

# Um Serviço de Distribuição de Vídeo Sob Demanda Baseado em uma Rede de Servidores D-VoD

Tiago Salmito<sup>1</sup>      João Paulo Farias<sup>1</sup>      Gledson Elias<sup>1</sup>      Guido Lemos      Luiz Leite  
mahatma@natalnet.br      jpfarias@natalnet.br      gledson@dimap.ufrn.br      guido@di.ufpb.br      leduardo@natalnet.br

<sup>1</sup> Laboratório NatalNet – Departamento de Informática e Matemática Aplicada – DIMAp  
Universidade Federal do Rio Grande do Norte – UFRN

***Resumo.** Este artigo apresenta a arquitetura de um serviço de distribuição de vídeo e a implementação de servidores que dão suporte a arquitetura proposta. O serviço foi concebido para distribuir vídeo em rede de vídeo digital composta por servidores (equipamentos e software) interligados por uma rede IP. O serviço foi testado no backbone da RNP, mostrando que é viável distribuir vídeo de alta qualidade em ambientes com restrição de banda passante.*

## 1. Introdução

Este trabalho apresenta a arquitetura de um serviço de distribuição de vídeo baseado em uma rede de servidores D-VoD implementados e testados pela equipe do GTVD-RNP. Embora aspectos de gerenciamento e segurança sejam igualmente importantes para a funcionalidade do sistema como um todo, este artigo não detalha tais aspectos. A entrega eficiente do fluxo de vídeo está no centro das atenções de um sistema de vídeo sob demanda distribuído. O Grupo de Trabalho de Vídeo Digital (GTVD) é uma iniciativa da RNP que visa induzir o desenvolvimento de uma nova geração de aplicações de vídeo digital para explorar o potencial de redes de alta velocidade no país.

Como aplicação piloto foi proposta uma infra-estrutura baseada na RNP para implantar um serviço de vídeo sob demanda com suporte a replicação de servidores e otimização da utilização dos recursos de rede.

## 2. Arquitetura do Serviço de Vídeo Sob Demanda

A arquitetura proposta para o serviço de distribuição de vídeo baseia-se em uma estrutura de múltiplos níveis de distribuição: servidor-proxy, proxy-proxy e proxy-cliente. Vale ressaltar que podem existir diversos níveis de distribuição proxy-proxy. A Figura 1 ilustra a arquitetura de distribuição de vídeo.

Os servidores de vídeo compõem o repositório permanente de vídeo digital. A idéia é que as instituições colaboradoras responsáveis pelas aplicações usuárias do serviço, mantenham seus próprios servidores, que armazenarão apenas os vídeos disponibilizados pela respectiva instituição no serviço. Opcionalmente, para incrementar os requisitos de tolerância à falhas, uma instituição poderá utilizar diversos servidores de vídeo, onde cada vídeo poderá ser armazenado em um subconjunto qualquer destes servidores. Ou seja, a arquitetura é baseada em um esquema de replicação parcial, onde os servidores não precisam estar sincronizados em todo o seu conteúdo, mas apenas nas réplicas de vídeo que coexistem simultaneamente em alguns deles.

Para satisfazer aos requisitos da arquitetura proposta, foi realizada uma implementação modular composta de diversos componentes para tratar a diversidade de fontes e destinos de vídeos, e um único componente de controle. Em tempo de execução, é possível acoplar novos tipos de fontes e destinos, viabilizando assim uma implementação única para servidores e proxies de vídeo.

A Figura 2 ilustra a seqüência de ações executada quando um cliente acessa o serviço. Inicialmente o cliente acessa via Internet uma aplicação de busca, que consulta um repositório de metadados dos vídeos disponibilizados pela instituição responsável. Os metadados são todas as informações sobre os vídeos e os endereços dos servidores fonte. Toda aplicação de busca deverá retornar os seguintes elementos: o identificador do vídeo desejado, os endereços dos servidores fonte do vídeo, e o endereço do serviço de gerenciamento.

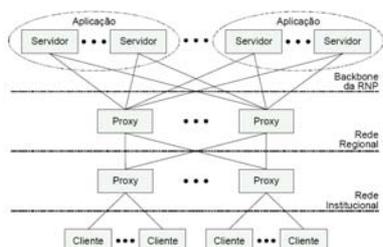


Figura 1. Arquitetura de distribuição

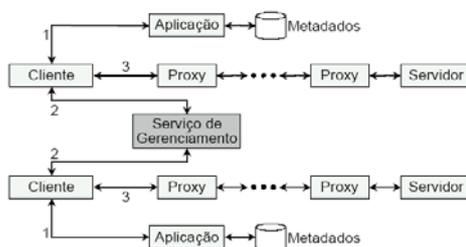


Figura 2 – Seqüência de Acesso

Uma vez recuperado o identificador do vídeo, o cliente então envia uma requisição ao serviço de gerenciamento, informando o identificador do vídeo desejado e os endereços dos servidores fonte daquele vídeo. O serviço de gerenciamento é responsável por selecionar um dos servidores disponíveis, juntamente com a cadeia de proxies, que serão utilizados para atender a solicitação do cliente. Como resultado, o serviço de gerenciamento retorna a melhor rota de distribuição (servidor e proxies) a ser utilizada para distribuir o vídeo até o cliente.

Em seguida, o cliente envia uma requisição do vídeo desejado ao primeiro proxy da rota de distribuição. Por sua vez, o proxy requisitado solicita o vídeo ao próximo proxy da rota de distribuição, até que a requisição chega ao servidor fonte do vídeo. Neste momento, a distribuição é iniciada e o cliente começa a receber o fluxo de vídeo.

### 3. Arquitetura do Servidor D-VoD

O D-VoD (*Distributed Video on Demand*) é um servidor de vídeo sob demanda que foi desenvolvido para atender os requisitos determinados pela arquitetura do sistema de vídeo sob demanda GTVD. Mesmo sendo o serviço independente do servidor de vídeo, este servidor foi desenvolvido para ser possível a transmissão distribuída de conteúdo através de um esquema de *caches* hierárquicas com o controle do gerente do serviço VoD.

A arquitetura do D-VoD, como visto na Figura 3, está definida em três camadas. A primeira camada é constituída por módulos especializados em recuperar dados de alguma forma, seja localmente ou remotamente, de acordo com a demanda do servidor. Estes são chamados módulos Fonte, pois atuam como fontes de dados.

A camada de controle gerencia a arquitetura modular, controlando quais módulos fontes estarão recuperando dados para quais destinos. Ela também se preocupa com a recuperação das estatísticas do servidor e com os meios de funcionamento dos módulos fonte e destino. Nessa camada também se encontram programas de funções específicas que podem ser acoplados ao D-VoD como módulos de controle. Existe também um módulo especial chamado Gerente, responsável pelo gerenciamento remoto do servidor, seja automático ou manual (por uma interface HTTP ou uma GUI), e pela divulgação das estatísticas recuperadas.

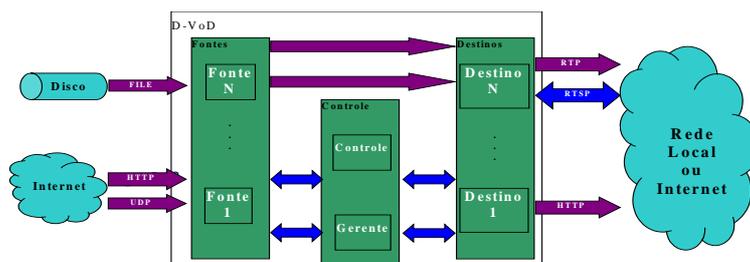


Figura 3. Arquitetura conceitual do D-VoD

A terceira camada é responsável por ser a interface entre os clientes e as fontes de dados. Ela é constituída por módulos especiais que implementam um ou mais protocolos de transporte e controle para transmitir os dados recuperados diretamente pelas fontes. Portanto esses módulos são chamados de módulos de destino.

O paradigma de programação orientada a objetos se mostra perfeitamente apto para a modelagem da arquitetura do D-VoD. Portanto este paradigma foi usado para modelar a arquitetura conceitual do servidor. A Figura 4 mostra a especificação da modelagem apresentada através da notação UML.

## 4. Módulos Fonte

Os módulos fonte são classes que implementam os métodos definidos pela interface da classe de fachada *Fonte*, para a recuperação de dados, seja em seu espaço de armazenamento local, ou de um dispositivo de captura de vídeo em tempo real, ou de um outro servidor da rede, para um espaço de memória do servidor, que é então repassado ao módulo de destino que está necessitando do dado naquele instante. Cada fonte de dados do D-VoD deve estender a classe *Fonte* e implementar seus métodos virtuais *Criar*, *Abrir*, *Posicionar*, *Ler* e *Fechar*.

### 4.1. Fonte Cache

A característica distribuída do serviço VoD prevê a necessidade de projeto e implementação de servidores atuando como *proxy*, cuja principal característica é a presença de uma *cache* de armazenamento temporário. Nesses servidores, os vídeos são guardados apenas temporariamente nessa *cache* em disco e, quando um vídeo que não existe na *cache* local é solicitado a um servidor *proxy*, a própria *cache* é responsável por buscar este vídeo de outro servidor fonte ou mesmo de outros servidores *proxy* em uma rede de distribuição hierárquica.

A política adotada pelo D-VoD foi a de remoção de blocos que foram acessados menos recentemente, ou LRU (*least recently used*) por sua simplicidade e eficiência em aplicações de vídeo sob demanda. Essa política consiste em excluir o último bloco

presente na *cache* do vídeo que foi acessado há mais tempo, isto é, a política LRU só é aplicada para determinar o objeto (vídeo) que foi acessado há mais tempo, e não o bloco acessado há mais tempo. Os blocos que pertencem ao mesmo vídeo são tratados diferentemente da política LRU, sendo escolhido o bloco que foi acessado mais recentemente.

## 5. Módulos de Destino

Os módulos de destino são módulos que implementam os métodos da classe *Destino*. Os módulos de destino implementam um ou mais protocolos de transporte ou controle de dados, necessitando de uma fonte geradora de dados do tipo *Fonte*.

A escolha dos protocolos implementados no D-VoD baseou-se no requisito compatibilidade com os exibidores de vídeos disponíveis no mercado. Foi implementado o protocolo HTTP, que é o protocolo de transferência de dados mais utilizado na *web*. Foi também implementado um protocolo mais específico para a transmissão de dados em tempo real, ou *Streaming* de dados, o RTSP ou “*Real Time Streaming Protocol*”. Desta forma o D-VoD pode enviar os dados recuperados pelas fontes usando os protocolos de transporte TCP, por HTTP, e UDP e RTP por RTSP.

## 6. Resultados Obtidos

O sistema de VoD do GTVD foi testado na infraestrutura de rede da RNP. Neste cenário de testes foi implementado um motor de busca em JSP que procura em uma base de dados LDAP. Para distribuição dos vídeos foram utilizados sete servidores D-VoD configurados como *Proxy* e dois configurados como servidores fonte distribuídos pelos vários pontos de presença (PoP) da rede RNP no Brasil, como ilustrado na Figura



5.

Figura 5. D-VoDs na RNP

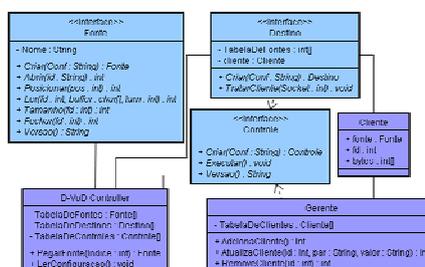


Figura 4 – Diagrama UML do D-VoD

Esta Rede de Vídeo digital montada na rede institucional da RNP está operacional e os testes realizados demonstram que o serviço é capaz de distribuir vídeos de alta qualidade mesmo considerando restrições nos enlaces interligando os servidores D-VoD.

## 7. Referências

[1] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext transfer protocol - HTTP/1.1", RFC 2068, Janeiro 1997.  
 [2] Schulzrinne, H., Gettys, J., Rao, A., Lanphier, R., "Real Time Streaming Protocol - RTSP/1.0", RFC 2326, Abril 1998.