For packet n to be copied into the output buffers, packet $n - W_f^o$ must have been emitted from all the outputs. Such a window control is a *flow* control mechanism since the input buffer adapts its forwarding rate to the consumption rates at the output buffers. The flow control mechanism between the output buffers and the input buffer does yet not ensure that the backpressure will reach the source: Consider two overlay nodes where the first (upstream) overlay node transmits packets to the second one and the path between the two overlay nodes is not congested. On the other hand, one outgoing branch of the second overlay node is highly congested. In this case the input buffer of the second overlay node will build up rapidly as it will stop forwarding packets as soon as the output buffer corresponding to the congested branch holds W_f^o packets.

To ensure a backpressure up to the source, the flow and congestion control mechanism must be coupled. To this purpose, the size of the input buffer of an overlay node is limited to W_c^o packets and the overlay node only sends back an acknowledgment when the arriving packet can be copied into all output buffers. If a packet can not be copied into

all output buffers, the input buffer starts to fill up and no acknowledgment will be sent back to the upstream overlay node. Since up to W_c^o packets (the maximum number of outstanding packets) may be in transit between two overlay nodes, we must be able to store W_c^o packets in the input buffer to avoid packet loss.

We have now introduced a coupling that assures that the backpressure can propagate all the way to the source. The parameter W_f^o controls the level of coupling (speed of backpressure): the larger the value of W_f^o, the weaker the coupling. When W_f^o is infinite for all overlay nodes, there is no backpressure and the receivers are completely decoupled.

A complete view of an overlay distribution tree is presented in figure 3.

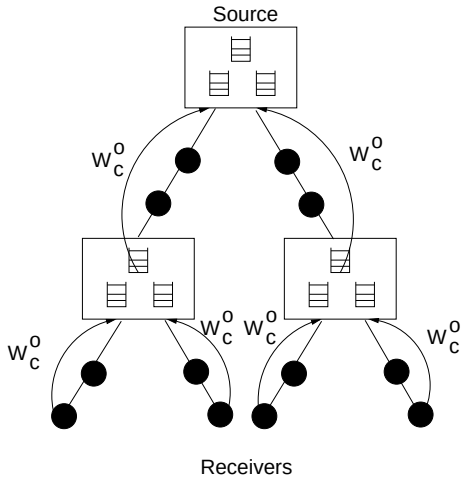

Figure 3: Overlay MCC model

2.3 Implementation of Overlay MCC Model

While the objective of the paper is to demonstrate that the Overlay MCC model outperforms the E2E MCC model in terms of performance, we also want to point out its simplicity of implementation. The basic operations of the overlay application are described in figure 4. Upon arrival of a new packet, the application dequeues the packet only if there is at least one free space in each outgoing buffer. Otherwise, the application stalls, which results in the TCP incoming application to decrease its advertised window, ensuring propagation of backpressure.

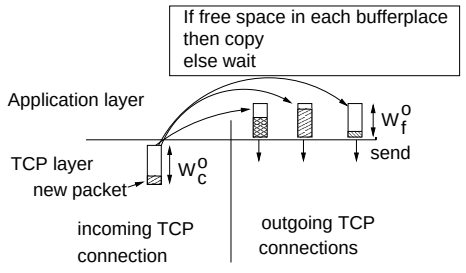

Figure 4: Implementation of the Overlay MCC model

3. MAX-PLUS ANALYSIS

We use a set of max-plus linear equations to model the behavior of the overlay MCC. The max operation denoted by $\vee$ is used instead of the usual addition and the addition denoted by $+$ is used instead of the usual multiplication. Both operations are defined inside $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$ with the convention of $-\infty + c = -\infty, \forall c \in \mathbb{R}_{max}$. It is straightforward to show that $(\mathbb{R}_{max}, \vee, +)$ is a commutative semi-ring with the following mathematical properties: commutativity, distributivity, associativity and identity elements. Here $\varepsilon = -\infty$ is the identity element of the $\vee$ operation and the $e = 0$ is the identity element of $+$ operation since $-\infty \vee c = c$ and $0 + c = c,\ \forall c \in \mathbb{R}_{max}$.

Let us introduce the following notation:

- $s_m^i, i \in \mathbb{N}, m \in \mathbb{N}$: service time of packet m at i where i, depending on the context, is a node or the (input or output) buffer of a given node.
- $x_m^i, i \in \mathbb{N}, m \in \mathbb{N}$: time instant when packet m leaves i where i, depending on the context, is a node or the (input or output) buffer of a given node.
- b: number of output buffers at each overlay node.
- ic: input buffer of current overlay node.
- $l(ic)$: legacy node upstream to ic.
- oc_k: k-th output buffer of current overlay node.
- in_k: input buffer of next overlay node corresponding to oc_k.

In our model we assume, as did Chaintreau [3], that the buffers in each router are infinite. In other words, there is no packet loss in the network. In each node, scheduling is FIFO and the cross traffic seen by the packets of our MC session is modeled by *independent and random* service times s_m^i (which do not include queuing times).

The time when a packet leaves a node in the overlay distribution tree is computed as follows:

- The k-th output buffer of a current overlay node sends packet m as soon as it has processed packet $m-1$, provided that it has received packet m from the input buffer and the window control allows packet m to be sent. So we have:

$$x_m^{oc_k} = (x_{m-1}^{oc_k} \vee x_m^{ic} \vee x_{m-W_c^o}^{in_k}) + s_m^{oc_k}$$

- The input buffer ic of the current overlay node sends packet m to each output buffer as soon as it has processed packet $m-1$, provided that the upstream legacy node $l(ic)$ has forwarded packet m and there is at least one free space in each output buffer. So we have:

$$x_m^{ic} = (x_{m-1}^{ic} \vee x_m^{l(ic)} \vee x_{m-W_f^o}^{oc_1} \vee \ldots \vee x_{m-W_f^o}^{oc_b})$$

- A legacy node i sends packet m as soon as it has processed packet $m-1$ and the previous node $i-1$ has forwarded packet m. So we have:

$$x_m^i = (x_m^{i-1} \vee x_{m-1}^i) + s_m^i$$

With the specified properties of $(\mathbb{R}_{max}, \vee, +)$, the max-plus algebra can be extended to matrices where the multiplication of two matrices is defined by the rule:

$$(\mathbb{AB})_{i,j} = \max_k (\mathbb{A}_{i,k} + \mathbb{B}_{k,j})$$

We define two special matrices: $\mathcal{E}$ is the matrix filled with ε everywhere and $\mathbb{I}$ is the identity matrix filled with e on the diagonal and ε everywhere. Given these definitions there hold: $\mathcal{E}\mathbb{C} = \mathcal{E}$ and $\mathbb{I}\mathbb{C} = \mathbb{C}$, $\forall \mathbb{C}$.

Since the scope of the max-plus algebra to can be extended to matrices, we can calculate the throughput of the overlay MCC model by the max-plus linear recurrence. Let us adopt the following notations:

- N: total number of nodes in a given topology.
- W: maximum of the window sizes W_c^o and W_f^o (in terms of packets).
- $X_m = (x_m^1, \ldots, x_m^N)$.
- Y_m block vector with entries $X_m, X_{m-1}, \ldots X_{m-W+1}$.

Then,

$$\begin{cases} Y_0 = (\underbrace{e, \ldots, e}_{N times}, \underbrace{\varepsilon, \ldots, \varepsilon}_{N(W-1) times}) \\ Y_m = \mathbb{P}_m Y_{m-1} \text{ for } m > 0 \end{cases} \quad (1)$$

where $\mathbb{P}_m$ is a block matrix defined as follows:

$$\mathbb{P}_m = \begin{pmatrix} \mathbb{S}_m & \mathcal{E} & \ldots & \mathcal{E} & \mathbb{W}_m \\ \mathbb{I} & \mathcal{E} & \ldots & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & \mathbb{I} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & \ldots & \mathcal{E} & \mathbb{I} & \mathcal{E} \end{pmatrix}$$

Each block in $\mathbb{P}_m$ is a square matrix of dimension $N \times N$, where $\mathbb{S}_m$ and $\mathbb{W}_m$ account for the service times and the window flow control mechanism respectively. An analysis similar to [3] may be carried out for the case of the Overlay MCC model described above and the generic terms of $\mathbb{S}_m$ and $\mathbb{W}_m$ can be calculated as follows:

- $(\mathbb{S}_m)_{ij}$: sum of services between nodes j and i packet m (including the nodes i and j) if this path exists and 0 otherwise. Then:

$$(\mathbb{S}_m)_{ij} = \max\Big(\sum_{k \in \mathbb{R}_{ji}} s_m^k, \varepsilon\Big) \quad (2)$$

  where $\mathbb{R}_{ji}$ represents the set of nodes between nodes j and i. If no path exists between nodes i and j, $\mathbb{R}_{ji}$ is an empty set and $(\mathbb{S}_m)_{ij}$ is equal to ε.

- $(\mathbb{W}_m)_{ij}$: If j is a node whose outgoing traffic rate controls (through a window mechanism) the outgoing traffic of another node j_0 in the distribution tree, then we will call j as a "controlling node" and j_0 as its "ancestor" (different from its predecessor which is its parent node). For example, in the case of the E2E MCC model, only the leaves of the tree (receivers) are controlling nodes and have a common ancestor, which is the root node. Then $(\mathbb{W}_m)_{ij}$ is the maximum over all paths between the ancestor j_0 and i of the sum of service times for packet m (including the nodes i and j). Then

$$(\mathbb{W}_m)_{ij} = \max\Big(\sum_{k \in \mathbb{R}_{j_0 i}} s_m^k, \varepsilon\Big), \quad (3)$$

  where j_0 is the ancestor of node j. If j is not a controlling node or there exists no path between nodes j_0 and i, $(\mathbb{W}_m)_{ij}$ is equal to ε.

The following theorem [1] allows us very easily to use the max-plus formalism to compute the throughput obtained:

THEOREM 1. *With i.i.d service times, the matrices $(\mathbb{P}_m)_{m \geq 0}$ are also i.i.d and this implies the existence of $\lim_{m\to\infty} \frac{\|Y_m\|}{m} = \gamma$ (called the Lyapunov exponent) with probability 1. γ corresponds to the inverse of the asymptotic throughput of the connection.*

Here $\|\cdot\|$ represents the matrix norm defined as $\|\mathbb{A}\| = \max_{i,j}(\mathbb{A})_{i,j}$. The assumption of identical service times is natural since the considered multicast session is assumed to be in steady state with fixed window sizes. However, the independence of the service times is not so clear. Here, the service times are considered to be independent for mathematical tractability.

Thanks to the above theorem, a simple linear recurrence based on Equation (1) may be used to compute the throughput of the overlay MCC model. By recursive computation of Y_m we can obtain the Lyapunov exponent (the inverse of the throughput) $\gamma = \lim_{m\to\infty} \frac{\|Y_m\|}{m}$ for large enough values of m. There is a trade-off between the accuracy of the estimation and the time required to estimate the throughput. We choose $m = 2000$ in our calculations, which was sufficient to have smooth curves and throughput values with small confidence intervals (less than 5% with a 95% probability).

4. NUMERICAL RESULTS

We choose a homogeneous binary tree topology of depth $P = 5$. In this topology, there are five overlay nodes (including the source itself) on the path from the source to any receiver. Between two overlay nodes there are two legacy nodes. We examine the influence of the following network parameters on the throughput:

- Number of active receivers,
- Window sizes W^e, W_c^o, and W_f^o,

Congestion due to the cross traffic is modeled as a random service delay at each node.

Sionce the simulations are computationally very expensive, we focus on networks of moderate size with up to 32 clients. Note, however, that the logarithmic decrease of the throughput observed in [3] is already noticeable for less than 32 clients. We consider a heavy-tail service time distribution and use the Bounded Pareto distribution with mean $\mu = 1$ and coefficient of variation equal to 100 ($a = 0.3333$, $p = 3 \times 10^7$, $\alpha = 1.5$). The probability density function of the bounded Pareto distribution is given by:

$$f(x) = \frac{a^\alpha}{1 - (a/p)^\alpha} \alpha x^{-(1+\alpha)}, \text{ where } a \leq x \leq p.$$

The reason of using a heavy-tail service distribution is twofold. First, it is well-known that modeling the exponential

service time distribution is not appropriate for the Internet traffic and highly varying laws (e.g. Bounded Pareto), are more appropriate [5]. Second, since it is numerically difficult to obtain results for a larger number of receivers (say 1000 to 10 000), using an heavy-tail distribution allows to model (at least to a certain extend) the behavior of the Overlay MCC model with a larger number of receivers. Indeed, as the group becomes larger, the probability that different subtrees of the distribution network experience different congestion conditions becomes higher. Using an heavy tail law results in clients experiencing very different conditions on their paths from the source and thus allow to stress the Overlay MCC model, creating conditions similar to the case of a larger network.

We first obtain numerical results for E2E MCC model. We see in Figure 5 that the throughput for E2E MCC model decreases as the number of receivers increases. In fact, Chaintreau [3] proved first that for exponentially distributed service times, the decrease is proportional to $\log(r)$ where r is the number of receivers.

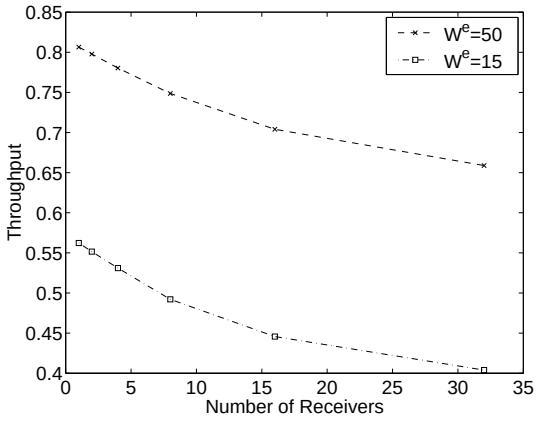

Figure 5: E2E MCC model for heavy tailed service times (P=5).

To compare the overlay and the E2E MCC model, we must choose meaningful window sizes. We have three parameters W^e, W_f^o, and W_c^o: W^e controls the number of outstanding packets in the network, W_f^o controls the level of coupling between branches of the MC tree, and W_c^o controls the number of outstanding packets in a branch between two overlay nodes. We want to compare the throughput performance of both MCC models for the same number of outstanding packets on each path from the source to a given receiver. If we denote the total depth of the tree as P, then the following equation must hold:

$$W^e = P \cdot W_c^o \tag{4}$$

For the Overlay MCC model, we will examine two cases, depending on the value of W_f^o being finite or infinite. For the finite case, we present results for three different values of W_f^o, namely when $W_f^o = 1$, $W_f^o = W_c^o$ and $W_f^o = 10 \times W_c^o$. Since there are three nodes (two legacy nodes and the overlay node itself) between two adjacent overlay node, we choose the minimal congestion window to be of size $W_c^o = 3$, since for $W_c^o < 3$ one node is always idle.

The throughput behavior of the Overlay MCC model can be seen in Figure 6 for different values of W_c^o when the output buffer sizes of all overlay nodes are assumed to be infinite. The throughput stays almost constant and is independent of the number of receivers. For $W_f^o = \infty$, a complete decoupling is achieved and there is no backpressure to the source. For instance, if there occurs a congestion in a subtree, the output buffer of the overlay node in charge of that subtree will fill up infinitely (we assume infinite buffer) and the source will never become aware of. For this reason, we consider finite values of W_f^o in the rest of this paper.

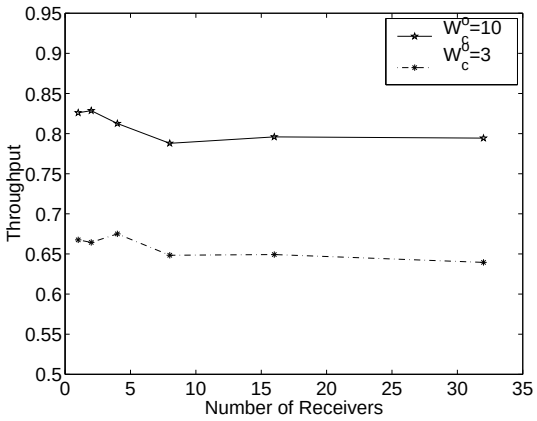

Figure 6: Overlay MCC model for heavy tailed service times ($P = 5$, $W_f^o = \infty$).

Let us define the *performance increase* as the ratio of the throughput of the Overlay MCC model to the throughput of the E2E MCC model. We assume that the value of W^e is calculated as in Equation (4).

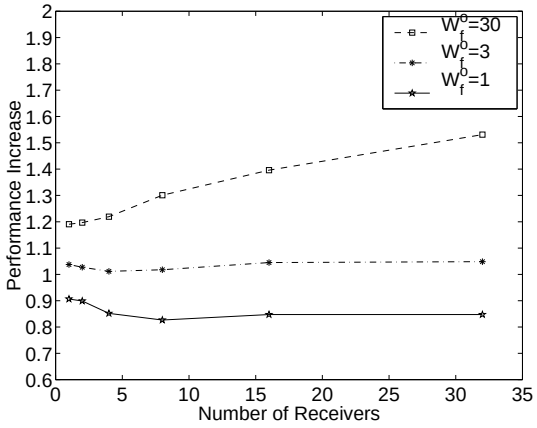

Figure 7: Performance comparison for heavy tailed service times ($P = 5$, $W_c^o = 3$)

We see in figure 7 the impact of the value of W_f^o on the performance of the overlay MCC model. For $W_f^o = W_c^o$, the overlay MCC model gives a throughput performance that is similar to the one for the E2E MCC model. For $W_f^o = 1$, the coupling of the different branches is the tightest possible one and the throughput of overlay MCC model is below

the throughput for the E2E MCC model. On the other hand, setting $W_f^o = 10 \times W_c^o$ allows to achieve a significant performance increase. The performance for $W_f^o = 10 \times W_c^o$ is similar to the one for $W_f^o = \infty$. This result is promising, as it demonstrates that overlay MCC is practically feasible since a *finite* input buffer in each overlay node is sufficient to make the throughput performance almost independent of the number of receivers.

However, the decoupling of the different branches of the overlay MCC is obtained at the expense of a larger time required for the backpressure to reach the source. Let us denote by **depth** h, $1 \leq h < P$, of an overlay node its distance from the source. Then, for the overlay MCC model, a maximum of $10 \times W_c^o \times h$ packets needs to be sent before the source becomes aware of the congestion occurring at depth of h whereas this number is $W^e = P \times W_c^o$ in the E2E MCC model. The time for backpressure to reach the source is highest in the overlay MCC model when a congestion occurs next to the receiver. For this worst case, 10 times more packets need to be sent under the overlay MCC model until the backpressure reaches the source.

For $W_c^o = 3$ and 32 receivers, a throughput increase of about 50% is observed for the overlay MCC in case of heavy tailed service times (see Figure 7). The performance increase observed for the overlay model is mainly due to the severe throughput degradation of the E2E MCC model with increasing number of receivers. In other words, while the performance of the overlay MCC model is not significantly affected by an increase in the number of receivers, the performance for the E2E MCC model decreases drastically in case of the heavy tailed service times. For the E2E MCC model, packets stuck at the congested node will reduce the overall throughput since the source stops sending packets even to the parts of tree where there is no congestion. In the overlay MCC model, the output buffers give the necessary flexibility to tolerate transient congestion in varying parts of the distribution tree without degradation in throughput.

5. CONCLUSIONS

The throughput degradation of E2E MCC with increasing number of receivers was first shown by Chaintreau [3]. However, no counter measure was proposed to prevent such a throughput degradation. We propose a novel MCC model for overlay networks that makes the throughput performance of a multicast session independent of the number of receivers. In particular,

- we propose a TCP-friendly multicast congestion control model for overlay networks that combines a window-based congestion control between adjacent overlay nodes with a window-based flow control mechanism inside each overlay node. The flow and congestion control interact to allow backpressure to reach the sources.

- we formalize the overlay MCC model via a set of linear recurrence equations, which allows us to use theoretical results from the max-plus algebra to compute the throughput of a MC session as a function of the number of the receivers, the window and buffer size, and the service time distribution.

The preliminary results obtained show that for overlay MCC, the throughput does not significantly decrease with the number of receivers. If we compare the throughput performance of overlay MCC vs. E2E MCC, we see that for 32 receivers and properly chosen buffer sizes the overlay MCC yields a 50% improvement. The fact that for overlay MCC the throughput seems unaffected by the number of receivers is due to a high degree of decoupling of the different branches, which as been achieved, in the scenarios we considered, with *finite-size* buffers in the overlay nodes.

As future work, we intend to extend, either numerically or analytically, the results obtained here to the case of larger group sizes. The major challenge will be to find out whether the amount of buffer (or equivalently, for our scheme, the tightness of the flow control) increases as the group size increases. We may expect that if we continue to expand the size of the binary trees we have focused on, the buffer requirement might eventually increase unbounded. On the other hand, if we consider a quite moderate overlay network, say with a few tens of (overlay) nodes, with many clients connected to each end-node (in our simulations, we considered only one client per end-node), we can hope to maintain the nice properties of our scheme. We note eventually that there exists a trade-off between the level of reliability of the service and the number of clients. In our service model, we aim at providing a service that is TCP friendly on each branch of the tree and that is also fully reliable. A consequence of the latter constraint is that the overlay network must slow down in case of a long congestion in part of the distribution tree. For very large and highly scattered groups, some other solutions like hierarchy and grouping, need to be investigated to maintain a high level of reliability without sacrificing the overall bandwidth.

Acknowledgments

This research was supported by Intel Corporation.

6. REFERENCES

[1] F. Baccelli, G. Cohen, G. Olsder, and J.-P. Quadrat, *Synchronization and Linearity, an Algebra for Discrete Event Systems*, Wiley, 1992.

[2] J. Byers et al., "Improved Congestion Control for IP Multicast Using Dynamic Layers", In *Proc. NGC 2000*, November 2000.

[3] A. Chaintreau, F. Baccelli, and C. Diot, "Impact of Network Delay Variation on Multicast Sessions Performance with TCP-like Congestion Control", *To appear in IEEE Transactions on Networking*, 2002.

[4] S. J. Golestani and K. K. Sabnani, "Fundamental Observations on Multicast Congestion Control in the Internet", In *Proc. of INFOCOM'99*, pp. 990–1000, New York, USA, March 1999.

[5] W. Gong et al., "On the Tails of Web File Size Distributions", In *Proc. of 39-th Allerton Conference on Communication, Control, and Computing*, October 2001.

[6] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast", In *SIGCOMM 96*, pp. 117–130, August 1996.

[7] L. Rizzo, "pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme", In *Proc. of ACM SIGCOMM'00*, Stockholm, Sweden, August 2000.

[8] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer", In *Proc. of IEEE INFOCOM'98*, pp. 996–1003, San Francisco, CA, USA, March 1998.

[9] J. Widmer and M. Handley, "Extending Equation-based Congestion Control to Multicast Applications", In *SIGCOMM 2001*, August 2001.

[10] J. Widmer and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification", , Internet Draft, November 2001.