# Otimização da qualidade de vídeo em redes multicast não cooperativas

Marcelo Dias de Amorim[1], Otto Carlos M. B. Duarte[2] e Guy Pujolle[3]

[1]LRI - Univ. Paris XI – Bât. 490 – 91405 – Orsay Cedex – France

[2]GTA/UFRJ – P.O. Box 68504 – 21945-970 – Rio de Janeiro – RJ - Brasil

[3]LIP6 - Univ. Paris VI – 8, rue du Capitaine Scott – 75015 - Paris – France

**Resumo**

Uma das mais importantes características a serem inseridas nas aplicações multimídia das próximas gerações é a de adaptação. Este artigo propõe o Algoritmo Direto que otimiza o grau de satisfação dos diferentes destinatários em um ambiente de distribuição multiponto de vídeo em camadas. Este algoritmo é baseado em um mecanismo que controla dinamicamente as taxas de transmissão das diferentes camadas de vídeo, baseado em pacotes de controle enviados pelos receptores à fonte. O principal objetivo é de combinar eficientemente estes pacotes de controle nos nós intermediários de forma a evitar implosão de pacotes na fonte. O esquema proposto é otimizado para atingir a melhor qualidade possível do vídeo nos receptores. É também proposto o Algoritmo Direto com camadas virtuais. O esquema de camadas virtuais induz os nós intermediários a guardar um certo número extra de estados da sessão multicast, o que reduz a degradação do vídeo para os receptores como um todo. Os resultados mostram que o esquema proposto leva a uma excelente melhoria da qualidade global do vídeo em receptores heterogêneos sem custo extra de banda-passante.

**Abstract**

Adaptability is the most promising feature to be applied in future robust multimedia applications. In this paper, we propose the Direct Algorithm to improve the degree of satisfaction at heterogeneous receivers in multi-layered multicast video environments. The algorithm relies on a mechanism that dynamically controls the rates of the video layers and is based on feedback control packets sent by the receivers. The algorithm also addresses scalability issues by implementing a merging procedure at intermediate nodes in order to avoid packet implosion at the source in the case of large multicast groups. The proposed scheme is optimized to achieve high global video quality and reduced bandwidth requirements. We also propose the Direct Algorithm with a virtual number of layers. The virtual layering scheme induces intermediate nodes to keep extra states of the multicast session, which reduces the video degradation for the whole of receivers. The results show that the proposed scheme leads to improved global video quality at heterogeneous receivers with no cost of extra bandwidth.

Este trabalho recebeu suporte da CAPES, COFECUB, CNPq, UFRJ, RNRT e CNRS.

# 1 Introdução

Heterogeneidade em equipamentos intermediários e finais é um dos maiores problemas para o desenvolvimento de aplicações multicast nas atuais arquiteturas de rede tais como a Internet. Em uma sessão multicast, se a fonte transmitir em uma taxa compatível com os receptores mais lentos, os caminhos que levam àqueles mais rápidos serão sub-utilizados. Por outro lado, se a fonte transmitir em mais alta velocidade, os receptores mais lentos não serão atendidos. Essas características são intrínsecas à distribuição de vídeo, pois grande parte das aplicações de vídeo são multicast por natureza [1, 2, 3].

Neste artigo é proposto um sistema adaptativo que visa otimizar a qualidade global da sessão multicast em redes com receptores heterogêneos. Este sistema é baseado em fontes/receptores cooperativos e utiliza a técnica de codificação em camadas para diferenciar grupos de receptores com capacidades similares [4, 5, 6, 7]. Fontes e receptores trocam entre si pacotes de controle contendo informações sobre o estado da rede. Baseada nestes pacotes, a fonte adapta as taxas de transmissão das diferentes camadas de vídeo à condição atual da rede. Esta proposta é, entretanto, diretamente ligada a algumas questões de escalabilidade quando o número de receptores da sessão é muito alto. Neste caso, quando todos os pacotes enviados pelos receptores chegam na fonte, ocorre o que chamamos implosão de pacotes na fonte.

O principal componente do sistema é o Algoritmo Direto, que tem por objetivo eliminar o problema de implosão de pacotes através de um procedimento de combinação de pacotes concorrentes em nós intermediários da rede. Este algoritmo é implementado de forma a maximizar a qualidade global do vídeo nos receptores. A qualidade global do vídeo é mensurada pelo nível de satisfação de cada receptor, isto é, pela diferença entre as taxas de vídeo requerida e recebida. Através da análise de diferentes topologias de rede, é provado que o esquema proposto resulta em melhor qualidade global de recepção nos diferentes destinatários. Além disso, os resultados mostram que a proposta leva a uma redução na quantidade de banda-passante alocada mantendo altos valores de satisfação do usuário.

É também abordado neste artigo a utilização do Algoritmo Direto com um número virtual de camadas. Neste caso, os nós intermediários consideram que a fonte é capaz de transmitir um número maior de camadas de vídeo. Será mostrado que esta técnica evita o descarte de informações relevantes que, se armazenadas, poderiam ser confrontadas com outras de sub-redes vizinhas. Isso significa que o esquema de camadas virtuais reduz a probabilidade de que informações importantes (sob o ponto de vista da sessão multicast) sejam eliminadas. É mostrado que a utilização desta técnica leva a resultados próximos ao limite empírico. O limite empírico é obtido quando nenhum algoritmo de combinação de pacotes é realizado nos nós intermediários, mas apenas na fonte.

A qualidade global da sessão multicast é calculada pelo agregado dos índices de satisfação de cada receptor. Através da análise de diferentes topologias de rede, é mostrado que a proposta realmente resulta em ótimos valores de qualidade de recepção em receptores heterogêneos. Além disso, os resultados também mostram que os valores obtidos não exigem alocação extra de banda-passante.

Este artigo está organizado da seguinte forma. Na seção 2 é proposto e analisado o Algoritmo Direto. A seção 3 apresenta o Algoritmo Direto com camadas virtuais. Muitas análises e simulações são apresentadas na seção 4. Finalmente, a seção 5 conclui este artigo.

# 2 Otimizando a satisfação nos receptores

O sistema adaptativo proposto neste artigo consiste em uma sessão multicast onde a fonte de vídeo utiliza a técnica de codificação em camadas. A taxa de transmissão de cada camada é baseada em pacotes de reconhecimento enviados pelos receptores. O principal componente deste sistema é o algoritmo usado para combinar os pacotes nos nós intermediários de forma a evitar implosão de pacotes de controle na fonte. Neste esquema, a fonte envia em multicast um pacote de controle (que chamaremos pacote de prospecção) aos destinatários. Ao atravessar a rede, os nós intermediários marcam estes pacote com a banda-passante disponível em cada um dos enlaces, $c_s(l)$. Quando um receptor $r$, pertencente à sessão multicast $m$, recebe este pacote de prospecção, ele calcula imediatamente a máxima vazão que ele pode receber, dada por

$$b^{r,m} = \min_{l \in L_{s,r}} c_s(l), \tag{1}$$

onde $L_{s,r}$ é o grupo de enlaces atravessados pelo pacote de prospecção entre a fonte $s$ e o receptor $r$. Os receptores criam então pacotes de retorno (reconhecimento) contendo a taxa máxima de recepção e os enviam à fonte. No momento de rotear estes pacotes, os nós intermediários os combinam de forma a evitar implosão na fonte. O ideal exigiria que as rotas fonte/destinos e destinos/fonte fossem simétricas, o que pode ser resolvido através da utilização de algoritmos de roteamento assimétricos como por exemplo o HBH [8, 9].

Para o procedimento de combinação de pacotes, define-se como *pacote de retorno ótimo* aquele que satisfaz às seguintes definições.

**Definição 1** *A degradação global do vídeo para uma sessão multicast em múltiplas camadas é igual à soma das diferenças entre a taxa de vídeo requerida e a taxa de vídeo recebida, para todos os receptores pertencente à sessão.*

**Definição 2** *Após a execução do procedimento de combinação de pacotes, o pacote de retorno ótimo é aquele que resulta na* **menor** *degradação global do vídeo nos receptores.*

Um pacote de reconhecimento possui $k$ entradas, $1 \leq k \leq L$, onde $L$ é o número de camadas de vídeo enviadas pela fonte. Cada entrada $e_i, i = 1 \ldots k$, possui por sua vez dois campos: a vazão $f_{e_i}$ e o número de receptores $f^*_{e_i}$ requerindo a vazão $f_{e_i}$. Nosso sistema é similar ao proposto em [10], onde os pacotes de reconhecimento são armazenados em um vetor temporário $\vec{t}$ para processamento futuro. As entradas do vetor temporário são organizadas de forma que $f_{e_{i-1}} < f_{e_i} < f_{e_{i+1}}, \forall i$. Se $f_{e_i} = f_{e_{i+1}}$, então $e_i$ e $e_{i+1}$ são combinados segundo as equações

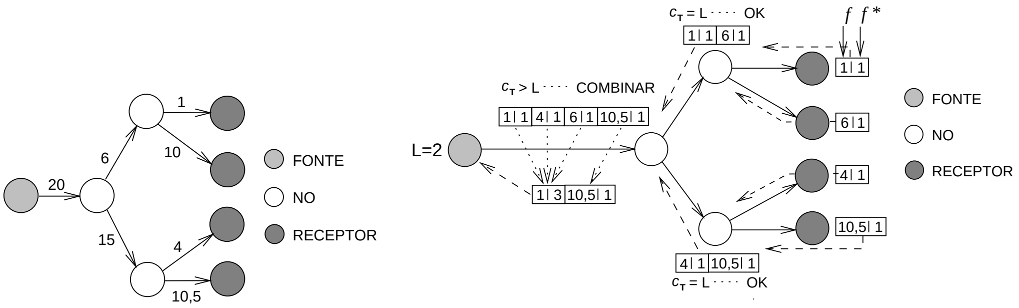
$$\begin{aligned} f_{e_{i,\oplus}} &= f_{e_{i,\ominus}}, \\ f^*_{e_{i,\oplus}} &= f^*_{e_{i,\ominus}} + f^*_{e_{i+1,\ominus}}, \end{aligned}$$

onde os símbolos $\ominus$ e $\oplus$ representam os valores das variáveis respectivamente antes e após o procedimento de combinação de pacotes. Se o vetor temporário possuir um número de entradas $c_T$ maior que o número de camadas transmitidas pela fonte, então o nó executa

Na prática, duas entradas são automaticamente combinadas se a distância $f_{e_{i+1}} - f_{e_i}$ for menor que um determinado valor $\alpha$. Neste caso, diz-se que estas entradas são “compatíveis.”

um algoritmo para eliminar $c_T - L$ entradas. Chama-se de " pacote candidato" qualquer pacote que pode ser obtido após o procedimento de combinação. Aquele que será escolhido para ser transmitido à fonte depende da política adotada pelo algoritmo.

O método clássico elimina a entrada $j$ que resulta na mais elevada "taxa de otimização" $G$. A taxa de otimização é equivalente a taxa total recebida pelos receptores quando se elimina a entrada $e_j$, sendo calculada como $G = \sum_{i=1,i\neq j}^{c_T} f_{e_i} f_{e_i}^*$. O número de receptores $f_{e_j}^*$ da entrada eliminada é adicionado ao número de receptores na $(j-1)$-ésima entrada. Estes passos são executados em um *loop* até que o número de entradas no vetor temporário seja reduzido ao máximo número de camadas transmitidas pela fonte.




(a) Estado inicial da rede. (b) Procedimento de combinação.

Figura 1: Evolução do procedimento de combinação.

Embora o descarte seja feito sobre as entradas que resultam na mais elevada taxa de otimização, isto não garante totalmente a otimização da qualidade do vídeo nos receptores. A figura. 1(a) mostra uma situação onde a fonte transmite duas camadas de vídeo. A taxa de otimização $G_{e_j}$ é obtida quando se descarta entrada $e_j$.

No nó que precede a fonte, o vetor temporário possui quatro entradas e portanto duas devem ser eliminadas. A seqüência abaixo e a figura 1(b) mostram cada passo do algoritmo de redução de entradas.

**Passo 1** $\vec{t} = \{(1,1),(4,1),(6,1),(10.5,1)\}$
$G_{e_2} = 18,5$
$G_{e_3} = 19,5$
$G_{e_4} = 17$

$G_{e_3}$ é a máxima taxa de otimização, então a entrada $e_3$ é eliminada e $f_{2,\oplus}^* = f_{2,\ominus}^* + f_{3,\ominus}^*$. Imediatamente antes da eliminação da segunda entrada extra, o vetor temporário possui as seguintes entradas

$$
\begin{array}{lll}
e_{1,\ominus} \Rightarrow & f_{1,\ominus} = 1 & f_{1,\ominus}^* = 1 \\
e_{2,\ominus} \Rightarrow & f_{2,\ominus} = 4 & f_{2,\ominus}^* = 2 \\
e_{3,\ominus} \Rightarrow & f_{3,\ominus} = 10,5 & f_{3,\ominus}^* = 1
\end{array}
$$

A primeira entrada nunca é descartada pois corresponde ao receptor mais lento.

Por razões de simplicidade de notação, em alguns exemplos deste artigo serão suprimidas as unidades das variáveis.

A segunda etapa pode então ser concluída.

**Passo 2** $\vec{t} = \{(1,1),(4,2),(10.5,1)\}$
$G_{e_2} = 13,5$
$G_{e_3} = 13$

Na etapa 2, $G_{e_2}$ é maior que $G_{e_3}$. Assim, $e_2$ é descartada e o nó transmite à fonte o pacote resultante. Este procedimento é ilustrado na figura 1(b). De acordo com o pacote resultante, os receptores localizados na sub-árvore serão capazes de receber duas camadas de vídeo. A primeira a uma taxa igual a 1 e a segunda a uma taxa igual a 9.5. Entretanto, esse valores não são ótimos porque eles não resultam no mais alto índice de satisfação dos usuários. Na verdade, as camadas que resultariam no mais alto índice de satisfação dos usuários seriam $l_1 = 1$ e $l_2 = 5$. Será mostrado na seqüência a razão pela qual o método clássico não resultou no valor ótimo.

Seja $\mathbf{S}$ um sub-conjunto das entradas do vetor temporário e o número de elementos de $\mathbf{S}$ igual a $|\mathbf{S}| = c_T - L$. $\gamma_{\mathbf{S}}$ é a degradação global do vídeo quando se descarta as entradas $e_i \in \mathbf{S}$. $r_{r_i}$ é a taxa do vídeo requerida pelo receptor $r_i$ e $\hat{r}_{r_i}$ é a taxa recebida. $\gamma_{r_i}$ é a degradação no receptor $r_i$, isto é, $\gamma_{r_i} = r_{r_i} - \hat{r}_{r_i}$. Por conseqüência, tem-se que

$$\gamma_{\mathbf{S}} = \sum_{i=1}^{N} \gamma_{r_i}, \tag{2}$$

onde $N$ é o número total de receptores na sessão multicast. No exemplo acima, tem-se

$\gamma_{(e_2,e_3)} = 8,5,$
$\gamma_{(e_2,e_4)} = 7,5,$
$\gamma_{(e_3,e_4)} = 8.$

Vale lembrar que a mais baixa degradação corresponde à mais elevada qualidade global do vídeo nos receptores, o que equivale, no caso acima, a $\gamma_{(e_2,e_4)} = 7,5$. Estes valores correspondem à camada de base sendo transmitida a uma taxa igual a $l_1 = 1$ e à segunda camada a uma taxa $l_2 = 5$. Na seqüência será mostrado que o método clássico não leva a esses valores pois o algoritmo de combinação é executado em passos consecutivos independentes.

Seja $\mathbf{U}$ todos os pacotes possíveis de serem obtidos pelo algoritmo de combinação. Seja também $\mathbf{O} \subset \mathbf{U}$, $|\mathbf{O}| = L$, o subconjunto de entradas que resultam no mais baixo nível de degradação do vídeo nos receptores. Além disso,

$$\vec{t} = \{(f_{o_1}, f^*_{o_1}), \ldots, (f_{o_2}, f^*_{o_2}), \ldots, (f_{o_L}, f^*_{o_L}), \ldots, (f_{c_T}, f^*_{c_T})\}$$

é o vetor temporário a ser reduzido. Seja $\hat{\mathbf{O}} \subset \mathbf{U}$, $|\hat{\mathbf{O}}| = L$, um outro subconjunto diferente de $\mathbf{O}$ em pelo menos um dos elementos. A desigualdade seguinte deve então ser verdadeira:

$$\sum_{\forall k, k \notin \mathbf{O}} f^*_k (f_k - f_{\mathbf{O}_x}) < \sum_{\forall k, k \notin \hat{\mathbf{O}}} f^*_k (f_k - f_{\hat{\mathbf{O}}_x}), \quad \forall \hat{\mathbf{O}}, \tag{3}$$

A banda-passante total que os receptores poderão receber é igual a 10.5. Isso corresponde à primeira camada igual a 1 e a segunda a 9.5.

onde $\mathbf{O}_x$ deve ser escolhido de forma que $\mathbf{O}_x \leq f_k < \mathbf{O}_{x+1}$. O método clássico não satisfaz essa desigualdade se, em *qualquer* um dos passos do algoritmo, existir ao menos uma entrada $e_v \in \mathbf{O}$ de tal maneira que

$$f^*_{e_v}(f_{e_v} - f_{e_{v-1}}) < f^*_{e_k}(f_{e_k} - f_{e_{k-1}}), \quad \forall k, k \neq v. \tag{4}$$

A análise seguinte será realizada no caso onde o vetor temporário possui quatro entradas e duas devem ser eliminadas. Tem-se então

$$\mathbf{S}_T = \{(f_{e_1}, f^*_{e_1}), (f_{e_2}, f^*_{e_2}), (f_{e_3}, f^*_{e_3}), (f_{e_4}, f^*_{e_4})\}. \tag{5}$$

Supondo que as entradas a serem mantidas sejam $e_1$ e $e_3$, as desigualdades seguintes devem ser verdadeiras:

$$\begin{cases} f^*_{e_2}(f_{e_2} - f_{e_1}) < f^*_{e_3}(f_{e_3} - f_{e_2}) + f^*_{e_4}(f_{e_3} - f_{e_2}), \\ \\ f^*_{e_4}(f_{e_4} - f_{e_3}) < f^*_{e_3}(f_{e_3} - f_{e_2}) + f^*_{e_3}(f_{e_2} - f_{e_1}). \end{cases} \tag{6}$$

Combinando-se as equações (3) e (6), e após alguma manipulação, conclui-se que o método clássico não pode resultar nos valores ótimos se

$$\begin{cases} f^*_{e_3}(f_{e_3} - f_{e_2}) < f^*_{e_2}(f_{e_2} - f_{e_1}) < f^*_{e_3}(f_{e_3} - f_{e_2}) + f^*_{e_4}(f_{e_3} - f_{e_2}), \quad \text{e} \\ \\ f^*_{e_3}(f_{e_3} - f_{e_2}) < f^*_{e_4}(f_{e_4} - f_{e_3}) < f^*_{e_3}(f_{e_3} - f_{e_2}) + f^*_{e_3}(f_{e_2} - f_{e_1}). \end{cases} \tag{7}$$

O Algoritmo Direto proposto neste artigo evita situações como as descritas acima. Além disso, resulta no pacote ótimo em cada um dos nós pois é executado em apenas um *loop*. Em um caso geral onde a fonte transmite $L$ camadas de vídeo $(l_1 ... l_L)$ e em algum dos nós intermediários o vetor temporário possui $c_T$ entradas, o Algoritmo Direto funciona da seguinte maneira. Faz-se $l_1 = f_{e_1}$ e em seguida são computados $l_2 ... l_L$ de forma a minimizar a *soma das perdas* $\Delta_S$:

$$\Delta_S = \sum_{\forall k, k \notin \mathbf{O}} f^*_k(f_k - f_{\mathbf{O}_x}). \tag{8}$$

No Algoritmo Direto, todas as entradas são descartadas no mesmo *loop*, pois todos os pacotes candidatos são examinados. Além disso, ao se calcular o mínimo valor de $\Delta_S$, garante-se que a qualidade global do vídeo seja otimizada nos nós intermediários. É mostrado abaixo o pseudo-algoritmo correspondente:

```
Algoritmo Direto (in: vetor temporário;
                  out: pacote de retorno ótimo)
γ = 0;
γ_tmp = 0;
fim = 1;
Calcular O^tmp ← "primeiro pacote candidato";
enquanto (∃ "próximo pacote candidato")
{
   para (k = 1;k ≤ c_T;k + +)
      γ_tmp = γ_tmp + f*_k(f_k − f_{O_x});
   se (γ_tmp < γ) então
   {
      γ = γ_tmp;
      O = O^tmp;
   }
   Calcular próximo O^tmp ← "pacote candidato";
}
```

Uma questão a ser abordada é com relação à complexidade do algoritmo, visto que para cada procedimento de combinação de pacotes o nó precisa considerar todas as possibilidades de descarte de entradas. Seja um nó $n$ na iminência de executar o algoritmo de combinação. Este nó possui $k$ enlaces de entrada $p_i^n$, $i = 1 \ldots k$, e um enlace de saída $p_{out}^n$. Será realizada na seqüência a análise no caso extremo onde o processamento no nó $n$ é máximo, de forma a se definir os limites de desempenho do sistema. Para tal, são assumidas as seguintes premissas:

- todos os enlaces de entrada $p_i^n, i = 1 \ldots k$, pertencem à árvore multicast e, por conseqüência, possuem em algum nó abaixo deste ponto ao menos um receptor enviando pacotes de reconhecimento;
- no momento do procedimento de combinação, um pacote de reconhecimento chegou em cada um dos nós de entrada. Isso significa que todos os pacotes chegam no nó dentro de um intervalo $T$ igual ao período do temporizador inicializado com a chegada do primeiro dos pacotes de reconhecimento. Em outras palavras, se o primeiro pacote chegar em $t_0$ e o último em $t_{last}$, então e desigualdade $t_0 + T > t_{last}$ deve ser verdadeira;
- todos os pacotes de entrada possuem o máximo número de entradas $L$, que é igual ao número de camadas de vídeo transmitidas pela fonte. Considera-se também que as entradas de todos os pacotes são "não-compatíveis", isto é, elas não possuem taxas suficientemente próximas para serem automaticamente combinadas.

A primeira tarefa do algoritmo de combinação é a de criar o vetor temporário pela concatenação (e organização em ordem crescente) dos pacotes de entrada. Após esta etapa, o número de entradas do vetor temporário é

$$\hat{n}_e = kL. \tag{9}$$

Os pacotes de saída candidatos são obtidos pelo vetor resultante quando $\hat{n}_e - L$ entradas são descartadas do vetor temporário inicial. O número de pacotes candidatos é dado por

$$\gamma = \frac{(\hat{n}_e - 1)!}{(L-1)!(\hat{n}_e - L)!} = \frac{\prod_{i=1}^{kL+1}(kL-i)}{\prod_{i=1}^{L+1}(L-i)\prod_{i=L}^{kL+1}(kL-i)} = \frac{\prod_{i=1}^{L-1}(kL-i)}{\prod_{i=1}^{L+1}(L-i)} \tag{10}$$

A figura 2 mostra o comportamento de $\gamma$ para diferentes valores de $L$ e $k$. Observa-se que após um certo valor as influências de $L$ e $k$ sobre $\gamma$ são maiores. Entretanto, em situações práticas, $k$ e $L$ são relativamente pequenos. Em alguns codificadores de vídeo, o número de camadas gira em torno de três. De volta à figura 2, para $L = 3$ observa-se que $\gamma$ permanece dentro de uma faixa razoável mesmo para altos valores de $k$. Para $k = 15$ (isto é, o nó possui 15 enlaces de entrada), o número de pacotes de entrada é 1000. Além disso, o processamento do pacote de saída é baixo pois o algoritmo usa apenas operações dos tipos adição e comparação.

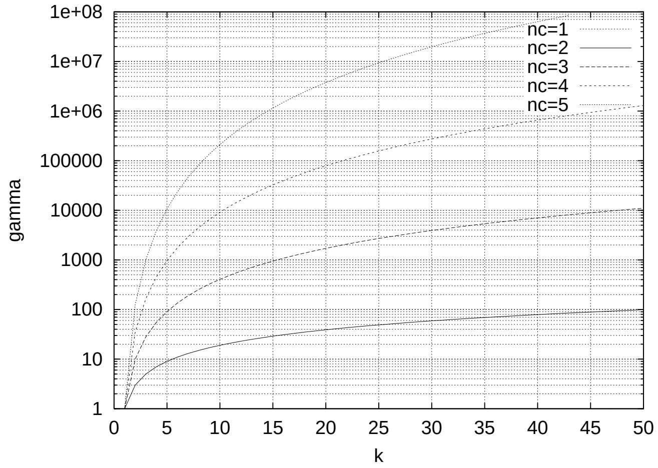



Figura 2: Como $\gamma$ responde a $k$ e $L$.

Mesmo se o número de pacotes candidatos parecer alto, pode-se utilizar um método híbrido realizado em duas fases. Durante a primeira, o procedimento de combinação utiliza o método clássico para uma primeira redução do número de entradas do vetor temporário. Quando um limite razoável (pré-estabelecido) for alcançado, o procedimento é chaveado para o Algoritmo Direto. Utilizando-se o método híbrido é possível manter a complexidade do sistema sob controle. A figura 3 mostra o número de pacotes candidatos para um faixa de valores de $L$ e $k$. Foi escolhido um limite de complexidade de forma a manter o número de pacotes candidatos abaixo de $\beta_{max} = 100$. Quando comparadas as curvas das figuras 2 e 3, nota-se que a limitação introduzida pelo algoritmo proposto é facilmente evitada pelo método híbrido.

# 3 Algoritmo com camadas virtuais

Dependendo do número de nós intermediários realizando o procedimento de combinação de pacotes, é possível que entradas que melhorariam a qualidade global do vídeo sejam indevidamente descartadas. Quando executado o algoritmo de combinação, os nós manipulam apenas os pacotes oriundos dos receptores localizados na sua sub-árvore correspondente. Receptores em outras sub-árvores não têm qualquer influência sobre os pacotes candidatos. Este é o preço de uma baixa sinalização.

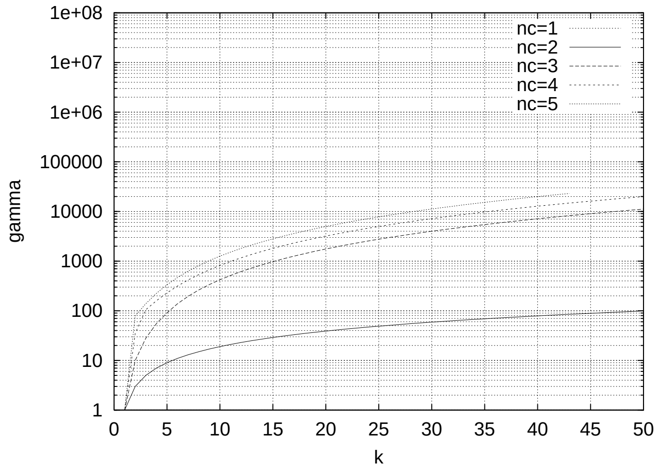



Figura 3: Como $\gamma$ responde a $k$ e $L$ quando o método híbrido é utilizado.

Em tal cenário "não-cooperativo", o esquema de camadas virtuais induz os nós intermediários a guardar estados extras da rede nos pacotes de reconhecimento. Os nós eliminam um número de entradas menor que o valor real que deveria ser eliminado. É utilizada uma variável $\lambda$ que substitui o número de camadas enviadas pela fonte. Assim, $\lambda > L$ e o número de entradas eliminadas é $c_T - \lambda$. Será visto na seqüência que o esquema com camadas virtuais reduz as limitações de sub-redes não-cooperativas.

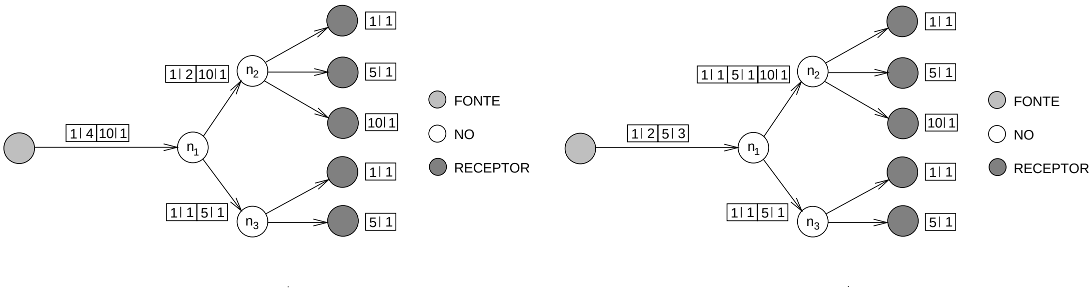
Seja um fragmento de rede onde $n_g$ nós são conectados ao nó $n_0$ e existem $k_{n_j}$ pacotes de entrada no nó $n_j$. Seja também $\beta_{p_i^j}$ o número de entradas no pacote $p_i^j$. O algoritmo é executado em todos os nós pertencentes à sessão multicast, ou seja:

$$\sum_{i=1}^{k_{n_j}} \beta_{p_i^j} > L, \quad \forall j. \tag{11}$$

$\gamma_{n_0}$ é a degradação global do vídeo no nó $n_0$ quando o pacote ótimo de saída para o nó $n_j, j = 1 \ldots k_{n_g}$, for $\mathbf{O}_{n_j}$. É possível que $\exists \hat{\mathbf{O}}_{n_j}, j = 1 \ldots k_{n_g}$, de forma que após a aplicação do esquema de camadas virtuais tenhamos $\hat{\gamma}_{n_0} > \gamma_{n_0}$.

A figura 4 descreve uma situação onde a fonte transmite duas camadas de vídeo. No caso normal, o nó intermediário $n_2$ descarta $c_T - L = 1$ entrada, que no caso é $e_2 = (5, 1)$. Naturalmente, esta informação é perdida e não pode ser recuperada nos próximos nós. O pacote que chaga ao nó $n_3$ também contém uma entrada com o mesmo valor $f = 5$. Se esta entrada não fosse eliminada pelo nó $n_2$, o algoritmo em $n_1$ não a eliminaria, mas sim a entrada correspondente a $f = 10$.

O esquema de camadas virtuais reduz a probabilidade que este tipo de inconsistência aconteça. No exemplo acima, para $\lambda = 3$ o algoritmo de combinação não é executado no nó $n_2$. Conseqüentemente, a entrada associada aos receptores que desejam receber $f = 5$ não é eliminada. No nó $n_1$, esta informação ainda existe a entrada eliminada é a equivalente a $f = 10$, ou seja, aquela que resulta na mais alta qualidade do vídeo.




(a) Sem camadas virtuais. (b) Utilizando-se uma camada virtual.

Figura 4: Evolução do procedimento de combinação.

# 4 Análise

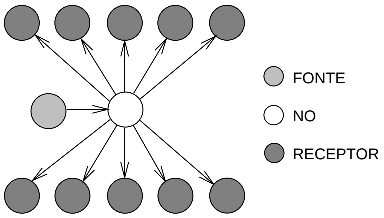
Esta seção analisa o Algoritmo Direto e compara seu desempenho com o do método clássico. É avaliado o nível de degradação do vídeo nos receptores. São utilizados dois tipos de topologia. O primeiro é em modo denso, onde um grande número de receptores são conectados a um único nó. Em tal cenário, cada procedimento de eliminação deve descartar um número elevado de entradas. Este tipo de topologia visa testar a eficiência do Algoritmo Direto. O segundo tipo de topologia consiste em uma árvore binária que emula uma rede em modo esparso. O objetivo é testar como o Algoritmo Direto responde a cenários em cascata e também verificar que o uso do esquema em camadas virtuais é indicado em tais casos.

A medida utilizada para avaliar a eficiência do algoritmo é a degradação do vídeo (ou, inversamente, a qualidade global do vídeo). Para cada receptor na sessão multicast é computada a diferença entre a taxa de vídeo requerida e a taxa recebida (ver detalhes na seção 2). É também avaliado o compromisso entre o número de camadas e o nível de satisfação dos usuários na sessão multicast.

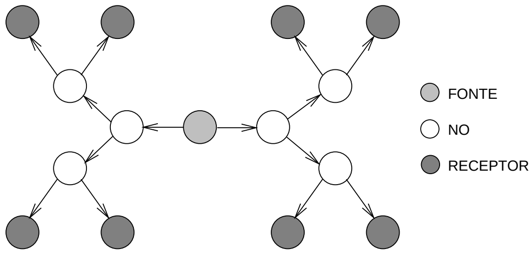
Para a rede em modo denso, foi utilizada a topologia da figura 5(a). Esta topologia consiste em uma fonte e dez receptores conectados através um nó. Nas simulações, todos os enlaces possuem a mesma capacidade $C = 1$ Mbps e atrasos idênticos $\tau = 10$ ms. Ao se utilisar o mesmo atraso para todos os enlaces, torna-se a análise independente dos parâmetros que não seja a banda-passante disponível. São introduzidas nos enlaces fontes de tráfego exponenciais para representar tráfegos de fundo.

A figura 6(a) mostra a degradação do vídeo para a topologia 1. O Algoritmo Direto sempre resulta na mais baixa degradação quando comparado com o método clássico. É interessante observar que a razão média entre os dois métodos é $\sim$25%, o que é excelente para uma rede relativamente pequena com 10 receptores. Além disso, em tal cenário o Algoritmo Direto resulta no limite empírico, ou seja, no melhor resultado possível. Isto era esperado, visto que para a rede inteira o algoritmo de combinação é executado apenas uma vez. Nas redes onde os receptores são distribuídos de maneira densa, quanto menor for a profundidade da árvore multicast, mais próximo do limite empírico é o resultado obtido pelo Algoritmo Direto. De qualquer forma, o Algoritmo Direto é sempre mais eficiente que qualquer outro método em redes do tipo não-cooperativas.

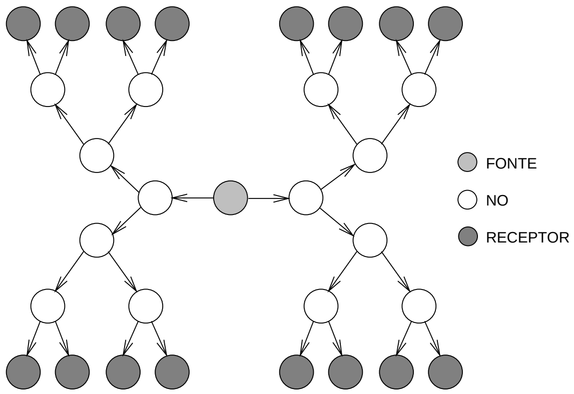
A mesma análise realizada acima foi feita sobre a topologia em modo esparso com 8




(a) Topologia 1.

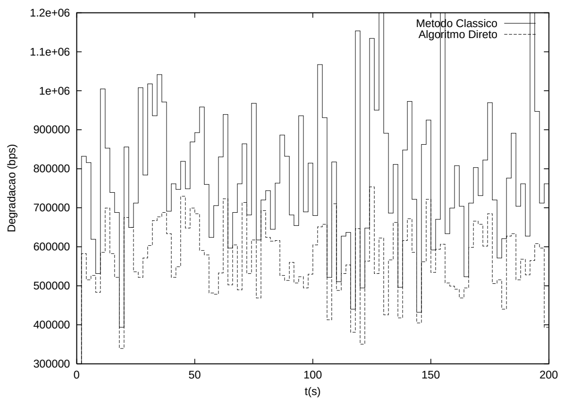



(b) Topologia 2.

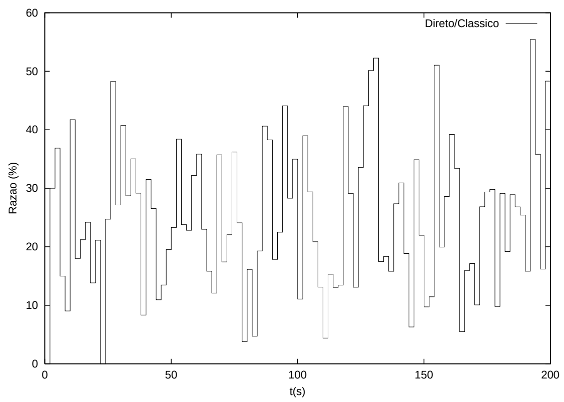



(c) Topologia 3.

Figura 5: Topologias de simulação.



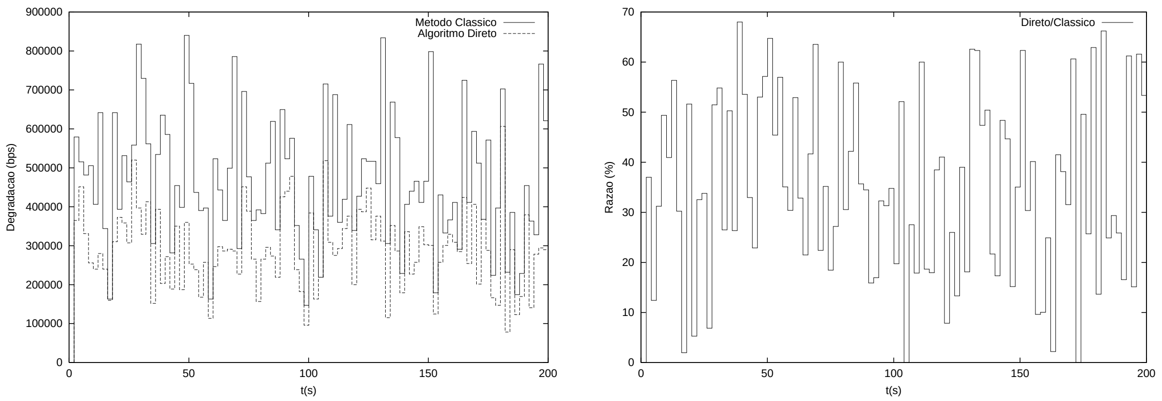

(a) Degradação global do vídeo. Os resultado obtidos pelo Algoritmo Direto são equivalentes ao limite empírico.




(b) Razão entre o Algoritmo Direto e o método clássico.

Figura 6: Resultados para a topologia 1.

receptores, mostrada na figura 5(b). Os parâmentros de simulação são os mesmo utilizados para a topologia 1. As degradações globais do vídeo para o Algoritmo Direto e para o método clássico estão representadas na figura 7(a). Neste caso, os resultados obtidos pelos dois métodos são mais próximos do que para a topologia 1. Isto se deve ao fato do número de entradas descartadas em cada *loop* ser inferior ao primeiro caso. No exemplo da topologia 1, são eliminadas $c_T - L = 10 - 3 = 7$ entradas. Ao executar esta operação, a probabilidade do método clássico levar a valores menos eficientes se deve ao fato do algoritmo ser executado em vários passos consecutivos. No caso da topologia 2, o procedimento de combinação é distribuído em diferentes nós e cada um deles elimina no máximo $c_T - L = 6 - 3 = 3$ entradas. Assim como para a topologia 1, a figura 7(b) mostra a razão entre o Algoritmo Direto e o método clássico. Mesmo neste caso, o Algoritmo Direto é cerca de 35% mais eficiente que o método clássico em pelo menos 50% do tempo.



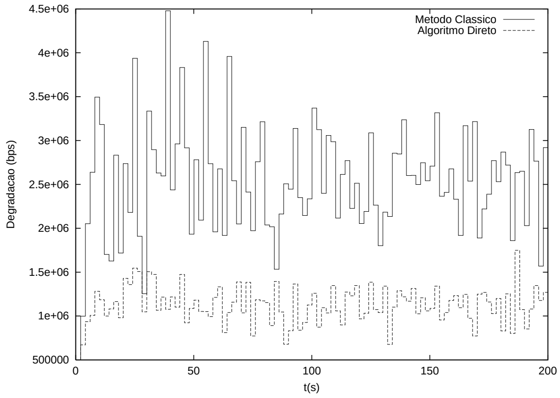

(a) Degradação global do vídeo.

(b) Razão entre o Algoritmo Direto e o método clássico.

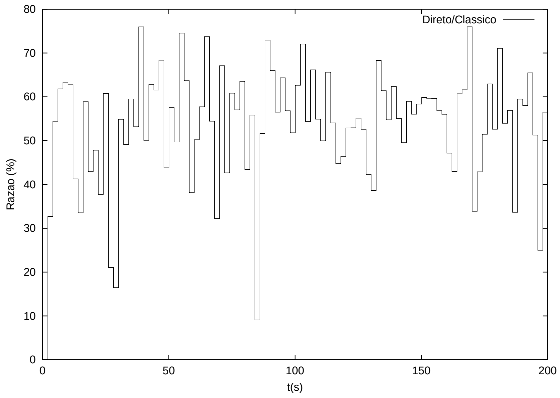
Figura 7: Resultados para a topologia 2.

A próxima análise é voltada para um estudo mais profundo das topologias em modo esparso. A rede simulada está representada na figura 5(c), que é exatamente a mesma da figura 5(b) com a diferença de possuir 16 receptores (cinco níveis na árvore). Mais uma vez os parâmetros de simulação são os mesmos. O máximo número de entradas descartadas, assim como para a topologia 2, é 3 e o número de passos consecutivos do método clássico é no máximo 3. A diferença é unicamente o grau da árvore. É provável então que as vantagens obtidas pelo Algoritmo Direto sejam ainda mais acentuadas. As figuras 8(a) e 8(b) mostram que essa vantagem realmente existe e que um aumento igual a 1 na profundidade da árvore levam a níveis de degradação de qualidade do vídeo de cerca de 55% melhores para o Algoritmo Direto.

Será avaliado na seqüência o desempenho do Algoritmo Direto com camadas virtuais. Para tal foi utilizada a topologia 3 da figura 5(c). A figura 9(a) mostra os resultados para $\lambda = 4, 5$ e 6. O número de camadas de vídeo transmitidas pela fonte é 3. Assim como para as demais simulações realizadas até então, também é mostrada a relação entre o Algoritmo Direto e o método clássico para $\lambda = 6$. Os resultados mostram que uma simples manipulação de variáveis melhora a qualidade global do vídeo em cerca de 60%.

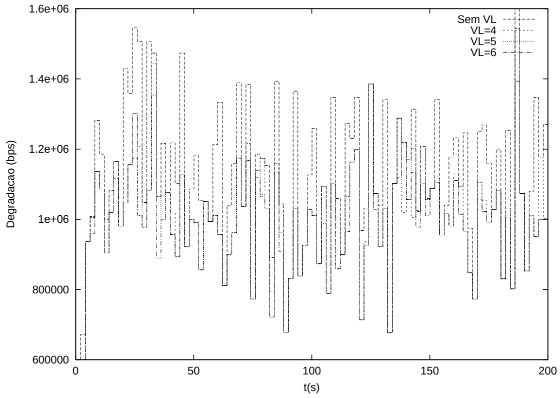



(a) Degradação global do vídeo.

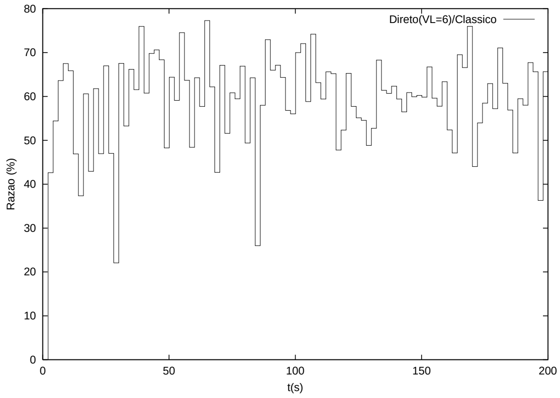



(b) Razão entre o Algoritmo Direto e o método clássico.

Figura 8: Resultados para a topologia 3.




(a) Degradação global do vídeo com camadas virtuais (VL).




(b) Razão entre o Algoritmo Direto com camadas virtuais (VL) e o método clássico.

Figura 9: Resultados para a topologia 1 com camadas virtuais.

# 5 Conclusões

Este artigo tratou da otimização do número de informações de reconhecimento enviadas pelos receptores em uma sessão multicast de forma a evitar a implosão de pacotes de controle na fonte. Além disso, o sistema busca sempre obter a melhor qualidade do vídeo nos receptores. O uso de pacotes de reconhecimento permite que a fonte calcule as camadas de vídeo de forma a servir os receptores com as melhores taxas possíveis. Foi proposto o Algoritmo Direto que combina de uma maneira eficiente os pacotes de reconhecimento nos nós intermediários de forma a se alcançar a menor degradação do vídeo nos receptores. Em primeiro lugar, o algoritmo de combinação de pacotes é realizado em apenas um *loop*, o que o torna consciente de todos os pacotes candidatos a serem transmitidos à fonte. De forma contrária ao método clássico, onde o algoritmo é executado em passos consecutivos, o Algoritmo Direto *sempre* resulta nos melhores valores possíveis em redes não-cooperativas. A propósito, para se evitar o descarte de informações relevantes, o Algoritmo Direto ainda pode usar o conceito de camadas virtuais.

O Algoritmo Direto foi analisado e simulado para dois tipos de topologia. A primeira, em modo denso, onde os nós intermediário podem ter um número alto de enlaces de entrada, e a segunda, em modo esparso, onde os receptores são bem distribuídos na rede. Os resultados mostram que o uso do esquema de camadas virtuais reduz o problema de não cooperação entre sub-redes. Nas redes em modo denso, as vantagens do uso do Algoritmo Direto são ainda mais acentuadas porque o algoritmo aproxima o comportamento do limite empírico, que corresponde ao melhor resultado possível.

## Referências


[1] B. J. Vickers, C. Albuquerque e T. Suda, "Source-adaptive multi-layered multicast algorithms for real-time video distribution", *IEEE/ACM Transactions on Networking*, dezembro de 2000.

[2] X. Li, M. H. Ammar e S. Paul, "Video multicast over the internet", *IEEE Network Magazine*, março de 1999.

[3] M. D. de Amorim, O. C. M. B. Duarte e G. Pujolle, "Multi-criteria arguments for improving the fairness of layered multicast applications", in *Lecture Notes in Computer Science 1815*, Paris, France, pp. 1–10, maio de 2000.

[4] J. Nonnenmacher e E. W. Biersack, "Scalable feedback for large groups", *IEEE Transactions on Networking*, vol. 7, no. 3, junho de 1999.

[5] D. Saparilla e K. W. Ross, "Optimal streaming of layered video", in *IEEE Infocom*, Tel-Aviv, Israel, março de 2000.

[6] S. Sarkar e L. Tassiulas, "Distributed algorithms for computation of fair rates in multirate multicast trees", in *IEEE Infocom*, Tel-Aviv, Israel, março de 2000.

[7] D. Rubenstein, J. Kurose e D. Towsley, "The impact of multicast layering on network fairness", in *ACM Sigcomm*, Cambridge, Massachusets, USA, setembro de 1999.

[8] L. H. M. K. Costa, S. Fdida e O. C. M. B. Duarte, "Hop-by-hop multicast routing protocol", tech. rep., LiP6 – UPMC, novembro de 2000. URL: http://www-rp.lip6.fr/publications/costa/hph.ps.gz.

[9] J. Moy, "Multicast Extensions to OSPF". RFC 1584, maio de 1994.

[10] B. J. Vickers, C. Albuquerque e T. Suda, "Adaptive multicast of multi-layered video: Rate-based and credit-based approaches", in *IEEE Infocom*, San Francisco, CA, USA, março de 1998.