Comunicação Multiponto Móvel baseada em Redes Ativas

Flavio Paiva Junqueira — flavio@gta.ufrj.br
Otto Carlos M. B. Duarte — otto@gta.ufrj.br

Grupo de Teleinformática e Automação - GTA
COPPE/EE - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
C. P. 68504 - CEP 21945-970 - Rio de Janeiro - Brasil

Resumo

Neste artigo é proposto um modelo ativo de comunicação multiponto confiável e escalável para um serviço destinado a estações móveis. Para este modelo, é definido um protocolo ativo, baseado nos conceitos de redes ativas que promovem um serviço de rede flexível e adequado às necessidades das aplicações. Neste protocolo, a adesão das estações móveis é feita diretamente à árvore de distribuição, sem a existência de agentes intermediários. As principais vantagens encontradas nesta abordagem são a diminuição do atraso na difusão de mensagens e do tráfego de informação de gerenciamento de mobilidade.

Palavras-chave: redes ativas, protocolo, mobilidade, multiponto.

Abstract

This paper presents a scalable reliable multipoint communication active model for mobile hosts. An active protocol is defined for this model, based on the active networks concepts that provide a flexible and suitable network service to applications. In this protocol, the mobile host joins directly to the distribution tree, without intermediate agents. This strategy reduces the messages' delay from senders to receivers and the mobility management information traffic.

Keywords: active networking, protocol, mobility, multipoint.

1 Introdução

Atualmente existe uma grande variedade de aplicações distribuídas como trabalho cooperativo, simulação interativa distribuída (DIS - *Distributed Interactive Simulation*), teleconferência etc, que demandam os mais diversos requisitos do sistema de comunicação. Com a tecnologia atualmente empregada, utilizam-se diferentes pilhas de protocolos para diferentes serviços, o que acarreta na necessidade de uma variedade de pilhas. Para não haver um grande número destas pilhas, as aplicações são divididas em classes de serviço. No entanto, para atender a novas aplicações que surgem com novos requisitos, novas pilhas de protocolos devem ser criadas e novas classes definidas. Desta forma, este processo de criação e definição tende a sobrecarregar a rede em termos do número de serviços oferecidos. Além disso, o tempo gasto hoje para alteração nos protocolos e nas entidades do sistema de comunicação não é suficiente para atender as necessidades destas aplicações emergentes, de forma que a existência de um serviço de rede ajustável aos requisitos das aplicações torna-se necessário [1].

A partir desta observação, surgiu a linha de pesquisa denominada de redes ativas. Em uma rede ativa, apenas um serviço é oferecido. Este serviço é flexível, pois é realizado através de computações específicas na própria rede, necessárias a uma dada aplicação [2]. Este conceito provém das denominadas redes inteligentes para telecomunicações, onde a implantação de novos serviços de forma eficiente, rápida e de baixo custo é conseguida através da separação entre o controle do serviço e o controle da transmissão [3, 4].

A forma como computações específicas são inseridas na rede para atuar sobre o fluxo de informação de uma dada aplicação varia entre os diversos modelos presentes nos trabalhos de pesquisa na área. O modelo discreto separa os mecanismos de injeção de programas em um nó programável, do processamento dos pacotes de dados que fluem pelo nó. Os programas nos nós são disparados por informações contidas nos cabeçalhos dos pacotes de dados. Zegura *et al* [5, 6, 7, 8] utilizam este modelo em redes ATM para a solução do problema de congestionamento nos nós. Durante a ocorrência de um congestionamento, os comutadores consultam o cabeçalho das células, que indicam um procedimento a ser executado. A finalidade deste procedimento é diminuir o número de células para transmissão de acordo com a necessidade da aplicação. Como parte do projeto *Switchware* do departamento de ciência da computação da universidade da Pensilvânia, Smith *et al* [4, 9] propõem componentes de rede como pontes e comutadores que reconhecem nas suas portas de entrada mensagens denominadas *Switchlets*, contendo código para programar estes componentes.

Em oposição ao modelo discreto, o modelo integrado propõe a substituição de pacotes contendo dados por pacotes contendo código para execução e dados. Em uma arquitetura para redes ativas baseada neste modelo é possível que os procedimentos carregados pelos pacotes tomem decisões a respeito do seu destino, como por exemplo, o pacote pode: ser replicado para transmissão em múltiplos enlaces, unir-se a outros pacotes para diminuir a quantidade total de dados enviados e até mesmo optar pelo descarte. Yemini *et al* propõe uma rede programável com a linguagem *Netscript*, voltada para a gerência de redes e baseada na troca de agentes móveis [10]. Nesta proposta, o procedimento a ser executado que é carregado pelo agente é executado somente em nós determinados pelo próprio procedimento, fornecendo suporte para migração. Agregado ao projeto *Switchware*, Gunter *et al* [11] estão desenvolvendo uma linguagem de Script para programação a nível de pacotes denominada PLAN (*Programming Language for Active Networks*). Os recursos disponíveis para a programação nesta linguagem são limitados, mas adequados às necessidades da rede, tornando-a própria para uma execução rápida. O projeto *Liquid Software* [12] tem como objetivo o desenvolvimento de novas tecnologias de compilação para facilitar o uso de código móvel na rede. Tennenhouse *et al* [2, 13, 14] denominam pacotes contendo código e dados de cápsulas. Utilizando o conceito de cápsulas, esta equipe desenvolveu o sistema ANTS (*Active Networking Toolkit System*) a partir do qual é possível o desenvolvimento de protocolos ativos. O desenvolvimento do trabalho proposto neste artigo está baseado neste sistema.

O modelo integrado permite que cada aplicação desenvolva seu próprio serviço de rede, o que fornece um alto de grau de flexibilidade em termos do atendimento aos requisitos de cada aplicação. Já o modelo discreto, permite apenas que algum tipo de supervisor acrescente um novo serviço de rede e que os usuários utilizem-no. Apesar de não promover um alto grau de flexibilidade, esta abordagem possui menos problemas de segurança do que o modelo integrado. As questões voltadas à segurança não estão no escopo deste trabalho.

Desta forma, este artigo propõe um modelo ativo para comunicação multiponto móvel, utilizando os recursos disponibilizados pelo modelo integrado de redes ativas. O intuito deste modelo ativo é melhorar este tipo de comunicação em termos de atraso e do número de entidades de comunicação participantes no gerenciamento da informação de mobilidade, necessários nas propostas atuais para IP móvel. O caso em foco é o de distribuição de dados, no qual uma estação móvel necessita receber informações constantemente de uma dada árvore de distribuição enquanto muda sua posição no decorrer do tempo. Este caso se aplica, por exemplo, na distribuição de dados do mercado financeiro mundial.

Este artigo está estruturado da seguinte forma. A seção 2 apresenta as pricipais características de uma arquitetura de rede baseada no modelo integrado. Nesta seção é apresentado ainda, o sistema ANTS. Os principais conceitos que envolvem a comunicação multiponto e móvel são colocados na seção 3. O protocolo ativo, com as definições das cápsulas e seus algoritmos, é apresentado na seção 4. A descrição detalhada do modelo de comunicação proposto neste artigo é feita na seção 5. Por último, na seção 6 algumas conclusões são realizadas.

2 Redes ativas - o modelo integrado

No modelo integrado baseado nas propostas de Tennenhouse *et al*, as computações específicas de cada aplicação realizadas nos nós da rede correspondem a execução dos programas (ou também chamados de fragmentos de código) contidos nas cápsulas. No entanto, a execução deste programas só é possível se houver um acordo com relação ao modelo computacional adotado por todos os nós ativos da rede [15]. Neste contexto, define-se o modelo computacional como sendo o conjunto de instruções e recursos disponíveis para os fragmentos de código contidos nas cápsulas [13].

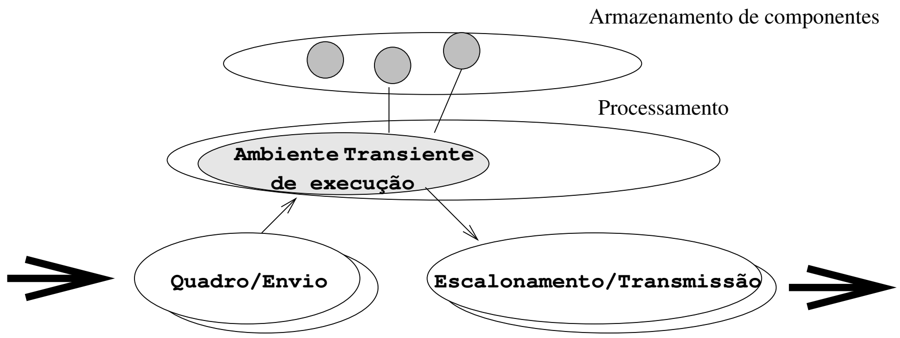

Figura 1: Organização de um nó ativo

Uma visão conceitual da organização de um nó ativo é apresentada na figura 1. Ao ser recebida pelo nível de enlace, o conteúdo da cápsula é enviado para um ambiente transiente de execução, onde pode ser avaliado de forma segura. Supõe-se que os programas são compostos de instruções primitivas que realizam computações básicas no conteúdo de cápsulas ou tomam decisões a respeito de seus destinos, isto é, cápsulas podem: ser replicadas para envio em mais de um enlace de saída, aguardar a chegada de outras cápsulas (*rendez-vous*), se mesclar com outras cápsulas, ser descartadas etc. É possível, ainda, que os programas invoquem métodos externos que forneçam acesso a recursos fora do ambiente transiente. Sendo assim, a execução pode resultar no escalonamento de zero ou mais cápsulas e é possível que seja deixado algum tipo de informação em um ambiente não-transiente para as próximas cápsulas do fluxo ao qual a cápsula, em execução, pertence. O término da execução resulta na finalização do ambiente transiente, sendo reutilizado para a execução de outra cápsula.

Durante o desenvolvimento do modelo computacional, decisões a respeito do acesso a recursos fora do ambiente transiente de execução devem ser tomadas. Em uma abordagem restrita, os programas em cápsulas não teriam acesso a recursos externos ao ambiente transiente. Embora isto seja suficiente para o propósito de diversas aplicações, outras requerem acesso a recursos além do ambiente transiente [13]. Programas podem acessar recursos externos das seguintes formas:

- **componentes de fundação**: os componentes de fundação são métodos que implementam o acesso controlado dos programas aos recursos disponíveis nos nós. Estes métodos estão disponíveis universalmente e são implementados fora do ambiente transiente. Para cada cápsula, apenas um subconjunto destes métodos estará disponível, dependendo dos privilégios concedidos por um mecanismo de autenticação. Estes métodos existem para promover acesso a tabelas de roteamento e de estados dos enlaces de transmissão ligados ao nó, se for previsto roteamento com qualidade de serviço (QoS) [15];

- **armazenamento ativo**: ao término da execução de uma cápsula, o espaço de memória ocupado pelo ambiente transiente de execução é reutilizado para dar lugar à execução das próximas cápsulas, e nenhuma informação é guardada por este ambiente. O espaço de armazenamento ativo é um espaço não-transiente, onde uma cápsula pertencente a um dado fluxo pode consultar e atualizar o estado deixado pelas cápsulas anteriores do seu fluxo. Este estado permanece armazenado temporariamente, e, por isso, deve ser constantemente atualizado. O nome dado a este estado é *soft-state*;

- **extensões**: as extensões permitem a criação de novos métodos e classes nos nós para que possam ser utilizados por outras cápsulas. Desta forma, diminui-se o código carregado pelas cápsulas.

Outra decisão importante a ser tomada durante o desenvolvimento do modelo computacional diz respeito ao tipo de codificação dos programas em cápsulas. A codificação pode ser realizada através de código fonte, intermediário ou objeto. A utilização de código fonte implica na interpretação do programa em tempo de execução. As principais vantagens deste tipo de codificação são a facilidade de implementação dos programas em cápsulas e a portabilidade. Por outro lado, a interpretação em tempo de execução acarreta em perda de desempenho. Outro tipo ainda, é o código intermediário que é interpretado também, mas possui a vantagem de ter as fases de pré-processamento, análise léxica e checagem estática feitas em tempo de compilação. Com isto, mantém-se a portabilidade e ganha-se no desempenho em tempo de execução. O último tipo é o código de máquina, onde os programas em cápsulas estão em código objeto particular de uma arquitetura. É o melhor em termos de desempenho, mas perde-se na portabilidade.

Dentre os três tipos de codificação, o código intermediário é o mais equilibrado em termos das vantagens e desvantagens citadas. Diversas pesquisas estão em andamento atualmente para aprimorar o processo de execução para este tipo de código. A utilização de compilação *Just-in-Time*, onde a conversão de código intermediário é feita em tempo de execução e sob demanda, é uma proposta para ganho de desempenho. Com isto, o custo de compilação é distribuído pelo tempo de execução do programa [12].

Portanto, a definição do modelo computacional é essencial para o desenvolvimento de uma arquitetura para redes ativas baseada no modelo integrado. Decisões devem ser tomadas acerca do acesso a recursos externos ao ambiente transiente de execução, do tipo de codificação e de qual linguagem será utilizada.

2.1 O sistema ANTS

O sistema ANTS foi desenvolvido por Wetherall *et al*, pertencentes ao grupo SDS (*Software Devices and Systems*) do MIT. O serviço de rede oferecido pelo sistema ANTS não é fixo, e sim flexível, pois diferentes aplicações são capazes de introduzir novos protocolos na rede através da especificação de rotinas a serem executadas nos nós da rede que transmitem as suas mensagens [1].

Com esta finalidade, o sistema ANTS, substitui pacotes convencionais por cápsulas, roteadores e estações por nós ativos. Além disto, implementa um mecanismo de distribuição de código que assegura que rotinas de processamento sejam transferidas de forma automática e dinâmica para os nós onde são necessárias.

As rotinas de processamento executadas pelas cápsulas são agrupadas nos nós ativos em protocolos. Cada protocolo está associado a um padrão de comportamento que define o processamento que deve ocorrer por toda a rede ativa de acordo com a dinâmica da aplicação.

Um protocolo ativo, pela definição do sistema, é dividido em grupos de código, que são uma coleção de tipos de cápsulas relacionadas, cujas rotinas de processamento são transferidas como uma unidade pelo sistema de distribuição de código. Cápsulas, como já foi visto anteriormente, substituem pacotes convencionais, onde a principal diferença é a inclusão de uma referência para a rotina de processamento utilizada para processá-la nos nós ativos. A relação entre estas entidades é apresentada na figura 2.

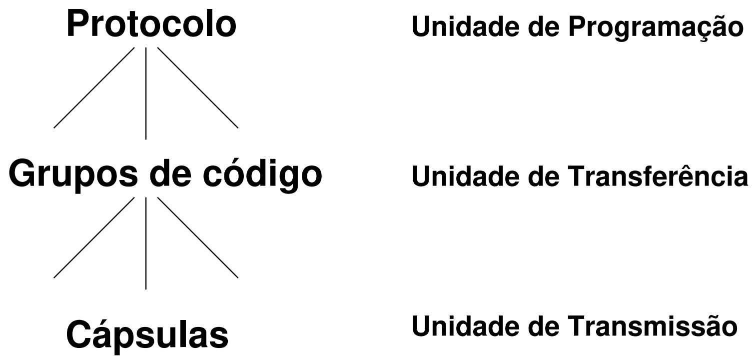

Figura 2: Hierarquia de composição de cápsulas

A execução das rotinas de processamento definidas por uma aplicação ocorre em um ambiente restrito que limita o acesso a recursos compartilhados. Estas rotinas são compostas de um conjunto de instruções primitivas exportadas pelos nós, combinadas com as contruções de controle da linguagem de programação. Neste sistema, a linguagem Java é utilizada para programação das rotinas. Observa-se que a compilação de um programa em Java gera código intermediário [16].

Dentre as categorias de instruções primitivas presentes no ANTS, duas foram utilizadas no desenvolvimento do protocolo ativo para comunicação multiponto móvel. As primitivas de **controle de operação** permitem que cápsulas transmitam, descartem, suspendam ou acordem a si mesmas. As primitivas de **acesso ao armazenamento não-transiente**, isto é, acesso ao *soft-state*, que neste sistema, são objetos com significado definido pela aplicação.

Vistos a definição de protocolo ativo, o comportamento dos nós ativos e a programação das rotinas de processamento, torna-se necessária a definição do mecanismo de distribuição de código. Ao registrar o código de um novo protocolo no nó ativo local, cápsulas referentes a este novo protocolo podem ser injetadas na rede e recebidas dela. Conforme as cápsulas atravessam os nós da rede, um protocolo leve é utilizado para transferir as rotinas de cápsulas, incrementalmente, de um nó ativo para o próximo. A sequência de eventos ilustrando a operação deste protocolo é mostrado na figura 3.

As cápsulas possuem uma identificação do seu tipo e do protocolo ao qual pertencem. Ao receber uma cápsula, o nó ativo verifica o seu *cache* de protocolos, que armazena as

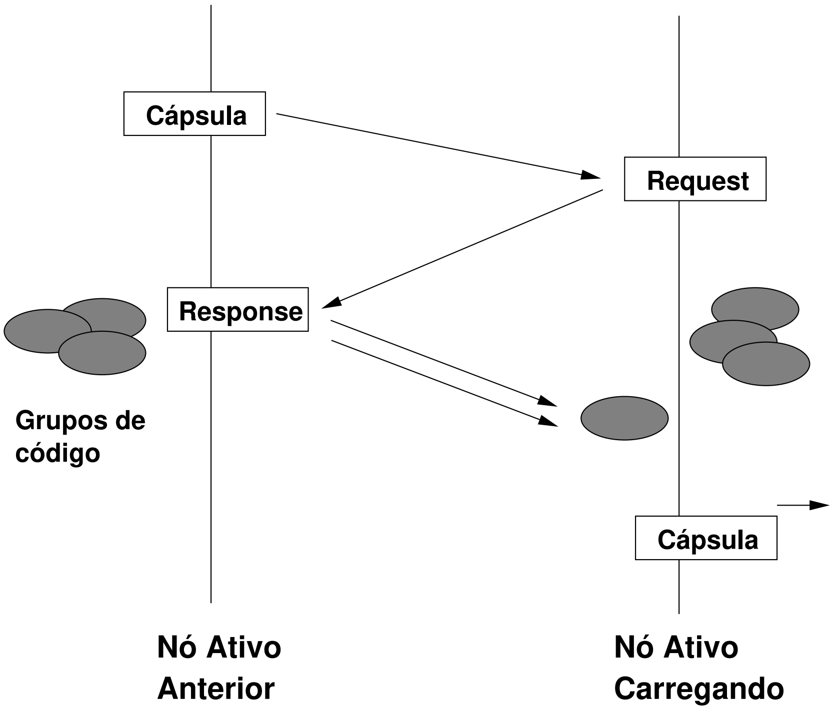

Figura 3: Carregamento sob demanda de grupos de código

rotinas de processamento dos protocolos em uso. Se o código requerido não estiver presente, um pedido é feito ao nó anterior, do qual recebeu esta cápsula, e esta é suspensa. O nó anterior, ao receber o pedido, envia imediatamente o grupo de código que contém as rotinas necessárias. Ao receber a resposta, o nó ativo atual incorpora o código ao seu *cache* de protocolos e ativa a cápsula para execução.

O modelo para desenvolvimento de protocolos apresentado pelo sistema ANTS foi utilizado no protocolo proposto neste artigo. Esta seção apresentou uma visão geral do sistema ANTS e as suas características que são utilizadas nas seções seguintes deste artigo.

3 Comunicação multiponto e mobilidade

O aumento do número de aplicações com requisitos de comunicação multiponto despertou o interesse de diversos pesquisadores pelos problemas relacionados a este tipo de comunicação. Entre os diversos requisitos relacionados à comunicação multiponto estão a escalabilidade e a confiabilidade dos protocolos de comunicação, a garantia de ordenação dos pacotes e o roteamento multiponto.

Quando uma aplicação multiponto exige confiabilidade do protocolo de transporte, uma abordagem utilizando reconhecimentos negativos (NACKs) ao invés de reconhecimentos positivos (ACKs) é preferível [17]. Isto ocorre devido ao fato de que a emissão de ACKs por cada receptor e para cada pacote recebido, gera sobrecarga na rede e a necessidade de sincronização no servidor com esta emissão de reconhecimentos [18]. O protocolo SRM (*Scalable Reliable Multicast*) utiliza intervalos aleatórios de tempo, de acordo com a distância do transmissor, para o envio de NACKs para evitar a implosão destes reconhecimentos e de retransmissões. A função de ordenação é deixada para a aplicação. O protocolo XTP (*eXpress Transfer Protocol*) utiliza um algoritmo denominado balde (*bucket*) que fornece uma transferência de dados estatisticamente confiável. Neste algoritmo, a preferência é dada aos receptores mais rápidos, isto é, que reportam mais rápido aos transmissores [18].

Outra área de interesse dentro de comunicação multiponto é o roteamento. Os algoritmos para roteamento multiponto utilizam de alguma forma uma árvore de distribuição. Esta distribuição pode aparecer de três formas: inundação, árvore baseada no transmissor e árvore baseada em núcleo (CBT - *Core-based Tree*) [18]. Quando o algoritmo utiliza o processo de inundação, tem-se a vantagem da gerência de pouca ou nenhuma informação dos grupos multiponto nos nós. Como desvantagem deste processo, tem-se o desperdício de mensagens enviadas sem necessidade. Se o algoritmo utilizar árvore baseada no transmissor, tem-se a vantagem da economia de mensagens desnecessárias, com a desvantagem da necessidade de gerenciamento das informações de grupo nos nós. Outra desvantagem deste processo existe para o caso de múltiplos transmissores, pois desta forma, devem existir tantas árvores quanto transmissores. Este problema é resolvido utilizando-se árvores baseadas em núcleo [19]. Neste tipo de algoritmo, para difundir uma mensagem para o grupo, esta deve ser enviada para o núcleo ou raiz da árvore. A desvantagem deste algoritmo está no atraso acrescentado para o envio primeiro para o núcleo.

Para o caso de distribuição de informação abordado neste artigo, onde as estações receptoras podem mover-se com o passar do tempo, adiciona-se os problemas de mobilidade. As soluções apresentadas para a mobilidade com o IP envolvem a presença de agentes locais e estrangeiros que servem como intermediários para a recepção de mensagens [20]. Isto para que a camada de transporte (TCP) não perca a conectividade com as mudanças de ponto, mantendo o seu endereço IP. O ideal para a aplicação multiponto seria que os dados fossem enviados diretamente para a estação móvel, sem o uso de agentes intermediadores. Esta opção é apresentada em [21], com o agente estrangeiro realizando o serviço de um roteador multiponto e mascarando o endereço das estações móveis. No entanto, o agente estrangeiro pode não desejar prestar este serviço, e a sequência da comunicação multiponto estaria quebrada.

O protocolo ativo proposto neste artigo implementa soluções para vários dos problemas mencionados para o caso de comunicação multiponto com mobilidade, utilizando conceitos conhecidos para a solução de problemas como confiabilidade e escalabilidade. A solução dada elimina a presença de agentes intermediários na difusão de mensagens multiponto.

4 Protocolo ativo para comunicação multiponto móvel

O protocolo ativo proposto neste trabalho segue a abordagem do sistema ANTS, onde os protocolos são formados por grupos de códigos, e cada grupo de código é formado por cápsulas. É importante observar que o grupo de código é a unidade trocada entre os nós ativos. São definidos dois grupos de código: **distribuição de dados** e **gerência de distribuição**. Esta divisão existe devido ao comportamento apresentado perante o caso específico de distribuição de dados multiponto, onde o transmissor utiliza-se somente do grupo de distribuição de dados e os receptores do grupo de gerência de distribuição.

4.1 Grupo de gerência de distribuição

No grupo de gerência de distribuição existem três tipos de cápsulas que são: **inscrição**, utilizadas por receptores que desejam aderir ao grupo multiponto, realizando a expansão da árvore de distribuição, ou para confirmar periodicamente a adesão ao grupo; **redirecionamento**, utilizadas pelos receptores durante a troca de domínio (*handoff*) para não perderem as cápsulas de dados multiponto que já se encontravam em curso; por último, cápsulas de **NACK**, que pedem aos nós ativos intermediários o reenvio de unidades de dados perdidas.

A seguir são apresentados os algoritmos em linguagem estruturada referentes a cada cápsula e uma breve explicação do seu funcionamento.

4.1.1 Cápsula de Inscrição

```
Procedimento Inscrição
Se existe registro de grupo
   Então extrair
   Senão criar
Se existe indicação para o enlace de saída
   Então Se tempo próximo de expiração
      Então
           atualizar tempo de expiração da indicação
           seguir caminho reverso
      Senão cápsula descartada
   Senão
      criar indicação para o enlace de saída
      seguir caminho reverso
Fim-Procedimento
```

A cápsula de inscrição primeiro busca o registro do grupo multiponto de interesse com as indicações para os próximos enlaces de saída que formam a árvore de distribuição. Caso o registro não exista, é criado. Se a indicação para o enlace de saída em direção ao receptor interessado no grupo multiponto já existir, o algoritmo verifica se a indicação está próxima de expirar, pois neste caso a cápsula deve continuar percorrendo o caminho reverso a fim de manter o *soft-state*. Por outro lado, se a indicação ainda não existir, ela é criada e a cápsula continua o seu percurso em direção ao transmissor. É importante observar que o tempo de expiração deve ser feito por indicação de enlace de saída e não pelo registro completo do grupo. Uma estação desliga-se de um grupo quando deixa de enviar cápsulas de inscrição, se o tempo de expiração fosse por registro e não por indicação, uma vez dentro de um grupo, a estação permaneceria na sua árvore de distribuição até o desligamento global de todos os integrantes.

4.1.2 Cápsula de Redirecionamento

```
Procedimento Redirecionamento
Se existe registro de grupo
   Então extrair
   Senão criar
Se existe indicação para o enlace de saída
   Então
      atualizar tempo de expiração da indicação
      transmitir cápsula pelo enlace de saída
   Senão
      criar indicação para o enlace de saída
      transmitir cápsula pelo enlace de saída
Fim-Procedimento
```

O algoritmo para a cápsula de redirecionamento é semelhante ao algoritmo de inscrição. A diferença está em que as cápsulas de inscrição percorrem o caminho inverso das cápsulas de dados, enquanto que as cápsulas de redirecionamento percorrem o caminho das cápsulas de dados no sentido direto do nó ativo antigo para o nó ativo atual onde se encontra a estação móvel. Por estar percorrendo o caminho no sentido direto, as cápsulas de redirecionamento não podem ser descartadas até chegar ao nó ativo atual, pois necessita ter certeza de que existem indicações nos nós ativos até o destino.

4.1.3 Cápsula NACK

Procedimento NACK
Se `existe registro de grupo`
    Então `extrair`
    Senão `descartar cápsula`
Se `unidades de dados requeridas presentes no` *buffer* `do nó`
    Então
        `pedido de retransmissão`
        `descartar a cápsula`
    Senão Se `outra cápsula NACK requeriu as mesmas unidades de dados`
        Então `descartar cápsula`
        Senão `prosseguir pelo caminho inverso`
Fim-Procedimento

Para cada grupo multiponto, em um determinado nó ativo, existe um *buffer* que guarda uma determinada quantidade de unidades de dados. Durante a execução do procedimento NACK, verifica-se a existência das unidades de dados desejadas neste *buffer*. Caso elas estejam presentes no nó ativo intermediário em questão, o pedido de retransmissão é efetuado. Por outro lado, se não estiverem presentes, antes de ser transmitida para o próximo nó no sentido reverso do tráfego de dados, deve-se verificar se existe no registro de grupo multiponto do nó a marcação de uma outra cápsula de NACK, enviada anteriormente por outro receptor com pedido para as mesmas unidades de dados. Se esta afirmativa for verdadeira, a cápsula atual é descartada. Caso contrário, uma marcação é feita no registro de grupo multiponto do nó e a cápsula NACK prossegue o caminho na busca das unidades de dados para retransmissão.

4.2 Grupo de distribuição de dados

No grupo de distribuição de dados existe apenas a cápsula de **dados multiponto** que utiliza as indicações existentes nos nós intermediários para percorrer a árvore de distribuição e entregar as unidades de dados aos receptores.

4.2.1 Cápsula de dados multiponto

Procedimento UDados
Se `existe registro de grupo`
    Então `extrair`
    Senão `descartar cápsula`
Se `unidade de dados já existe no` *buffer*
    Então `descartar cápsula`
    Senão
        Para `cada enlace de saída indicado no registro de grupo`
            `enviar uma cópia da cápsula de dados`
`Colocar unidade de dados no` *buffer*
Fim-Procedimento

As cápsulas de dados multiponto simplesmente percorrem a árvore de distribuição construída pelas cápsulas de inscrição e de redirecionamento. Em cada nó ativo intermediário, envia-se uma cópia da cápsula de dados para cada enlace de saída indicado no registro de grupo.

Nesta seção, foram definidas as cápsulas que compõem o protocolo ativo junto com os seus algoritmos e as suas funcionalidades. Na próxima seção uma visão mais detalhada do funcionamento do modelo, a partir da descrição do comportamento de transmissores e receptores e da interação entre as cápsulas participantes.

5 Descrição do modelo de comunicação

A flexibilidade provida pelo sistema ANTS para os usuários da rede, a partir do fornecimento de elementos de uso geral, como o *soft-state* e a linguagem de programação Java, permite a adequação do serviço de rede à aplicação. Em determinados casos é possível obter-se melhorias em termos de banda passante, atraso dos dados e número de mensagens trafegando na rede, melhorando o desempenho destas aplicações.

Quando se tratam de estações móveis que mudam fisicamente de posição na rede no decorrer do tempo, torna-se necessária, normalmente, a presença de nós intermediários como agentes locais e estrangeiros. Para o caso específico de distribuição de dados onde é necessário quando, por exemplo, um empresário do setor financeiro encontra-se viajando pelo mundo, mas deseja receber constantemente dados do mercado financeiro mundial. Portanto, supõe-se que existe um servidor de dados que transmite para os receptores através de uma árvore de distribuição, ou um nó da rede que corresponde à raiz de uma árvore de distribuição baseada no algoritmo CBT (*Core-Based Tree*) para onde todos que querem difundir suas mensagens devem enviá-las.

O ideal para este tipo de aplicação é que o fluxo de informação seja enviado diretamente para a estação móvel sem a presença de nós intermediários, isto é, sem passar primeiro por um agente local e em seguida por um agente estrangeiro, como é sugerido nas propostas que envolvem IP multiponto. Assim, o primeiro passo dentro do modelo proposto neste artigo é a adesão direta da estação móvel à árvore de distribuição multiponto do transmissor desejado. Isto é feito através do envio de uma cápsula de inscrição ao grupo. Como é mostrado na figura 4, esta cápsula percorre o caminho na direção do transmissor até encontrar o próprio ou um outro nó ativo que já faça parte da árvore de distribução multiponto. O caminho a ser percorrido é dado por um algoritmo de roteamento. O roteamento não é um problema abordado neste trabalho, sendo suposto, a princípio, um roteamento baseado em rotas estáticas.

A cada nó ativo pelo qual a cápsula de inscrição passa, procura pelo registro do grupo multiponto de interesse, onde existem indicações dos enlaces de saída para este grupo. Se o registro for encontrado, a cápsula verifica se o tempo de validade do *soft-state* está expirando. Caso esteja expirando, a cápsula realiza a atualização e prossegue caminho. Caso contrário ela é descartada. Dado que o sistema exige o envio periódico destas cápsulas de inscrição para confirmar a adesão dos receptores, esta solução é escalável. Esta afirmação é verdadeira porque conforme uma cápsula de inscrição percorre o caminho inverso na direção do transmissor, se esta cápsula descobre que o registro foi atualizado recentemente por outra, então esta cápsula é descartada. Assim, este procedimento não permite a implosão de cápsulas de inscrição. As cápsulas de dados multiponto percorrem a árvore de distribuição que é formada pelas indicações existentes nos nós ativos.

Quando uma estação móvel muda fisicamente de posição, isto é, passa para o domínio de outro nó ativo (*handoff*), uma cápsula de redirecionamento é enviada pelo nó ativo

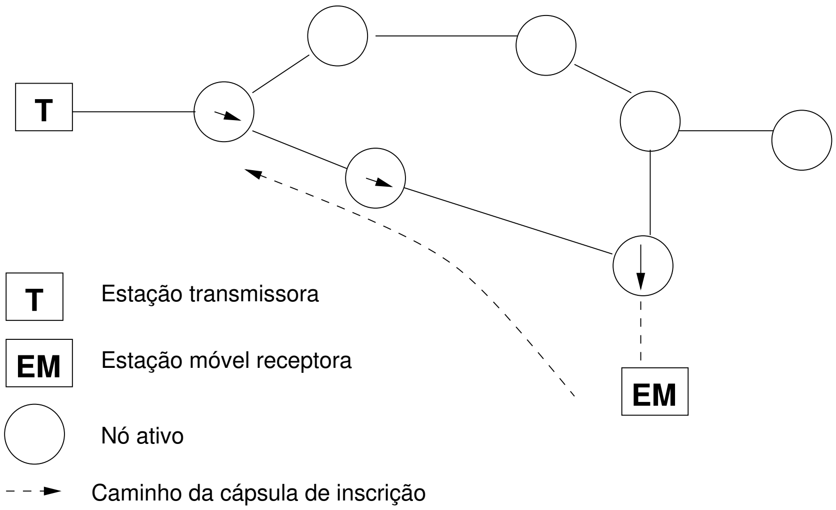

Figura 4: Caminho da cápsula de inscrição

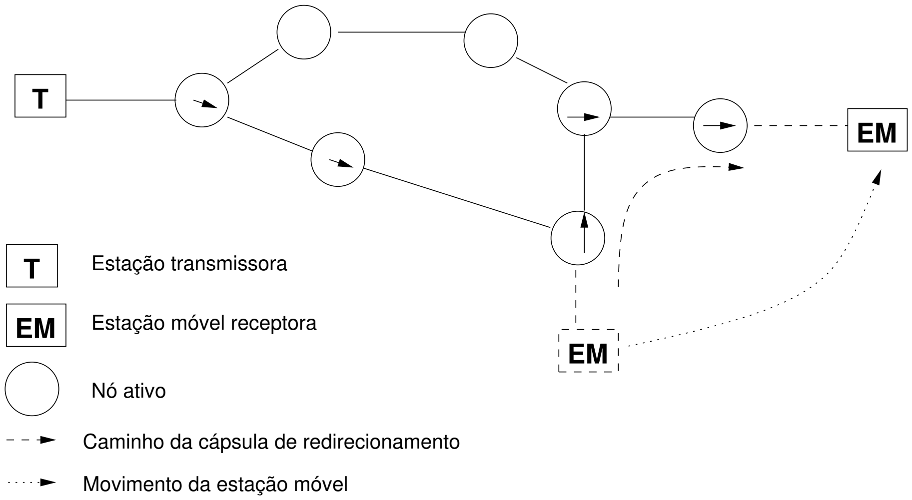

Figura 5: Caminho da cápsula de redirecionamento

antigo em direção ao nó ativo atual, como é mostrado na figura 5. Observa-se que isto é feito a partir da sinalização do enlace da necessidade de mudança de domínio. Desta forma, as cápsulas de dados multiponto destinadas a estação móvel e que já estavam em curso não deixam de ser entregues. Após o *handoff* a estação móvel deve enviar imediatamente uma cápsula de inscrição através do nó ativo atual, a fim de atualizar o caminho das cápsulas de dados multiponto.

O redirecionamento feito durante a fase de *handoff* pode ocasionar dois problemas: desordem e duplicação dos pacotes. A solução proposta, para manter um grau de simplicidade, não inclui a ordenação destas cápsulas, deixando-a para a aplicação usuária do serviço de rede. O problema de duplicação é resolvido facilmente graças ao *buffer* de unidades de dados por grupo previsto em cada nó ativo presente na árvore de distribuição. Antes de se multiplicar e se enviar pelos enlaces indicados no registro de grupo, a cápsula de dados multiponto verifica se sua unidade de dados já não está presente no *buffer*. Se estiver, ela é descartada.

A confiabilidade deste protocolo multiponto é conseguida com a combinação de cápsulas

NACK e do *buffer* de unidades de dados que armazena as unidades recebidas através de cápsulas de dados multiponto. A quantidade de unidades de dados comportada por este *buffer* dependerá diretamente do período médio de envio de cápsulas de dados e da quantidade máxima de registros disponível por grupo.

Ao detectar um salto no sequenciamento das unidades de dados, a estação deve enviar uma cápsula NACK contendo os números de sequência faltosos, de forma que sejam buscados nos nós intermediários da rede. A cápsula NACK percorre todo o caminho reverso até o transmissor até que as unidades de dados desejadas sejam encontradas nos *buffers*. São então reenviadas em cápsulas de dados multiponto.

Para evitar a implosão de cápsulas NACK, ao passar por um nó ativo realizando o caminho inverso, esta cápsula deve verificar no registro de grupo se alguma cápsula de NACK já passou por este caminho demandando os mesmos números de sequência faltosos. Se já houver registro de uma passagem anterior a cápsula é descartada, caso contrário registra a sua passagem e segue caminho.

Assim, cápsulas NACK em conjunto com as outras cápsulas definidas para este protocolo utilizam o conjunto de recursos disponibilizado pelo modelo integrado de redes ativas para fornecer um serviço de distribuição de dados baseado em comunicação multiponto, confiável e escalável destinado a estações móveis.

6 Conclusões

Os diferentes requisitos exigidos dos sistemas de comunicação pelas aplicações distribuídas sugerem a necessidade de um serviço de rede flexível e ajustável às necessidades destas aplicações. A proposta das redes ativas é tornar a rede programável de forma que computações específicas possam ser realizadas sobre os fluxos de informação de acordo com a dinâmica das aplicações.

O modelo integrado, no qual se baseou este trabalho, substitui pacotes contendo apenas dados, por pacotes contendo dados e código executável. No contexto do sistema ANTS, os pacotes convencionais são substituídos por cápsulas que contêm dados e uma referência para uma rotina de processamento associada ao seu tipo de cápsula. Assim, ao percorrer os nós ativos da rede, estas rotinas de processamento tomam decisões a respeito do destino das cápsulas, podendo alterar o conteúdo dos seus dados e utilizar os recursos disponíveis nos nós ativos para comunicarem-se com outras cápsulas.

Utilizando a flexibilidade provida pelo modelo integrado, foi proposto neste artigo um modelo ativo de comunicação multiponto escalável e confiável para estações móveis. Neste modelo ativo, as estações móveis aderem diretamente à árvore de distribuição, sem o uso de agentes intermediários. A partir deste modelo, foi proposto um protocolo ativo baseado no esquema de desenvolvimento definido pelo sistema ANTS. Foram definidas as cápsulas necessárias, assim como os algoritmos para as rotinas de processamento. A cápsula de dados multiponto do grupo de dados percorre a árvore de distribuição formada por cápsulas de gerência. Cápsulas de gerência foram definidas para que os receptores possam:

- aderir à árvore de distribuição multiponto para receber os dados de um determinado grupo multiponto;
- requisitar aos nós ativos intermediários a retransmissão de unidades de dados, que permanecem armazenadas nos *buffers* de grupo, evitando a implosão de pedidos de retransmissão e retransmissões.

Após a descrição do modelo de comunicação proposto, pôde-se observar as principais vantagens deste modelo, que são:

- a dimuição do atraso no envio de mensagens e a diminuição do tráfego de informações de gerência de mobilidade devido a inexistência de agentes intermediários;
- a escalabilidade, importante quando é prevista uma grande quantidade de usuários nos grupos, que é realizada através da filtragem, feita nos nós para as cápsulas de inscrição e NACK, e ainda através do descarte de cápsulas contendo unidades de dados já recebidas;
- a confiabilidade, obtida através do armazenamento de unidades de dados nos nós ativos intermediários e pedidos de retransmissão através de cápsulas NACK.

Como propostas para trabalhos futuros, tem-se a simulação para medidas de ganho em termos do atraso na transmissão das mensagens de dados e da economia nas mensagens de gerência de mobilidade, e a implementação de um protótipo utilizando as ferramentas do sistema ANTS.

Agradecimentos

Gostaríamos de agradecer o apoio financeiro fornecido por UFRJ, FUJB, PROTEM-CC, CNPq, CAPES e COFECUB.

Referências

[1] J. G. David Wetherall e D. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols", in *IEEE OPENARCH'98, a ser publicado*, São Francisco, CA, abril 1998.

[2] D. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. Wetherall e G. Minden, "A Survey of Active Network Research", *IEEE Communications Magazine*, pp. 80–86, janeiro 1997.

[3] G. Lauer, "IN Architectures for Implementing Universal Personal Telecommunications", *IEEE Network*, pp. 6–16, abril 1994.

[4] J. Smith et al., "SwitchWare: Accelerating Network Evolution", tech. rep., Universidade da Pensilvânia, junho 1996.

[5] S. Bhattacharjee, K. Calvert e E. W. Zegura, "On Active Networking and Congestion", tech. rep., GIT-CC-96-02, College of Computing, Georgia Tech, 1996.

[6] S. Bhattacharjee, K. Calvert e E. W. Zegura, "Implementation of an Active Networking Architecture", in *Workshop Gigabit Switch Technology*, Universidade de Washington, St. Louis, julho 1996.

[7] S. Bhattacharjee, K. Calvert e E. W. Zegura, "An Architecture for Active Networking", in *High Performance Networking*, White Plains, NY, abril 1997.

[8] A. Romanov e S. Floyd, "Dynamics of TCP Traffic over ATM Networks", *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 4, pp. 633–641, maio 1995.

[9] D. S. Alexander, M. Shaw, S. M. Nettles e J. M. Smith, "Active Bridging", in *ACM SIGCOMM'97*, Cannes, France, setembro 1997.

[10] T. Yemini e S. da Silva, "Towards Programmable Networks", in *IFIP/IEEE International Workshop on Distributed Systems*, outubro 1996.

[11] M. Hicks, P. Kakkar, J. T. Moore, C. A. Gunter e S. Nettles, "PLAN: A Programming Language for Active Networks". submetido ao ACM SIGPLAN, junho 1998.

[12] J. Hartman, U. Manber, L. Peterson e T. Proebsting, "Liquid Software: A New Paradigm for Networked Systems", Tech. Rep. 96-11, Universidade do Arizona, 1996.

[13] D. Tennenhouse e D. Wetherall, "Towards an Active Network Architecture", *Computer Communication Review*, vol. 26, no. 2, abril 1996.

[14] D. Wetherall e D. Tennenhouse, "The ACTIVE IP Option", in *Proceedings of the 7th SIGOPS Workshop*, Connemara, Irlanda, setembro 1996.

[15] F. Junqueira, L. H. K. Costa e O. C. M. B. Duarte, "Uma Arquitetura de Roteamento com Qualidade de Serviço baseada em Redes Ativas", in *Seminário Franco-Brasileiro de Sistemas Informáticos Distribuídos*, Fortaleza. Ceará, novembro 1997.

[16] Sun Microsystems Inc., *The Java Virtual Machine Especification*, 1995.

[17] D. Towsley e J. Kurose, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocol", *Journal on Selected Areas in Communications*, vol. 15, no. 3, abril 1997.

[18] C. Diot, W. Dabbous e J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms", *Journal on Selected Areas in Communications*, vol. 15, no. 3, abril 1997.

[19] T. Ballardie, P. Francis e J. Crowcroft, "Core-based Trees (CBT) in a Architecture for Scalable Inter-Domain Multicast Routing", in *SIGCOMM'93*, São Francisco, CA, pp. 13–17, setembro 1993.

[20] R. O. LaMaire, A. Krishna, P. Bhagwat e J. Panian, "Wireless LANS and Mobile Networking: Standards and Future Directions", *IEEE Communications*, agosto 1996.

[21] G. Xylomenos e G. Polyzos, "IP Multicast for Mobile Hosts", *IEEE Communications*, pp. 54–58, janeiro 1997.