

# Controle de Congestionamento com Suporte a ECN em Protocolos Multicast de Taxa Única

André Detsch, Marinho Barcellos

<sup>1</sup> PIPCA - Programa de Pós-Graduação em Computação Aplicada  
Centro de Ciências Exatas e Tecnológicas - UNISINOS

**Resumo.** A maior parte do tráfego que circula na Internet hoje é transmitido usando-se TCP, ou *Transport Control Protocol*. A estabilidade da Internet depende do controle de congestionamento exercido por esse protocolo nas estações fim, bem como de mecanismos equivalentes (que seguem a política de incremento aditivo e decremento multiplicativo) empregados nos demais protocolos. Mecanismos de controle de congestionamento atuais tipicamente se baseiam na detecção de perdas de pacotes como indicativo de congestionamento. A técnica de ECN (*Explicit Congestion Notification*), em que pacotes encaminhados por roteadores são “marcados” quando congestionamento é iminente (ou já ocorre), possibilita que o transmissor reduza a taxa de envio sem depender de descartes. Entretanto, a técnica de ECN ainda não é muito explorada, em especial no que se refere a protocolos multicast.

O presente trabalho propõe um mecanismo de controle de congestionamento para protocolos multicast confiável e o estende de forma a incorporar ECN. São considerados apenas protocolos de taxa única e baseados em janela, em que o transmissor mantém uma janela de envio para cada receptor. Este mecanismo é avaliado através de simulações em cenários tipicamente utilizados na avaliação de controle de congestionamento. Resultados de experimentos são analisados, e mostram as características de justiça (*fairness*) entre fluxos de múltiplas fontes e a amigabilidade ao TCP (*TCP-friendliness*) do esquema proposto. Os resultados demonstram, adicionalmente, o ganho significativo em eficiência obtido com o uso de ECN em multicast.

**Abstract.** Most of the traffic that circulates through the Internet nowadays is transmitted using TCP, or Transport Control Protocol. The stability of the Internet depends on the congestion control performed by this protocol in the end nodes, as well as equivalent mechanisms (that follow the additive increase and multiplicative decrease policy) employed by other protocols. Current congestion control mechanisms are typically based on packet loss as an indication of congestion. The ECN technique (Explicit Congestion Notification), in which packets forwarded by routers are marked whenever congestion is about to occur (or already occurring), allows a transmitter to reduce the sending rate accordingly without relying on packet drops. However, ECN has not been fully explored, particularly regarding multicast protocols.

This paper introduces a new congestion control mechanism for reliable multicast protocols, and extends it to incorporate and take advantage of ECN. Only single-rate and window-based protocols are considered; in such protocols, the sender maintains a sending window for each receiver. The proposed mechanism is analysed through simulations in scenarios typically employed in congestion control evaluation. Results show that the mechanism provides fairness between flows from multiple sources, and is TCP-friendly. Further, results demonstrate the substantial efficiency gain obtained with the use of ECN in multicast.

**Palavras-chave:** Controle de Congestionamento, Multicast, ECN.

## 1 Introdução

A técnica de *multicasting* permite a transmissão eficiente de dados para um grupo de receptores, evitando a replicação desnecessária de pacotes na rede. Sua utilização é ampla, abrangendo aplicações como transmissão de arquivos, vídeo-conferência e sistemas distribuídos baseados em comunicação em grupo. A criação de protocolos que atendam aos

diferentes requisitos de cada aplicação ainda é um tópico de pesquisa bastante importante na área de redes de computadores (por exemplo, [12] e [23]). Apesar da larga disponibilidade de multicast em equipamentos de rede, o uso de protocolos multicast na Internet tem sido freado, entre outros fatores, pela ausência nesses protocolos de mecanismos de controle de congestionamento compatíveis com o esquema adotado pelo *Transport Control Protocol*, ou TCP. Mecanismos de controle de congestionamento fim-a-fim permitem aos nós envolvidos na transmissão adaptar a taxa de geração de tráfego de acordo com as condições da rede.

O uso de algoritmos para controle de congestionamento é fundamental para que os protocolos de comunicação possam coexistir em um ambiente de rede como a Internet. Uma vez que o TCP é o protocolo dominante atualmente ([17]), a *Internet Engineering Task Force* (IETF) define que qualquer mecanismo de controle de congestionamento deve ser amigável ao TCP; mais precisamente, “um fluxo unicast é considerado amigável ao TCP se este não reduz o *throughput* de uma conexão TCP concorrente mais de que um outro fluxo TCP o faria” ([21]).

O desenvolvimento de algoritmos de controle de congestionamento para protocolos multicast envolve uma série de aspectos não presentes no caso de transmissões unicast. Particularmente, a escalabilidade do esquema representa um grande desafio, tendo sido foco de diversos projetos de pesquisa recentes ([23], [19], [18]). Para o caso de multicast, define-se que um fluxo é amigável ao TCP se para cada par transmissor-receptor, a amigabilidade ao TCP é obedecida ([21]).

Na Internet, a detecção de situações de congestionamento por parte dos protocolos de transporte está, quase na sua totalidade, baseada na detecção de perdas de pacotes. Ou seja, o congestionamento só é remediado após o descarte de pacotes por parte dos roteadores. No caso dos roteadores *drop-tail* ([10]), estes descartes só ocorrem em virtude de transborde (*overflow*) nas filas, ou seja, os protocolos só podem detectar congestionamento em uma situação já desfavorável. Roteadores *Random Early Detection* (RED) ([7]) amenizam este problema ao efetuarem descartes aleatórios de pacotes antes de as filas ficarem completamente cheias. Entretanto, perdas continuam sendo geradas como forma de sinalizar às estações uma situação de congestionamento. Dentro deste contexto, a tecnologia de *Explicit Congestion Notification* (ECN) ([6], [16]) representa uma potencial melhoria, pois permite que as estações envolvidas na comunicação detectem um congestionamento em formação (pró-ativamente) **antes** que ocorra o descarte de pacotes. Sinais de congestionamento são gerados, explicitamente, pelos roteadores capacitados. Apesar das vantagens apresentadas por esta técnica, seu uso em protocolos de congestionamento multicast ainda não é explorado.

Este trabalho possui duas contribuições importantes: propõe um esquema de controle de congestionamento para protocolos multicast confiável de taxa única<sup>1</sup>, considerando a técnica de *polling*, e então estende o mesmo de forma a explorar ECN, com ganhos significativos em termos de desempenho.

O restante do documento está organizado como segue. A Seção 2 aborda a técnica de ECN e discute como controle de congestionamento é aplicado às diferentes categorias de protocolos multicast de taxa única. A Seção 3 propõe um algoritmo de controle de

---

<sup>1</sup>Em protocolos de taxa única, o transmissor envia dados na mesma taxa para todos os receptores.

gestionamento para protocolos multicast confiável que, similarmente ao TCP, utilizam janelas deslizantes mas restringem confirmações através de *polling*. A Seção 4 apresenta uma avaliação por simulação do esquema proposto, demonstrando sua amigabilidade ao TCP e ilustrando os ganhos obtidos através do uso de ECN em multicast. Considerações finais e trabalhos futuros aparecem na Seção 5, que encerra o artigo.

## 2 ECN e Controle de Congestionamento Multicast

Com ECN, sinais são explicitamente gerados pelos roteadores com o objetivo de indicar aos nós fim a ocorrência de congestionamento. Ao contrário da técnica de *ICMP Source Quench* ([15]), onde mensagens adicionais são geradas e enviadas para os transmissores dos fluxos de dados, a sinalização ECN utiliza os pacotes de dados já existentes para transmitir, aos receptores, sinais indicando que a rede está congestionada.

A RFC 3168 ([16]) define a incorporação da técnica ao protocolo IP, através da utilização de um campo ECN (composto de dois bits) no octeto TOS (*Type Of Service*). Este campo pode assumir valores correspondentes a três situações: (i) *not-ECT* indica que o protocolo de transporte não possui suporte a ECN; (ii) *ECN-Capable Transport* (ECT) indica que o protocolo de transporte possui suporte a ECN; e (iii) *Congestion Experienced* (CE), código que é utilizado pelos roteadores para indicar aos nós fim que existe congestionamento na rede. A RFC 3168 ([16]) define ainda a forma de implementação da técnica nos roteadores e protocolos de transporte.

Nos roteadores com suporte a ECN, a marcação de pacotes deve obedecer a uma regra simples: um pacote deve apenas ser marcado, isto é, ter seu campo ECN ajustado para CE (*Congestion Experienced*), quando este possuir a indicação de que o protocolo de transporte suporta ECN (ECT - *ECN-Capable Transport*) e apenas nas mesmas situações em que um pacote sem indicação de suporte ECN seria descartado. Ou seja, uma marcação só ocorre como forma de “substituir” um descarte. Roteadores com suporte a ECN podem, mesmo no caso de um pacote com indicação de suporte a ECN, efetuar um descarte como forma de indicar congestionamento.

Já no caso dos protocolos de transporte, a RFC estipula que as ações tomadas pelo algoritmo de controle de congestionamento frente a um pacote com campo ECN ajustado para CE devem ser equivalentes às tomadas quando da detecção de descarte de um pacote. Desta forma, dois protocolos com algoritmos de controle de congestionamento que se diferenciam apenas pelo suporte ou não de ECN tendem a receber a mesma proporção dos recursos de rede disponíveis.

Protocolos multicast de taxa única, onde todos os receptores recebem dados na mesma taxa, variam quanto ao tipo de controle de erro: baseados em NACK (p.ex., [23]) e baseados em TCP-SACK (p.ex., [12]). A seguir, discute-se o emprego de ECN em ambas as classes de protocolos.

### 2.1 Protocolos de Taxa Única Baseados em NACK

Protocolos de taxa única baseados em NACK utilizam como *feedback* uma confirmação negativa (NACK) que indica ao remetente que um receptor detectou a perda de um pacote. Este pacote de NACK acumula tipicamente duas funções:

- alimentar o sistema de recuperação: ao receber um NACK o sistema de recuperação registra que o receptor de onde partiu o NACK não recebeu um determinado pacote, podendo, por exemplo, providenciar a retransmissão do mesmo; e
- alimentar o sistema de controle de congestionamento: um NACK pode levar a uma redução na taxa de transmissão do remetente.

Quando existe um grande número de receptores, o problema de implosão de NACKs deve ser tratado, ou seja, deve haver um mecanismo de supressão de pacotes de NACK redundantes. Esta supressão, que pode ser realizada tanto nas estações participantes da transmissão ([8],[22]) quanto nos roteadores ([19]) e deve, além de considerar os aspectos referentes à recuperação de pacotes, considerar também o mecanismo de controle de congestionamento envolvido.

Como exemplos de mecanismos de controle de congestionamento para esta classe de protocolos podem ser citados o *Loss Tolerant Rate Controller* (LTRC) ([14]), o *TCP-Friendly Multicast Congestion Control* (TFMCC) ([23]) e o *Pragmatic General Multicast Congestion Control* (PGMCC) ([19]). O LTRC representa uma das primeiras iniciativas de controle de congestionamento multicast de taxa única, possuindo diversas limitações, como por exemplo a intolerância à variação nas condições de rede (devido à necessidade de pré-configuração dos valores de RTT<sup>2</sup>). Já o TFMCC e o PGMCC, iniciativas mais recentes, regulam a taxa de transmissão de acordo com a capacidade de recebimento do “pior” receptor. Em ambos os casos, o desafio consiste em determinar dinamicamente o “pior” receptor de um grupo potencialmente grande, sujeito a variações abruptas nas condições de rede.

A utilização de ECN nessa categoria de protocolos é relativamente direta. Conforme já mencionado, o recebimento de um pacote com marcação ECN ativada deve levar, em termos de controle de congestionamento, às mesmas medidas que a detecção de um pacote perdido. Portanto, o protocolo deve empregar um pacote que alimente o algoritmo de controle de congestionamento no remetente, mas que não alimente o algoritmo de recuperação, já que o pacote **não** foi perdido. Para fins de nomenclatura, um pacote com estas características pode ser chamado de “ECN-reply”. A geração destes pacotes está sujeita aos mesmos problemas que a geração de pacotes de NACK, em especial, implosão de *feedback*. Vale ressaltar que, devido às regras de marcação que os roteadores com suporte a ECN devem seguir, o volume de *feedback* não aumenta, pois o número de pacotes de “ECN-reply” aumenta na mesma proporção do que NACKs deixam de ser gerados. Atualmente, o suporte a ECN está sendo incorporado ao TFMCC (vide [22]).

## 2.2 Protocolos de Taxa Única Baseados em TCP-SACK

Nesta categoria, se enquadram os protocolos multicast de taxa única baseados em TCP-SACK ([13]), ou seja, onde o *feedback* gerado pelo receptor contém uma representação da janela de recepção, não apenas informações referentes a um único pacote. São exemplos desta categoria de protocolos o *Polling-based Reliable Multicast Protocol* (PRMP) ([2]), o *Tree-based Reliable Multicast Protocol* (TRAM) ([5]), o *Multicast TCP* (MTCP) ([18]) e o *TCP Single-source Multicast Optimization* (TCP-SMO) ([12]). Tanto o TRAM, baseado em taxa, quanto o MTCP, baseado em janela, organizam os receptores em uma árvore lógica. Tal árvore é usada como forma de obter escalabilidade, pois o mecanismo

---

<sup>2</sup>round-trip time.

de recuperação (retransmissão) e o recebimento de *feedback* não ficam centralizados em um único nó. Em termos de controle de congestionamento, a estruturação em árvore requer que informações sejam agregadas: o transmissor deve receber, indiretamente, informações de todos os nós da árvore. Concebido para ser o mais similar possível com o TCP-SACK, o TCP-SMO não utiliza estruturação em árvore. Apesar de permitir uma organização hierárquica, o PRMP não depende dessa estruturação para ser escalável, pois emprega a técnica de *polling* para restringir o *feedback*.

Em protocolos baseados em TCP-SACK, lacunas na janela passada em pacotes de *feedback* indicam perdas, e perdas são usadas como indicativo de congestionamento na rede. Ao receber um pacote com sinalização ECN ativada, um receptor deve indicar ao transmissor a ocorrência de congestionamento, porém não de perdas, o que torna necessária a utilização de um campo adicional no pacote de *feedback*. Ou seja, assim como no caso de protocolos baseados em NACK, deve haver uma separação entre o *feedback* referente ao mecanismo de recuperação de pacotes e o mecanismo de controle de congestionamento.

### 3 ECN em Multicast com Polling

#### 3.1 Protocolos de Polling

Em protocolos de multicast confiável, a técnica de *polling* é utilizada como maneira de evitar o problema da implosão de *feedback* ([2]). Basicamente, a técnica consiste na gerência, por parte do remetente, da taxa de *feedback* gerada pelos receptores. Desta forma, é possível manter no remetente o controle do andamento da transmissão, o que permite uma maior garantia e eficiência na entrega dos dados. Os mecanismos de ECN aqui propostos são aplicáveis a protocolos multicast baseados em *polling* em geral. Em [1], são descritos cinco modelos de protocolos dessa classe, dos quais se destacam (em termos de desempenho) os protocolos *Response Bucket Polling* (RBP) e *Periodic Polling* (PeP). Por essa razão, para o presente artigo foram conduzidos experimentos apenas com estes dois protocolos, descritos brevemente a seguir.

A idéia básica do RBP é limitar a quantidade de respostas em trânsito através de um mecanismo de *balde de fichas* (*token bucket*). O remetente mantém um balde de fichas, com tamanho pré-configurado, onde cada ficha representa uma permissão para solicitar uma resposta a um receptor. Quando uma solicitação de resposta (POLL) é enviada para algum receptor, uma ficha é retirada do balde, e quando uma resposta é recebida (ou a perda de POLL ou resposta é detectada), uma ficha é devolvida ao balde. Caso o balde se encontre vazio, nenhuma nova resposta pode ser solicitada. Conforme explicado em [1], a precisão do mecanismo pode ser afetada pelo cálculo incorreto dos RTTs. Já no protocolo PeP, o limitante para solicitação de *feedback* é o intervalo entre transmissões de POLLS: periodicamente, o remetente verifica se é necessário enviar um POLL a algum receptor.

#### 3.2 Algoritmo Proposto

Com base nos modelos abstratos, foi desenvolvido um algoritmo de controle de congestionamento baseado em janela que permite a utilização de ECN. Nestes modelos, quando um receptor recebe um pacote de dados, ele atualiza sua janela de recepção mas não envia um ACK imediatamente; confirmações de recebimento são enviadas em conjunto por um receptor similarmente a TCP-SACK, porém apenas mediante uma solicitação de *polling*.

Um novo mecanismo de controle de congestionamento multicast deve atender a uma série de requisitos: deve apresentar um comportamento amigável ao TCP ([5]), não apresentar o problema de *multiplicidade dos caminhos de perda*<sup>3</sup> ([3]) e, por se tratar de um mecanismo baseado em janela, não utilizar um parâmetro de regulação global para todos os receptores, para evitar uma redução demasiada na taxa de transmissão quando da existência de heterogeneidade nas condições de rede entre transmissor e receptores ([11]). Adicionalmente, um esquema com suporte a ECN deve respeitar o comportamento indicado na RFC 3168 ([16]).

A seguir, são apresentadas as etapas do algoritmo proposto: *Detecção de Congestionamento*, *Ajuste da Janela de Congestionamento* e *Obtenção da Taxa de Transmissão*.

### **Detecção de Congestionamento**

O processo de detecção de congestionamento é realizado pelos receptores. São considerados sinais de congestionamento a detecção de uma perda de pacote de dados e a detecção de um sinal ECN emitido por algum roteador, como explicado a seguir.

A perda de um pacote de dados é detectada através de um NACK na janela de recepção. Sempre que um pacote é recebido o receptor verifica, através do número de seqüência do pacote e do estado da janela de recepção, se a atualização da janela criou uma nova lacuna entre pacotes recebidos. Se este for o caso, considera-se que o evento da recepção do pacote indicou ao receptor que existe um provável congestionamento na rede. Diferentemente, a detecção de um sinal ECN é realizada verificando-se os bits reservados no cabeçalho IP.

Em ambos os casos, cada receptor  $R_i$  armazena o tempo de detecção do último sinal de congestionamento. Esta informação é mantida em uma variável denominada  $lct_i$  (de *last congestion time*). Este valor é repassado ao transmissor através de um campo contido em pacotes de respostas, para que o tamanho da janela de congestionamento possa ser ajustado.

### **Ajuste da Janela de Congestionamento**

Assim como no TCP ou em qualquer mecanismo de controle de congestionamento baseado em janela, o ajuste da taxa de transmissão se dá através da variação do tamanho da janela de congestionamento, segundo a política AIMD: *Additive Increase with Multiplicative Decrease*. No esquema aqui proposto, o ajuste é realizado de forma independente para cada receptor  $R_i$ , através do emprego, no transmissor, de um atributo  $cwnd_i$  (*congestion window*, janela de congestionamento). Assim como no TCP, o valor de  $cwnd_i$  serve para limitar o número de pacotes de dados que estão transitando pela rede entre o transmissor e  $R_i$ .

A variação de  $cwnd_i$  é realizada também de forma similar ao TCP, estabelecendo dois períodos diferentes, equivalentes ao *Slow Start* e *Congestion Avoidance*. A transição de um período para outro, bem como o ajuste do limite (*threshold*) envolvido, é realizado da mesma maneira que o TCP, sendo porém *Slow Start* aplicado apenas no início da transmissão.

O **aumento** de  $cwnd_i$  é realizado quando um pacote tem seu recebimento confir-

---

<sup>3</sup>*loss path multiplicity.*

mado, ou seja, sempre que uma resposta traz informações de ACKs referentes a dados com recebimento até então não confirmados por  $R_i$ . Quando isso acontece, o valor de  $cwnd_i$  é aumentado *de forma proporcional ao número de novas confirmações*. Este ajuste pode ser realizado de duas maneiras:

- quando em *Slow Start*, cada nova confirmação de recebimento de pacote de dados acarreta no aumento de um *slot* na janela de congestionamento, ou seja,  $cwnd_i = cwnd_i + 1$ ;
- quando em *Congestion Avoidance*, cada nova confirmação provoca o aumento correspondente a uma fração de *slot* inversamente proporcional ao tamanho atual da janela de congestionamento, ou seja,  $cwnd_i = cwnd_i + 1/cwnd_i$ .

A **redução** do tamanho da janela de congestionamento é realizada similarmente ao TCP: cada sinal de congestionamento faz com que  $cwnd_i$  seja reduzida pela metade (ou seja,  $cwnd_i = cwnd_i/2$ ). O transmissor considera apenas os sinais de congestionamento indicados pelo receptor quando estes estão espaçados por pelo menos um RTT. Esta filtragem é necessária para que sinais de um congestionamento já tratado não levem a uma nova redução da taxa de transmissão, e implementada conforme a seguir. O transmissor controla o espaçamento mantendo, para cada receptor  $R_i$ , uma variável  $lct_i^{eff}$  com o último valor de  $lct_i$  reportado que tenha efetivamente levado a uma redução em  $cwnd_i$ . Ao receber um novo valor de  $lct_i$ , o transmissor verifica se  $lct_i \geq lct_i^{eff} + rtt_i$ , onde  $rtt_i$  representa a estimativa atual de *round-trip time* entre o transmissor e  $R_i$ . Caso a condição seja verdadeira, o transmissor irá reduzir  $cwnd_i$  e atualizar  $lct_i^{eff}$  com  $lct_i$ .

### Obtenção da Taxa de Transmissão

A taxa de transmissão de um protocolo de taxa única deve obedecer aos requisitos de amigabilidade ao TCP para o receptor gargalo. Entretanto, não se pode simplesmente utilizar o mínimo dentre os tamanhos de janela de congestionamento e aplicá-lo sobre a janela de transmissão global, sob pena de reduzir demasiadamente a taxa no caso de o receptor com menor tamanho de janela não corresponder ao receptor com maior RTT ([11]).

Como forma de evitar este problema sem comprometer a amigabilidade ao TCP, o mecanismo aqui proposto calcula, para cada receptor  $R_i$ , um valor  $ht_i$  que corresponde ao pacote de maior seqüência que pode ser transmitido de acordo com o estado da janela de transmissão e  $cwnd_i$ . Desta forma, a transmissão é limitada pelo menor valor de  $ht_i$  dentre os receptores, e não pela janela de congestionamento mínima.

### 3.3 Considerações

Conforme explicado a seguir, o mecanismo proposto preenche os requisitos estabelecidos no início da Seção 3.2. O algoritmo: (a) é amigável ao TCP, pois imita o comportamento do mecanismo de controle de congestionamento do TCP, ajustando os valores de  $cwnd_i$  para cada receptor  $R_i$  segundo o comportamento AIMD, e tomando como referência o receptor mais lento antes de transmitir novos dados; (b) não apresenta o problema da multiplicidade de caminhos de perda, uma vez que existe um valor  $cwnd_i$  para cada receptor, isolando os sinais de congestionamento; (c) não utiliza um parâmetro de regulação único para todos os receptores (problema apontado em [11]): cada receptor possui um tamanho de janela de congestionamento próprio,  $cwnd_i$ ; (d) utiliza ECN conforme a RFC

3168, pois a informação de ECN é considerada como um sinal de congestionamento da mesma forma que um pacote perdido.

Apesar de atender aos requisitos expostos, o algoritmo proposto apresenta limitações. A principal é a diminuição da *responsividade*<sup>4</sup>: a menor frequência de *feedback* por receptor (em relação ao TCP), comum em protocolos multicast confiável, pode influenciar negativamente o comportamento do mecanismo de controle de congestionamento multicast empregado. Uma possível solução para esse problema é determinar os receptores “mais lentos” (a partir do estado das janelas de transmissão e dos RTTs) e reduzir o espaçamento entre respostas enviadas pelos mesmos.

## 4 Análise do Mecanismo Proposto

Esta seção traz uma avaliação por simulação do mecanismo proposto, conduzidas no simulador de redes VINT ns-2 ([4]). Conforme já mencionado, o esquema aqui proposto pode ser aplicado a todos os modelos de protocolo descritos em [1]. Para efeitos de avaliação, foram conduzidos experimentos com o RBP e o PeP. Os resultados obtidos foram similares entre os protocolos, permitindo apresentar a seguir apenas os resultados relativos ao RBP. Os parâmetros utilizados nos experimentos foram os mesmos empregados em [1]. A seguir, são estudadas as características de justiça<sup>5</sup> e amigabilidade ao TCP do mecanismo, para posterior avaliação do impacto da utilização da técnica de ECN sobre o desempenho do protocolo.

### 4.1 Avaliação de Justiça e Amigabilidade ao TCP

Antes de se avaliar o impacto do uso de ECN, foram realizadas simulações com o objetivo de observar o comportamento do algoritmo de controle de congestionamento frente à concorrência pela rede. Conforme visto anteriormente, um dos requisitos para um algoritmo de controle de congestionamento utilizado em ambientes como a Internet é obter justiça na utilização dos recursos entre fluxos concorrentes. Apesar de os experimentos aqui apresentados não validarem completamente o algoritmo, os mesmos são comumente utilizados e, de forma geral, aceitos na literatura como forma de avaliar o comportamento de mecanismos de controle de congestionamento (por exemplo, [19], [6]). A necessidade do uso de topologias mais complexas, que “exijam” mais do algoritmo e representem melhor a realidade da Internet já foi identificada ([9]), porém ainda não existe um conjunto de topologias definido que satisfaça esta necessidade.

A seguir são apresentados os resultados de simulações para avaliação da justiça entre fluxos que utilizam o mecanismo de controle de congestionamento proposto neste trabalho e entre um fluxo que utiliza este mecanismo e fluxos TCP (correspondente à justiça e amigabilidade ao TCP, respectivamente).

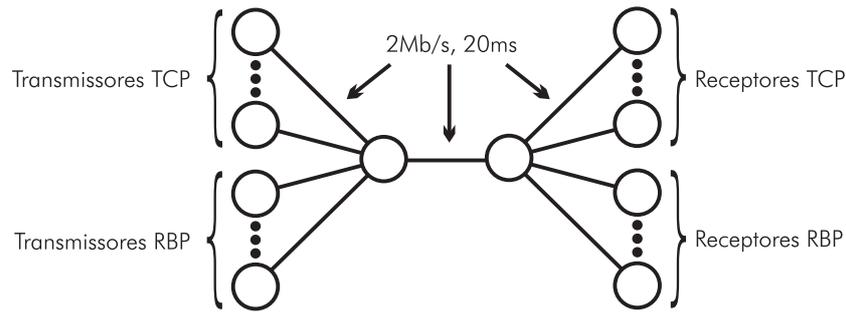
O cenário (Figura 1) empregado nos experimentos, em ambos os casos, é o mesmo, e se baseia na topologia tipicamente utilizada em estudos deste tipo ([19], [6]). A idéia é criar, de maneira simples (para isolar o problema foco da pesquisa), um “link gargalo” comum para diversos fluxos concorrentes.

### Justiça

---

<sup>4</sup>*responsiveness*, ou tempo que o protocolo leva para detectar e tratar uma situação de congestionamento.

<sup>5</sup>*fairness*.



**Figura 1: Topologia empregada nos experimentos de justiça e amigabilidade ao TCP.**

Para avaliar se o mecanismo proposto é justo, foram realizados experimentos onde quatro fluxos RBP concorriam entre si pela banda disponível no gargalo da topologia. Os experimentos se diferenciaram pelo mecanismo de gerência de filas utilizado pelos roteadores da topologia: *drop-tail*, **RED sem suporte a ECN** e **RED com suporte a ECN**.

As simulações foram configuradas de forma que os fluxos iniciassem em diferentes instantes: 0, 20, 40 e 60 segundos. O tempo de término de cada fluxo foi configurado para 160, 140, 120 e 100 segundos, e o número de receptores referente a cada fluxo foi de 20, 15, 10 e 5 receptores, respectivamente.

Um algoritmo que implemente justiça entre fluxos de um mesmo protocolo terá, em geral, uma alocação equânime da largura de banda disponível no gargalo. Na configuração utilizada, a taxa observada para cada transmissor deverá ser similar. Os gráficos a seguir apresentam a variação da taxa de transmissão dos nós remetentes, ou seja, a taxa bruta de dados (incluindo retransmissões e sobrecarga de comunicação) gerada por cada transmissor ao longo do tempo.

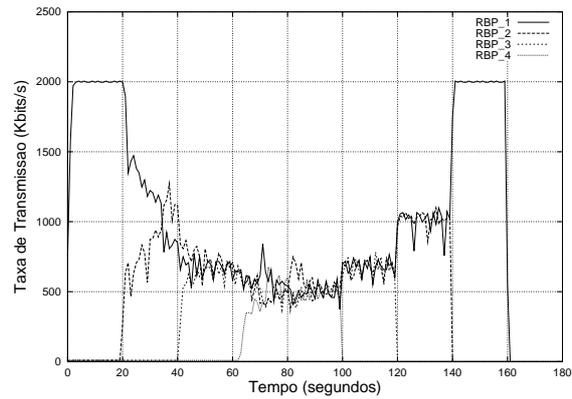
Conforme pode ser observado na Figura 2, os fluxos dividem igualmente a banda disponível; portanto o mecanismo proposto, nas condições analisadas, consegue manter a justiça entre diferentes fluxos. Note-se, também, que o algoritmo se adapta rapidamente quando do surgimento ou término de fluxos. Por fim, a constante variação na taxa de transmissão observada nos gráficos é característica de protocolos que imitam o comportamento AIMD do TCP.

### **Amigabilidade ao TCP**

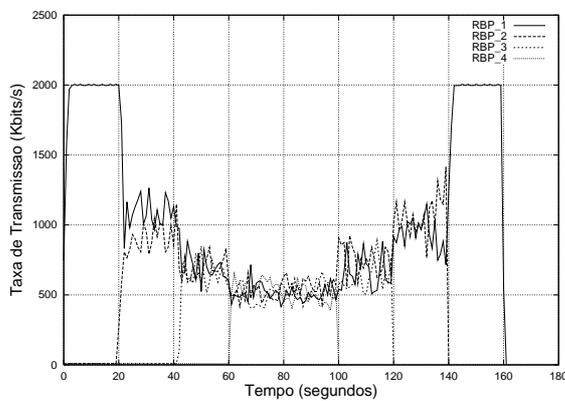
Este conjunto de experimentos tem por objetivo analisar o comportamento do protocolo frente a três fluxos TCP<sup>6</sup> concorrentes, como forma de avaliar a característica de amigabilidade ao TCP do mecanismo. Os cenários avaliados consideram roteadores com diferentes políticas de gerência de filas: *drop-tail*, **RED sem suporte a ECN** e **RED com suporte a ECN**. Nesse último caso, foram simulados cenários com utilização de ECN **ativada** ou **desativada** no TCP.

Foram realizados dois conjuntos de experimentos, variando-se a ordem de início do fluxo RBP em relação aos fluxos TCP. Em ambos os casos, foram usados três fluxos

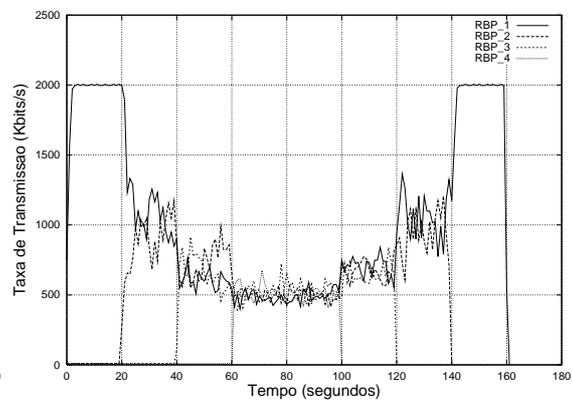
<sup>6</sup>A implementação de TCP utilizada como referência foi a *New Reno*, presente no ns-2.



(a) Drop-tail



(b) RED sem ECN



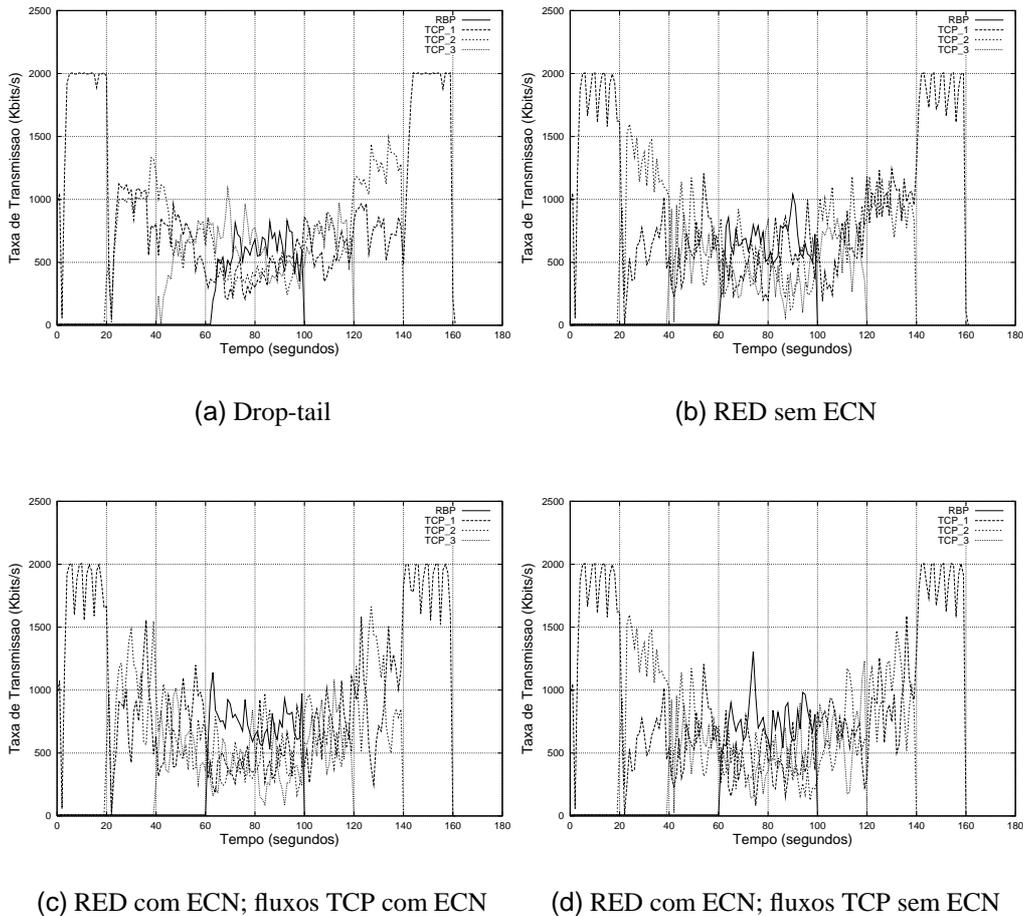
(c) RED com ECN

**Figura 2: Quatro fluxos RBP competindo.**

TCP unicast e um fluxo RBP multicast com número de receptores igual a 20. No primeiro conjunto, apresentado na Figura 3, os fluxos TCP são iniciados nos instantes 0, 20 e 40 segundos e terminados nos instantes 160, 140, 120 segundos, respectivamente. O fluxo RBP inicia no instante 60 e termina no instante 100 segundos.

No segundo conjunto, conforme ilustrado na Figura 4, o fluxo RBP foi iniciado no instante 0 segundos e terminado no instante 160 segundos, enquanto os fluxos TCP foram iniciados nos instantes 20, 40 e 60 segundos, e terminados nos tempos 140, 120 e 100 segundos, respectivamente.

Em ambos os conjuntos, as taxas de transmissão dos fluxos TCP e RBP reportadas nas Figuras 3 e 4 se equivalem. Os resultados obtidos mostram que o mecanismo proposto se comportou adequadamente frente ao TCP em todos os casos estudados, incluindo os experimentos onde o fluxo RBP utiliza ECN e os fluxos TCP não fazem uso da técnica (Figuras 3.(d) e 4.(d)).

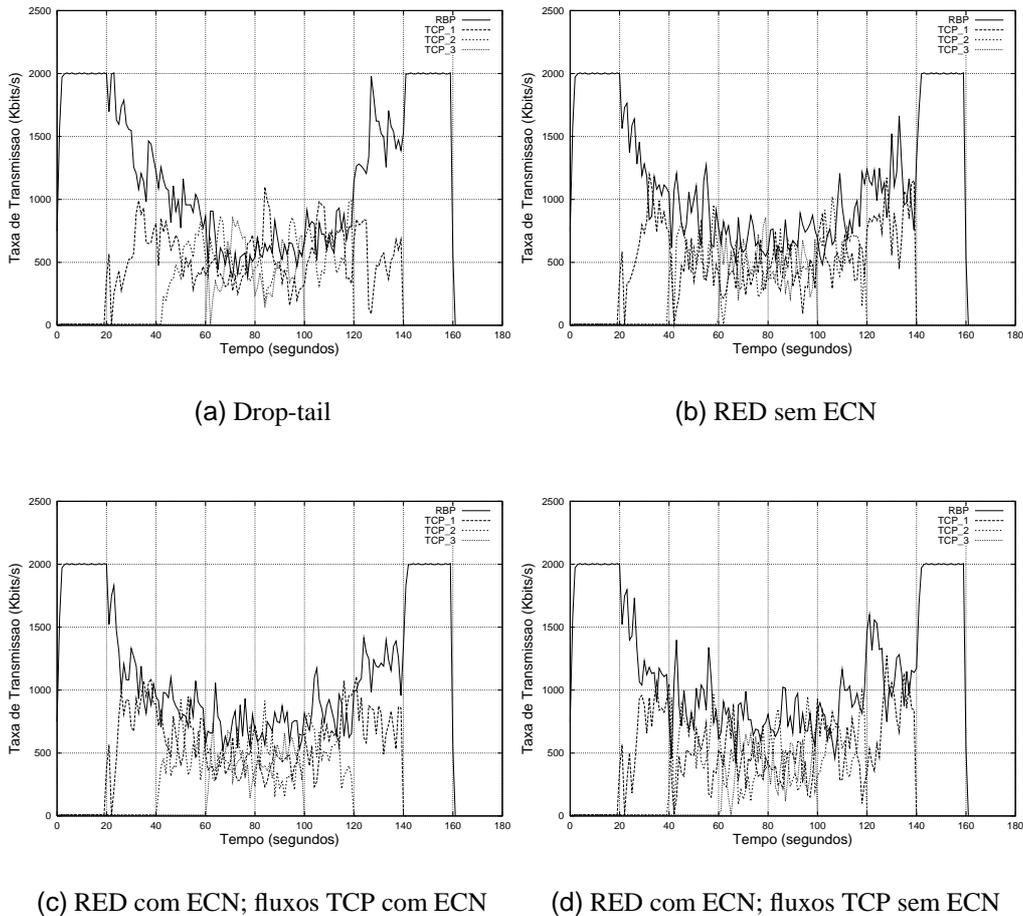


**Figura 3: RBP competindo com três fluxos TCP. Fluxo RBP iniciado após fluxos TCP.**

## 4.2 Impacto de ECN

O impacto da utilização de ECN se dá na medida em que descartes de pacotes são evitados, com efeito positivo em termos de latência (não é necessário detectar a perda de um pacote através de *timeout*) e de largura de banda (diminui o número de retransmissões). A topologia utilizada na seção anterior apresenta um único gargalo: as perdas de pacotes observadas pelos receptores são as mesmas. Entretanto, num ambiente real, os descartes podem ocorrer de forma independente para cada receptor. Por isso, optou-se pela utilização de uma topologia mais complexa, que contemplasse este fator. A idéia da topologia continua sendo simples: neste caso, em vez de todos os receptores compartilharem o mesmo gargalo, cada receptor tem um gargalo independente, compartilhando-o com fluxos diferentes dos fluxos que concorrem com os demais receptores (Figura 5).

Foram realizados experimentos variando-se o número de fluxos TCPs para cada receptor RBP (entre 1 e 10 fluxos TCP), e a política de gerência das filas dos roteadores que compõem a topologia (novamente variando entre *drop-tail*, RED sem suporte a ECN e RED com suporte a ECN). Por não ser compatível com a técnica de ECN, os experimentos com filas *drop-tail* servem apenas de referência para os demais resultados.



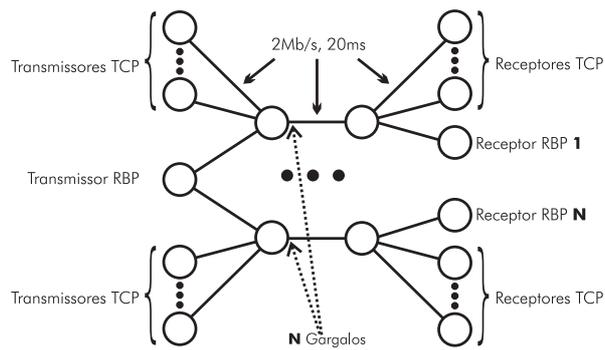
**Figura 4: RBP competindo com três fluxos TCP. Fluxo RBP iniciado antes dos fluxos TCP.**

Em todos os experimentos, o fluxo RBP, iniciado no instante 10 segundos, realizou a entrega confiável de 5000 pacotes, com 1000 bytes de dados cada, a 20 receptores. Como forma de inserir aleatoriedade e independência entre os gargalos, os fluxos TCP foram iniciados em tempos diferentes, de acordo com um valor aleatório entre 0 e 5 segundos. Os fluxos TCP foram terminados somente após o final da transmissão do fluxo RBP. O tempo de início do fluxo RBP foi postergado (para 10 segundos), para que a situação da rede se estabilizasse após o início dos fluxos TCP.

A seguir, são apresentados os resultados obtidos em função das seguintes métricas: o número de pacotes de dados (re)transmitidos, o *throughput* efetivo (*goodput*) da transmissão multicast e, por fim, o tempo médio de atraso para a entrega dos pacotes de dados a todos os receptores.

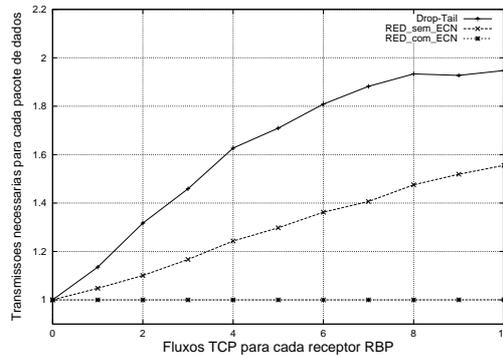
### Pacotes de Dados Transmitidos

Os resultados abaixo demonstram o impacto do uso da técnica de ECN sobre o número de pacotes de dados que precisam ser (re)transmitidos. Este número é diretamente proporcional ao número de perdas decorrentes de descartes. Os resultados obtidos foram normalizados pelo número de pacotes de dados a serem entregues pelo remetente RBP.



**Figura 5: Topologia empregada na avaliação do impacto de ECN**

Por exemplo, quando o resultado desta normalização é igual a 1, nenhuma retransmissão foi necessária; quando igual a 2, cada pacote exigiu, na média 1 retransmissão para que a entrega fosse realizada a todos os receptores.

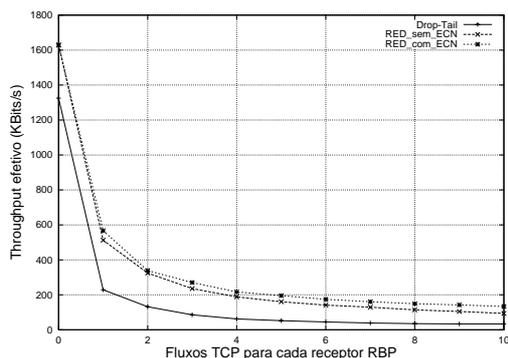


**Figura 6: Variação do número de transmissões necessárias com o aumento do número de fluxos TCP por receptor RBP.**

Através dos resultados obtidos (Figura 6), observa-se que, quando utilizada política RED com suporte a ECN, nenhuma retransmissão foi necessária, independentemente do número de fluxos TCP concorrentes. A ausência de retransmissões ocorre porque os descartes foram evitados através do uso da marcação de pacotes como forma de indicar congestionamento. Em contraste, no caso sem suporte a ECN, o número de retransmissões cresceu linearmente com o número de fluxos TCP concorrentes. Resultado ainda pior foi observado no caso da utilização da política *drop-tail* nos roteadores, tendo o índice ultrapassado 2 no caso de 10 fluxos TCP para cada receptor RBP.

### **Throughput Efetivo**

O *throughput* efetivo é determinado dividindo-se a quantidade de dados transferidos de maneira confiável pelo período de transmissão. Define-se *período de transmissão* como o intervalo compreendido entre o envio do primeiro pacote de dados e o recebimento da última confirmação positiva necessária (quando todos os pacotes tiverem sido confirmados por todos os receptores). Desconta-se aí o período inicial, para estabelecimento de sessão, e o período final, para fechamento da mesma, sobrecarga presente em todos os protocolos confiáveis com controle de grupo.

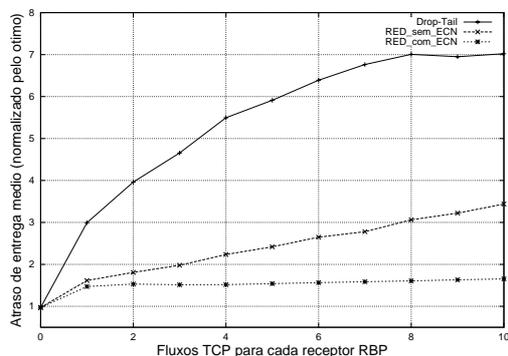


**Figura 7: Variação do *throughput* efetivo com o aumento do número de fluxos TCP por receptor RBP.**

Através do gráfico da Figura 7, observa-se que a utilização da técnica de ECN traz um ganho significativo em relação ao *throughput*. Em geral, o aumento de desempenho através de ECN foi maior do que a diferença resultante do uso da política RED no lugar da política *drop-tail*. Por exemplo, considerando 10 fluxos TCP concorrentes para cada receptor RBP, o *throughput* efetivo verificado para os três casos foi de 210, 75 e 33 Kbits/s; ou seja, o uso de ECN representou um ganho de 180% em relação a RED puro e 536% em relação a *drop-tail*.

### Atraso de Entrega

O atraso de entrega aqui considerado se refere ao intervalo de tempo entre o envio de um pacote e entrega do mesmo a todos os receptores, incluindo fatores como latência dos links, atraso de enfileiramento e perdas de pacotes (recuperados através de retransmissões). A entrega de um pacote só é considerada efetivada em algum receptor quando todos os pacotes anteriores (com número de seqüência menores) tenham sido recebidos pelo receptor. A minimização deste atraso é importante em aplicações interativas ou onde a transmissão ocorra de maneira esporádica, envolvendo um pequeno volume de dados, como no caso de sistemas de comunicação em grupo ([20]).



**Figura 8: Variação do tempo de entrega dos pacotes com o aumento do número de fluxos TCP para cada receptor RBP.**

O gráfico da Figura 8 ilustra o atraso de entrega médio da transmissão, normalizado pelo “atraso de entrega ótimo”, ou seja, considerando apenas as latências dos cami-

nhos envolvidos. Primeiramente, observa-se uma grande disparidade entre os resultados obtidos com o uso de roteadores RED e com o uso de roteadores *drop-tail*: o uso de roteadores RED, mesmo sem suporte a ECN, representa uma melhora substancial no atraso de entrega. Adicionalmente, a técnica de ECN diminui consideravelmente o aumento do atraso que ocorre conforme aumenta o número de fluxos TCP concorrentes.

## 5 Conclusões e Trabalhos Futuros

Este trabalho oferece duas contribuições importantes: primeiro, apresenta um mecanismo de controle de congestionamento para protocolos multicast de taxa única, mostrando que o mesmo implementa justiça entre seus fluxos e é amigável ao TCP. Segundo, porém não menos importante, propõe e avalia por simulações o uso da técnica de ECN nesses protocolos, enfatizando mecanismos baseados em *polling*.

Em diversas simulações realizadas no ns-2, o mecanismo proposto apresentou um comportamento adequado segundo as métricas de justiça e amigabilidade ao TCP. Foram realizados também experimentos com o objetivo de avaliar o impacto do uso da técnica de ECN em protocolos multicast. Os resultados obtidos demonstraram melhoras significativas no desempenho do protocolo quando da utilização da técnica.

Apesar da técnica de ECN ainda ser pouco utilizada em protocolos multicast, observou-se que seu emprego não apenas é viável como também pode ser realizado de maneira relativamente simples. Os autores acreditam que, dada a relação entre TCP-SACK e *polling*, essas conclusões podem ser de certa maneira estendidas a protocolos tais como os descritos na Seção 2.2.

Como trabalhos futuros, vislumbra-se estender as simulações realizadas com cenários mais realísticos, com topologias mais complexas e outras características de tráfego, particularmente considerando aplicações ligadas a comunicação em grupo. Além disso, pretende-se estudar o impacto do uso de ECN em outros protocolos multicast. Por fim, deseja-se aperfeiçoar o mecanismo de controle de congestionamento proposto para aumentar sua *responsividade*, realizando simulações que avaliem melhor este aspecto do mecanismo.

## Referências

- [1] M. P. Barcellos and A. Detsch. Avaliação de Desempenho de Protocolos Multicast com Conhecimento de Grupo Baseados em Polling. In SBC, editor, *SBRC*, volume 1, pages 1–16, Búzios, Brazil, 2002.
- [2] M.P. Barcellos and P. Ezhilchelvan. A Reliable Multicast Protocol using Polling for Scalability. In *IEEE INFOCOM'98*, pages 1180–1187. IEEE, April 1998.
- [3] S. Bhattacharyya, D. Towsley, and J. Kurose. The Loss Path Multiplicity Problem in Multicast Congestion Control. In *IEEE INFOCOM'99*, pages 856–863. IEEE, March 1999.
- [4] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in network simulation. *IEEE Computer*, 33(5):59–67, May 2000.

- [5] Dah Ming Chiu, Miriam Kadansky, Joe Provino, J. Wesley, H.P. Bischof, and H. Zhu. A Congestion Control Algorithm for Tree-based Reliable Multicast Protocols. In *IEEE INFOCOM 2002*, June 2002.
- [6] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, 24(5):10–23, 1994.
- [7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.
- [8] S. Floyd, V. Jacobson, Ching-Gung Liu, S. McCanne, and Lixia Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803, 1997.
- [9] S. Floyd and Eddie Kohler. Internet Research Needs Better Models, 2002.
- [10] P. Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti. Congestion Control Mechanisms and the Best Effort Service Model. *IEEE Network*, 15(3):16–25, 2001.
- [11] S. J. Golestani and K.K. Sabnani. Fundamental Observations on Multicast Congestion Control in the Internet. In *IEEE INFOCOM'99*, pages 990–1000, 1999.
- [12] S. Liang and D. Cheriton. TCP-SMO: Extending TCP to Support Medium-Scale Multicast Applications. In *IEEE INFOCOM 2002*, 2002.
- [13] M. Mathis and et al. TCP Selective Acknowledgement Options. RFC 2018, IETF, April 1996.
- [14] T. Montgomery. A Loss Tolerant Rate Controller for Reliable Multicast. Technical report, West Virginia University, August 1997.
- [15] J. Postel. Internet Control Message Protocol. RFC 792, IETF, September 1981.
- [16] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, IETF, September 2001.
- [17] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-End Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In *IEEE INFOCOM'99*, pages 1337–1345, 1999.
- [18] I. Rhee, N. Ballaguru, and G.N. Rouskas. MTCP: Scalable TCP-like Congestion Control for Reliable Multicast. Technical Report TR-98-01, 2 1998.
- [19] L. Rizzo. pgmcc: a TCP-friendly Single-rate Multicast. In *SIGCOMM*, pages 17–28, 2000.
- [20] P. Verissimo and L. Rodrigues. Group Orientation: a paradigm for modern distributed systems. In *5th ACM SIGOPS European Workshop*, September 1992.
- [21] J. Widmer, R. Denda, and M. Mauve. A Survey on TCP-Friendly Congestion Control. *IEEE Network*, 15(3):28–37, 2001.
- [22] J. Widmer and M. Handley. TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification. INTERNET DRAFT: draft-ietf-rmt-bb-tfmcc-01.txt, 4 November 2002.
- [23] Jörg Widmer and Mark Handley. Extending Equation-based Congestion Control to Multicast Applications. In *ACM SIGCOMM 2001*, 2001.