Especificação Formal de Uma Arquitetura de Suporte à Descoberta de Serviços em Redes Móveis Ad Hoc

Renato Bagatelli[1,3], David F. C. Moura[1,3] e Aloysio C. P. Pedroza[1,2]

[1] Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
{rbagatel, mdavid, aloysio}@gta.ufrj.br
[2] COPPE/EE - Programa de Engenharia Elétrica
Departamento de Eletrônica, Universidade Federal do Rio de Janeiro
[3] Grupo de Tecnologia da Informação, Instituto de Pesquisa e Desenvolvimento,
Exército Brasileiro

Resumo
Este trabalho apresenta uma arquitetura de suporte à descoberta de serviços em redes móveis *ad hoc*, utilizando uma metodologia de teste, verificação e análise de protocolos baseada em técnicas de verificação formal.
Foram feitas análises de protocolos, em especial de documentos do IETF (*Internet Engineering Task Force*), para redes *ad hoc* em vários níveis. Tais protocolos são descritos e analisados para dar suporte a descoberta de serviços, objetivo funcional principal deste trabalho.
São apresentados resultados da verificação de propriedades, validação de modelos e simulação de dois dos protocolos da arquitetura a partir de especificações escritas em LOTOS (*Language of Temporal Ordering Specification*) , visando ao final à validação do modelo completo.
Palavras-Chave: especificação formal, LOTOS, protocolos de comunicação, redes *ad hoc*, descoberta de serviços.

1 Introdução

Este trabalho apresenta uma arquitetura de suporte à descoberta de serviços em redes móveis *ad hoc*, utilizando uma metodologia de teste, verificação e análise de protocolos baseada em técnicas de verificação formal.

Uma rede *ad hoc* pode ser definida [1] como um sistema autônomo de plataformas (ou nós) móveis. Tais redes operam sem uma infraestrutura fixa que realize atividades de controle de rotas ou encaminhamento de pacotes - os próprios nós são os roteadores e/ou controladores da rede.

Nesse ambiente de grande mobilidade, seus nós devem funcionar com um mínimo de instalação de software possível [1]. Isso se dá devido a restrições de dimensões físicas e de fontes de alimentação nos dispositivos, o que reduz a capacidade de armazenamento e processamento nos mesmos; contudo, é desejável que todas as facilidades e serviços instalados em um nó da rede sejam accessíveis a todos os demais usuários, inclusive os pertencentes a outras redes, móveis ou até mesmo fixas. Assim, destaca-se a necessidade das estações disporem de uma arquitetura que suporte a localização dos serviços de que necessitam e o anúncio daqueles que possuem.

Os protocolos escolhidos para esta arquitetura foram o SLP (*Service Location Protocol*) [2], como protocolo de descoberta de serviços; o ADTCP (*Ad Hoc Transport Control Protocol*) [3] para a camada de transporte; o DSR (*Dynamic Source Routing Protocol*) [4], desempenhando atividades de roteamento; por fim, o MACAW (*Medium Access Protocol for Wireless LAN's*) [5], como protocolo de acesso ao meio. Além da descrição da arquitetura escolhida, foi realizada a especificação de alguns dos protocolos propostos, utilizando a linguagem LOTOS [6], acompanhada de verificação com o *software* de análise CADP (*Pacote de Desenvolvimento Caesar/Aldebaran*) [7].

Este trabalho está dividido da seguinte forma: na seção 2, descrevem-se as redes *ad hoc*, destacando aspectos de roteamento e descoberta de serviços. A arquitetura proposta é apresentada na seção 3, enfatizando cada protocolo do modelo. A metodologia utilizada é descrita na seção 4, servindo por base ao modelo de rede proposto na seção 5. A seção 6 apresenta estudos de caso aplicados à arquitetura, com especificação e resultados. Por fim, na seção 7, descrevem-se alguns comentários finais sobre o trabalho, bem como propostas de pesquisas futuras.

2 Redes Ad Hoc: Roteamento e Descoberta de Serviços

Uma rede *ad hoc* [1] é um sistema altamente dinâmico por natureza; há uma movimentação muito grande dos nós, em direções aleatórias [8]. Com isso, a conectividade entre as estações é um problema crítico; fatores como posição dos nós, potência de transmissão, sensibilidade do receptor e capacidade do canal fazem com que o estabelecimento de rotas seja altamente dependente do tempo. Tal cenário traz dificuldades para diversas aplicações neste ambiente, como a descoberta e a utilização dos diversos serviços disponíveis na rede, sem a necessidade de informar endereços ou nomes de estações na rede; este é o objetivo dos protocolos de descoberta de serviços [2,9,10,11,12].

Uma série de questões resulta deste cenário: obtenção da rota entre duas estações distintas [13], solicitação e anúncio de serviços, segurança das informações, critérios de seleção de serviços, mecanismos de estabelecimento de conexão entre estações, dentre outras tarefas, implicando na escolha e integração de protocolos em uma arquitetura que as execute de forma eficiente.

3 Uma Arquitetura para Redes Ad Hoc

Este trabalho apresenta uma arquitetura para redes *ad hoc*, conforme descrita na figura 1, onde vários protocolos atuam em cooperação para fornecer facilidades de descoberta de serviços a diferentes aplicações em ambientes de redes móveis *ad hoc*.

A maior parte das arquiteturas é organizada como um conjunto de camadas ou níveis superpostos. Uma camada n qualquer de uma estação comunica-se tão somente com a camada n de uma outra estação. Tal estratégia de divisão em camadas faz com que o projeto seja dividido em partes independentes entre si, o que facilita a formalização e a implementação de protocolos, bem como a manutenção da arquitetura. Com isso, são permitidas modificações particulares a cada protocolo, desde que mantidas as interfaces necessárias, de acordo com requisitos do serviço sob modelagem.

Entende-se por *protocolo* de uma camada como o conjunto de regras de comunicação governando o formato e o significado das mensagens trocadas dentro desta camada entre estações distintas. Já o *serviço* prestado por uma camada é o conjunto de operações que ela provê à camada superior; a definição de serviço descreve as interfaces e as atividades disponíveis à camada, sem detalhar como tais operações são implementadas, pois as estações se valem de protocolos para implementar suas definições de serviços.

Nesta proposta de arquitetura, foram adotados protocolos já abordados separadamente na literatura - por exemplo, *drafts* e RFC's (*Request for Comments*) do IETF. Os protocolos são:

- SLP (*Service Location Protocol*) [2], como protocolo de descoberta de serviços;
- ADTCP (*Ad Hoc Transport Control Protocol* [3]) para a camada de transporte;
- DSR (*Dynamic Source Routing Protocol*) [4], desempenhando atividades de roteamento; e
- MACAW (*Medium Access Protocol for Wireless LAN's*) [5], como protocolo de acesso ao meio.

Cada modelo é detalhado a seguir.

3.1 Protocolo de Descoberta de Serviços - SLP

O protocolo de localização de serviços (SLP - *Service Location Protocol*) [2] oferece um mecanismo dinâmico de configuração a aplicações em redes locais, permitindo que estações tenham acesso a informações acerca da disponibilidade, localização e configuração de serviços nesta rede local.

A configuração mínima recomenda dois tipos de agentes: os *agentes de usuários*, representando as aplicações dos usuários e os *agentes de serviços*, em nome dos serviços divulgados na rede. Os agentes de usuários difundem na rede requisições aos agentes de serviços, descrevendo serviços ou seus tipos e atributos; estes, caso possuam algum registro de serviço que seja adequado ao requisitado, enviam uma resposta, contendo a localização do(s) serviço(s).

Como forma de conferir escalabilidade ao protocolo, é permitida a adoção de um terceiro tipo de agente: o *agente de diretórios*. Esse agente atua como um repositório, seja registrando serviços, seja armazenando requisições dos agentes de usuários. Tais agentes de diretórios devem ser descobertos pelos agentes de usuários e de serviços por meio de mensagens de busca ou mediante emissão de anúncios pelo próprio agente de diretório.

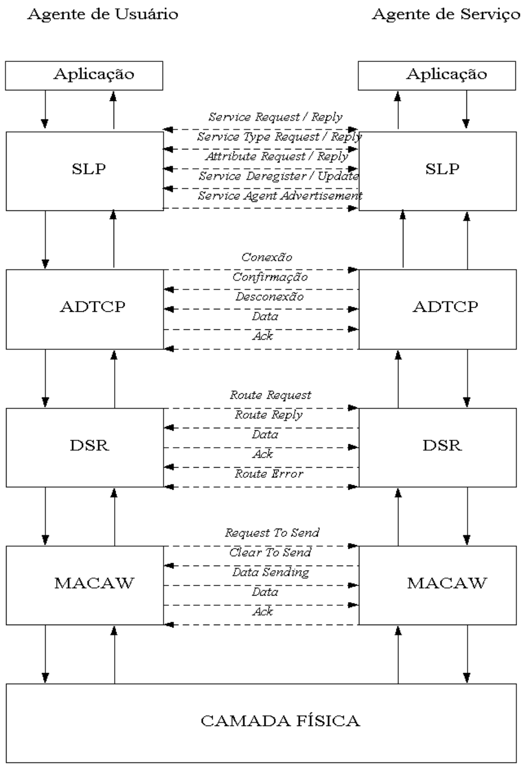

Figura 1. Arquitetura proposta neste trabalho.

3.2 Protocolo de Transporte - ADTCP

Diversas alternativas têm sido apresentadas na literatura nos últimos anos para aprimorar e otimizar o TCP (*Transport Control Protocol*) para redes sem fio, como a melhor forma de fornecer a estas aplicações confiabilidade à comunicação e capacidade de interconexão de redes distintas. Algumas propostas [14,15] se mostram voltadas exclusivamente a redes de telefonia celular, não se adequando ao cenário sob estudo. Outras [16,17,18,19] tornam obrigatória a existência de protocolos de controle complexos nas camadas inferiores em todos os nós, o que não se pode garantir devido a heterogeneidade das estações.

A proposta do protocolo de transporte para redes *ad hoc* (ADTCP - *Ad Hoc Transport Control Protocol*) [3] é apresentar um serviço de transporte a um só tempo compatível com implementações já disponíveis em redes fixas (em especial, o TCP-NewReno) e dotado de mecanismos de identificacção e reação aos diferentes comportamentos presentes em redes *ad-hoc* típicas, como perda de pacotes por congestionamento, desconexão, reconexão e entrega desordenada de pacotes devido à mobilidade e ruptura de canal físico.

Para tanto, propõe-se uma abordagem de sistemas de controle para resolução do problema. Métricas como a diferença de retardo, a taxa de entrega desordenada, a vazão e a taxa de perda de pacotes em um determinado intervalo de tempo são avaliadas no receptor. De acordo com os resultados obtidos, promove-se a identificação do comportamento da rede, num conjunto de cinco opções: congestionamento, desconexão, erro de canal, mudança de rota ou operação normal. O resultado da classificação é transmitido ao emissor, com o envio de uma mensagem de sinalização.

3.3 Protocolo de Roteamento - DSR

O grupo de trabalho no IETF que reúne as propostas de protocolos para redes *ad hoc* (*MANET Working Group*) apresenta 6 protocolos de roteamento [4,20,21,22,23,24]. O DSR (*Dynamic Source Routing Protocol*) [4] monta as tabelas de rotas com base apenas na necessidade do envio de mensagens, minimizando essas tabelas e conseqüentemente a necessidade de grande espaço em memória. As trocas de mensagens são ocasionais e os pacotes são menores. Atualmente, é o protocolo mais antigo do IETF para roteamento *ad hoc*, o de especificação mais completa, o mais difundido e o mais testado dentre os existentes, daí sua escolha.

O DSR funciona baseado em duas diretrizes principais: a descoberta e a manutenção de rotas.

- Descoberta de Rotas
  Quando um nó recebe um pacote a ser enviado, ele primeiro consulta a sua tabela de rotas; se possui uma rota para o destino indicado no pacote, transmite por aquele caminho. Se não, inicia o processo de descoberta de rotas (*route request*), por difusão do pacote de requisição de rota. Tal pacote contém o nó de origem, o de destino e a identificação da mensagem (Id).
  Cada nó da rede que recebe este pacote verifica se ele já fora recebido anteriormente, através da verificação do Id e dos nós origem e destino. Caso afirmativo, a mensagem é descartada; caso contrário, o nó transmite um pacote de reconhecimento (*ack*) ao nó emissor, além de verificar se ele próprio é o destino ou se dispõe de uma rota para o mesmo. Caso exista uma rota armazenada ou ele próprio seja o nó procurado, é transmitida uma mensagem de resposta (*route reply*) para a origem, com o seu endereço e o caminho a seguir. Se não, o nó coloca o seu endereço como próximo nó intermediário seqüencial e retransmite, por inundação. O próximo nó que receber a mensagem fará o mesmo, até que o (*route request*) seja encaminhado ao nó de destino.
- Manutenção da Rota (Route Maintenance)
  A manutenção das rotas é feita da seguinte forma: cada nó que transmite um pacote é responsável por armazenar a confirmação de recebimento pelo nó seguinte, sem a necessidade de retransmissão destas confirmações até o nó de origem.
  Caso haja uma ruptura em meio à rota, o nó que encaminhou o pacote ou mesmo a requisição de rota, ao não receber o (*ack*) de seu nó adjacente, após o disparo de seu temporizador, procura em sua *route table* uma nova rota

para reenviar a mensagem ou executa um novo *route request*, dentro de um limite de tentativa. Expirado o temporizador da última tentativa, o nó emite ao seu antecessor uma mensagem de erro (*route error*). Ao receber o (*route error*), o nó antecessor executa o mesmo procedimento do nó anterior; tal ação se propaga até a origem da mensagem, se necessário.

3.4 Protocolo de acesso ao Meio - MACAW

Alguns protocolos de acesso ao meio para redes *ad hoc* tratam este acesso probabilisticamente [25]; outros, interagindo diretamente com os nós para identificar as estações que estão transmitindo naquele momento [5,25].

Para provisão de suporte a descoberta de serviços, é necessário observar que cada terminal poderá estar prestando um determinado serviço e requisitando outro, completamente diferente, conferindo à rede uma infinidade de possibilidades de tráfego em intervalos de tempo variáveis. O tratamento estatístico de alguns protocolos não absorve todas essas possibilidades ao longo do tempo. Portanto, há a necessidade de um protocolo que promova o acesso às estações envolvidas na comunicação e que impeça outras de gerarem tráfegos de colisão, principalmente as chamadas "estações escondidas". Estes requisitos de projeto são especificados no protocolo MACAW [5], justificando sua adoção na arquitetura proposta.

No MACAW [5], a rede começa com todos os nós em silêncio. Quando um nó recebe um pacote e deseja transmiti-lo, escuta o canal e, após verificar que não há nenhuma transmissão, transmite um pacote RTS (*Request to Send*) contendo seu endereço, o do nó destino e o tamanho da mensagem. Ao receber o RTS, o nó destino emite imediatamente um CTS (*Clear to Send*) confirmando o tamanho da mensagem a ser recebida. Ao receber o CTS, o nó de origem emite um DS (*Data Sending*). Com o envio do pacote DS, o canal fica ocupado, o destinatário se prepara para receber os dados e as outras estações da rede acionam o algoritmo de "Back-off", que determinará um novo tempo de espera; isto impede que um outro nó tente abrir uma conexão.

Com isso, após o emissor enviar o pacote DS, transmite os dados e espera a confirmação do receptor através de um ACK. Após a troca de mensagens e dados, as estações voltam para o estado de escuta na rede.

4 Ferramentas e Metodologia de Projeto

4.1 Linguagem de Descrição Formal LOTOS

LOTOS [6,26] é uma Técnica de Descrição Formal padronizada pela ISO (*International Organization for Standardization*) para especificação de protocolos de comunicação e sistemas distribuídos.

LOTOS é composto por duas sub-linguagens:

– LOTOS Básico: representa o modelo estritamente comportamental do sistema. Em tal modelo, o sistema é descrito como um conjunto de processos, os quais são sincronizados mediante execução de ações previamente especificadas em pontos de interação compatilhados. Sua representação é baseada em álgebra de processos, notadamente em CCS (*Calculus of Communicating Systems*) [27] e CSP (*Communicating Sequential Processes*) [26].
– LOTOS Completo: inclui a definição dos tipos de dados envolvidos nas interações entre processos, com base na teoria de tipos de dados algébricos abstratos, em especial na linguagem de especificação ACTONE (*Abstract Data Type formalism*) [28]. Um tipo de dados é descrito por seus operadores e equações, as quais são especificadas mediante emprego de operações algébricas.

Operador	Significado
P !V ?X:T; A	Interação pela porta P, envio de um valor V e recepção na variável X de um valor do tipo T e execução da ação A
A [ ] B	Executa A ou B
A |[f,g,h]| B	Executa A e B em paralelo, com sincronização nas portas f,g,h
A ||| B	Executa A ou B em paralelo, sem sincronização
exit	Termina com sucesso
P [f,g,h] (F,G,H)	Chamada do processo P com parâmetros de portas f,g,h e parâmetros de valores F,G,H

Tabela 1. Operadores de LOTOS

4.2 CADP (Pacote de Desenvolvimento Caesar/Aldebaran)

O CADP [7] é um conjunto de ferramentas empregado em engenharia de protocolos, com o objetivo de promover a compilação, simulação, verificação formal e teste de descrições de protocolos escritas na linguagem LOTOS [6].

Dentre as ferramentas disponíveis, destacam-se pelo seu emprego neste estudo de caso:

– Caesar e Caesar.adt: compiladores que promovem a tradução de especificações em LOTOS em conjuntos de estados e transições, descrevendo todas as possibilidades de comportamento dos processos especificados. Tais sistemas podem ser representados explicitamente, como sistemas de transições rotuladas, ou implicitamente, como uma biblioteca de funções em C, o que permite não só a simulação, mas também a própria implementação em sistemas reais do protocolo especificado. A ferramenta igualmente apresenta

suporte a outros formalismos, como máquinas de estados finitos, redes de Petri, redes de autômatos comunicantes, dentre outros.
Tal ferramenta foi empregada na especificação dos modelos de serviço a serem oferecidos pelas diferentes camadas e nas descrições de comportamento e de tipos de dados a serem implementados nos diversos protocolos.
- Aldèbaran: ferramenta de verificação que efetua tarefas como a comparação e a minimização de sistemas de transições rotuladas e redes de autômatos comunicantes com respeito a relações de bissimulação.
De tais relações, as mais empregadas foram [27]:
  - Equivalência observacional: todo comportamento externamente observado de determinado processo pode ser igualmente realizado por uma ou mais ações de outro processo; esta relação foi utilizada para verificar se os protocolos especificados para cada camada representavam os serviços discriminados inicialmente como requisitos do projeto.
  - Equivalência forte: toda ação interna de um processo deve ser igualmente realizada por uma ação interna de outro processo; esta relação foi utilizada para verificar a relação entre implementações incrementais dos diferentes protocolos.
  - Minimização: redução de grafos de sistemas de transições rotuladas, de acordo com relações de equiavlência forte; este método foi utilizado para representar a essência do comportamento do protocolo, retirando redundâncias e permitindo uma análise mais simples de convergência.

4.3 Metodologia

A premissa fundamental para a especificação formal da arquitetura proposta é que cada protocolo seja especificado considerando que o da camada imediatamente inferior à sua presta sem erros os serviços necessários, de forma a permitir o seu bom funcionamento.

Assim, cada protocolo foi estudado em separado, dividindo o problema da rede em quatro, para que sejam interligados *a posteriori*, pela troca de mensagens (*unidades de dados de serviço - SDU's*) entre eles, com base no modelo da arquitetura. Para tanto, é necessária uma boa especificação do nó, não só quanto ao comportamento interno (protocolo), mas também quanto à sua interação com o ambiente externo (serviço).

Segue o roteiro da metodologia empregada para cada camada neste projeto:

- Edição das operações do protocolo e do comportamento observado do serviço por meio de especificações em LOTOS básico.
- Compilação das especificações e geração dos sistemas de transições rotuladas (LTS) correspondentes.
- Análise do número de estados e transições gerados e minimização dos modelos obtidos de LTS via equivalência módulo observacional [27].
- Verificação da existência de *deadlocks* e *livelocks*.
- Validação do modelo mediante comparação dos LTS do protocolo e do serviço via equivalência módulo observacional [27].

– Repetição da seqüência descrita anteriormente, com a inserção da especificação em LOTOS completo das estruturas de dados empregadas no protocolo e no serviço.

Por fim, serão representadas as ações integrando os protocolos para a formulação final da arquitetura completa.

No estágio atual de desenvolvimento, os protocolos escolhidos para composição da arquitetura estão sendo especificados para diferentes quantidades de estações na rede e validados em relação aos modelos de serviço, já inserindo mecanismos de temporização e passagem de parâmetros.

5 Modelo da Rede Ad Hoc

No modelo de rede proposto, todos os nós têm capacidade de processar todos os protocolos da arquitetura proposta, o que lhes possibilita iniciar a busca de uma nova rota e fazer novas descobertas de serviços [4]. Considera-se que as estações possuem potência de transmissão tal que permita o estabelecimento de enlaces ponto a ponto até o centro da rede; dispõem também de autonomia para se desligar (independentemente da posse de serviços), podendo sair e entrar na rede várias vezes e em locais diferentes.

Contudo, dois problemas se destacam ao se planejar a especificação da arquitetura: a dimensão exageradamente grande do conjunto de alternativas, fazendo com que o conjunto de estados e de soluções tenda a infinito e torne o problema de difícil abordagem matemática e a impossibilidade de representação explícita de restrições temporais em LOTOS, o que dificulta a expressão de todas as funcionalidades dos protocolos.

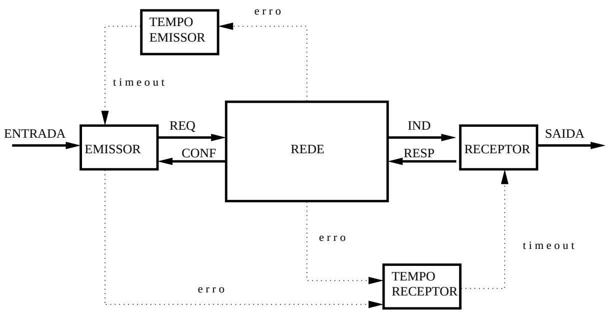

Figura 2. Modelo da rede *ad hoc.*

A proposta para resolução, conforme descrito na figura 2, é a especificação de dois processos especiais em LOTOS, os processos rede e tempo. O *processo rede* representará, nas especificações das camadas transporte e descoberta de serviços, uma abstração das diferentes situações tratadas diretamente pelos protocolos de roteamento e de acesso ao meio em uma rede *ad hoc*, como perda de pacotes, mensagem enviada diretamente ao nó de destino, mensagem enviada a um nó intermediário ou mobilidade de nós. Assim, descrevem-se situações típicas da rede, como congestionamento e erros do canal, desconexão de nós e mudança de rota de forma compacta, evitando a geração de um número muito grande de estados nos sistemas de transições rotuladas gerados.

Visto que o tempo não pode ser explicitamente expresso em LOTOS, o *processo tempo* [29] permite a correta representação de eventos que respeitem uma seqüência temporal, mediante o emprego de enlaces extra para fins de sincronização. Com isso, garante-se que restrições de causalidade não são violadas. Por exemplo, o disparo de um temporizador do nó origem do protocolo de roteamento [4] deve ocorrer tão somente em caso de perda de pacotes ou mudança de rota - isto se verifica por meio da sincronização de canais entre a ação de perdas de pacotes e o processo tempo que represente tal situação.

6 Estudo de Caso

De forma a apresentar a metodologia adotada para a verificação da arquitetura, tomam-se por exemplos os protocolos MACAW e DSR. Em primeiro lugar, foram gerados os LTS correspondentes às descrições em LOTOS completo do protocolo e do serviço MACAW; em seguida, foram especificadas e simuladas as descrições em LOTOS completo do protocolo e do serviço DSR.

6.1 MACAW

Na especificação elaborada, foram definidos dois processos que implementam o protocolo MACAW, interligados por um meio de comunicação simples e exercitados por dois usuários que implementam uma troca simples de dados. A arquitetura empregada para o protocolo MACAW é descrita em LOTOS pela expressão de comportamento a seguir:

```
behaviour
   hide phy_dreq1, phy_dind1, phy_dreq2, phy_dind2,
ma_udind, ma_udreq, ma_udstind, ma_conf in
   (
    (USER1 [ACC,ma_udreq, ma_udstind] ||| USER2 [DEL,ma_udind, ma_conf])
    |[ma_udreq,ma_udind,ma_udstind, ma_conf]|
    (MACAW1[ma_udreq,...,phy_dind1] (RTS of pdu_type)
     |||
     MACAW2[ma_udreq,..,phy_dind2] (RTS of pdu_type))
    |[phy_dreq1,phy_dind1,phy_dreq2,phy_dind2]|
     MEDIUM[phy_dreq1,...,phy_dind2])
```

A descrição em LOTOS do protocolo MACAW contém um parâmetro de domínio infinito - os dados transmitidos entre as estações (tipo DATA). Para restringi-lo e permitir a verificação do modelo, utilizou-se um subconjunto das mensagens possíveis, o que não traz prejuízo ao modelo.

Obteve-se que os modelos de protocolo e de serviço descritos apresentam equivalência observacional, demonstrando que o protocolo especificado atende aos requisitos do serviço. O número de estados e de transições dos modelos especificados de protocolo e serviço são descritos a seguir.

Descrição	Número de Estados	Número de Transições
Protocolo	139	147
Serviço	6	12

Tabela 2. Geração dos LTS do protocolo e do serviço MACAW.

6.2 DSR

O DSR foi descrito em LOTOS, com dois usuários e apresentando alternativas quanto ao número de nós da rede e quanto à utilização ou não de temporizadores. Mostra-se, como exemplo, a arquitetura da descrição para 5 nós.

```
behaviour
   hide phy_req1, ... , dsr_ind, dsr_resp in
      ((User1 [ACC, dsr_req, dsr_conf]|||User2 [dsr_ind, DEL, dsr_resp])
         |[dsr_req, dsr_conf, dsr_ind, dsr_resp]|
         (DSR1 [dsr_req, ... , dsr_conf]||| DSR2 [phy_ind21, ... , rRpC]
      ||| DSR3 [phy_req44, ... , rRpCA] ||| DSR4 [phy_ind25, ... , rRpC]
      ||| DSR5 [phy_ind51, ... , rRpD2C])
        |[phy_req1, ... , rRpCA]|
         Medium [phy_req1, ... , rRpCA])
```

Diversas simulações foram feitas para melhor visualizar e acompanhar o protocolo, bem como o comportamento de seus diagramas de estados e transições. Dada a complexidade da especificação do protocolo e o consequente consumo de recursos computacionais, desenvolveu-se uma metodologia incremental de descrição do protocolo. Assim, foram inseridos a cada versão verificada, conforme descrito na tabela 3, elementos de complexidade do modelo - temporizadores (coluna *Temporizador*), passagem de parâmetros (coluna *Parâmetros*), diferentes quantidades de nós (coluna *Nós*) transmissores (coluna *Tx*) e receptores (coluna *Rx*), além de tratamento de exceções (nós fora de alcance, laços nas tabelas de roteamento, erros de transmissão), até a apresentação de um modelo quase completo de nó no último modelo de protocolo descrito nesta tabela. Com isso, esta série de experimentos culminou com a especificação e verificação de modelos de nós que apresentam transparência em relação ao meio, independência diante dos demais nós da rede e dotados de todas as funcionalidades de transmissão e recepção requeridas de estações reais em redes *ad-hoc*. Para cada experimento, são descritos o número de estados (coluna *Estados*) e de transições gerados (coluna *Transições*), bem como o emprego de relações de minimização (coluna *Minimizado*), todos calculados pela ferramenta empregada.

Descrição	Temporizador	Parâmetros	Nós	Estados	Transições	Minimizado	Tx	Rx
Protocolo	não	não	2	15	15	não	1	1
Protocolo	não	não	2	14	14	sim	1	1
Protocolo	não	não	3	36	39	não	1	2
Protocolo	não	não	3	24	27	sim	1	2
Protocolo	não	não	4	51	56	não	1	3
Protocolo	não	não	4	26	31	sim	1	3
Protocolo	não	não	5	76	85	não	1	4
Protocolo	não	não	5	36	45	sim	1	4
Protocolo	sim	não	2	54	106	não	1	1
Protocolo	sim	não	2	47	95	sim	1	1
Protocolo	sim	não	3	2231	6154	não	1	2
Protocolo	sim	não	3	1472	4181	sim	1	2
Protocolo	não	sim	2	29	33	não	1	1
Protocolo	não	sim	2	15	16	sim	1	1
Protocolo	não	sim	3	3822	4670	não	1	2
Protocolo	não	sim	3	39	45	sim	1	2
Protocolo	sim	sim	3	138788	201940	não	1	2
Protocolo	sim	sim	3	73	106	sim	1	2
Protocolo	não	sim	2	8052151	19307730	não	2	2
Serviço	sim	não	3	6	12	sim	x	x

Tabela 3. Geração dos LTS do protocolo e do serviço DSR.

A simulação com dois nós, apesar de parecer desnecessária para o teste do protocolo, se torna importante na medida em que promove a correção da escrita do mesmo. Nota-se que, sem a utilização de temporizadores, o aumento dos nós gera um aumento suave de estados e transições. Porém, quando se acrescentam os temporizadores, o número de estados cresce substancialmente em resposta ao aumento do número de nós; isto se deve à representação de restrições temporais em LOTOS, que requisitam temporizadores sincronizados com as perdas do meio.

```
process DSR1 [dsr_req, ... , dsr_conf] : noexit :=
        ...;  RotReq [dsr_req, ... , dsr_conf]
   where
```

```
      process RotReq [dsr_req, ... , dsr_conf] : noexit :=
         phy_req1;
            ((hide TIMEOUT in TIMEOUT;
              RotReq [dsr_req, ... , dsr_conf])
              ...
    )
   endproc
...
process Medium [phy_req3, ... , rRpCA] : noexit :=
      phy_req1;
         (phy_ind23; Medium [phy_req3, ... , rRpCA]
           []
           (hide LOSS in LOSS; Medium [phy_req3, ... , rRpCA])
           ...
 )
endproc
```

Observando uma parte do protocolo mostrada como exemplo, verificam-se ações exercitando os processos *timeout* e *loss*, que podem se combinar sequencialmente ao longo do programa. Lembrando que estas devem sempre estar encadeadas e que se sincronizam ao longo das execuções das instruções, há de se notar que uma grande combinação das mesmas gera estes grandes números de estados e consequentemente de transições. Para corrigir este problema, é necessária a indexação destes conjuntos, cada *timeout* com o seu respectivo *loss*, diferenciando uma chamada das outras pela troca de parâmetros.

Como ilustração, é descrito na figura 3 o grafo de accessibilidade para o modelo com três nós (um transmissor e dois receptores) com emprego de temporizador e passagem de parâmetros gerado pela ferramenta utilizada, conforme listado no antepenúltimo ítem da tabela 3. São representados o envio de dados (transição entre estados 20 e 25), descrição de nós fora de alcance (estado 18 para estado 0), laços nas tabelas de roteamento (32 para 61), recepção (16 para 0) e reconhecimento de recepção de mansagens (72 para 4). Constata-se também que o protocolo não só é equivalente ao modelo de serviço proposto, mas também é vivo e reinicializável, visto que, de qualquer estado, é possível alcançar o estado inicial (estado 0), sem a presença de *deadlocks* ou *livelocks*.

Como resultado da verificação, pôde-se constatar que o campo ID, relacionado no documento do IETF à identificação de rotas, mostrou-se desnecessário à lógica do protocolo, visto não ter sido necessário testar este campo em nenhuma estrutura de decisão.

De acordo com a documentação original do protocolo, a verificação deste campo, para um determinado par nó origem/nó destino, permite a identificação de rotas com laços - definindo, consequentemente, a necessidade de descarte de pacotes. A partir da especificação em LOTOS, constata-se que a identificação de rotas livres de laços vem da observação, em cada nó da rota, se o mesmo já estava contido na lista de nós intermediários para um par origem/destino.

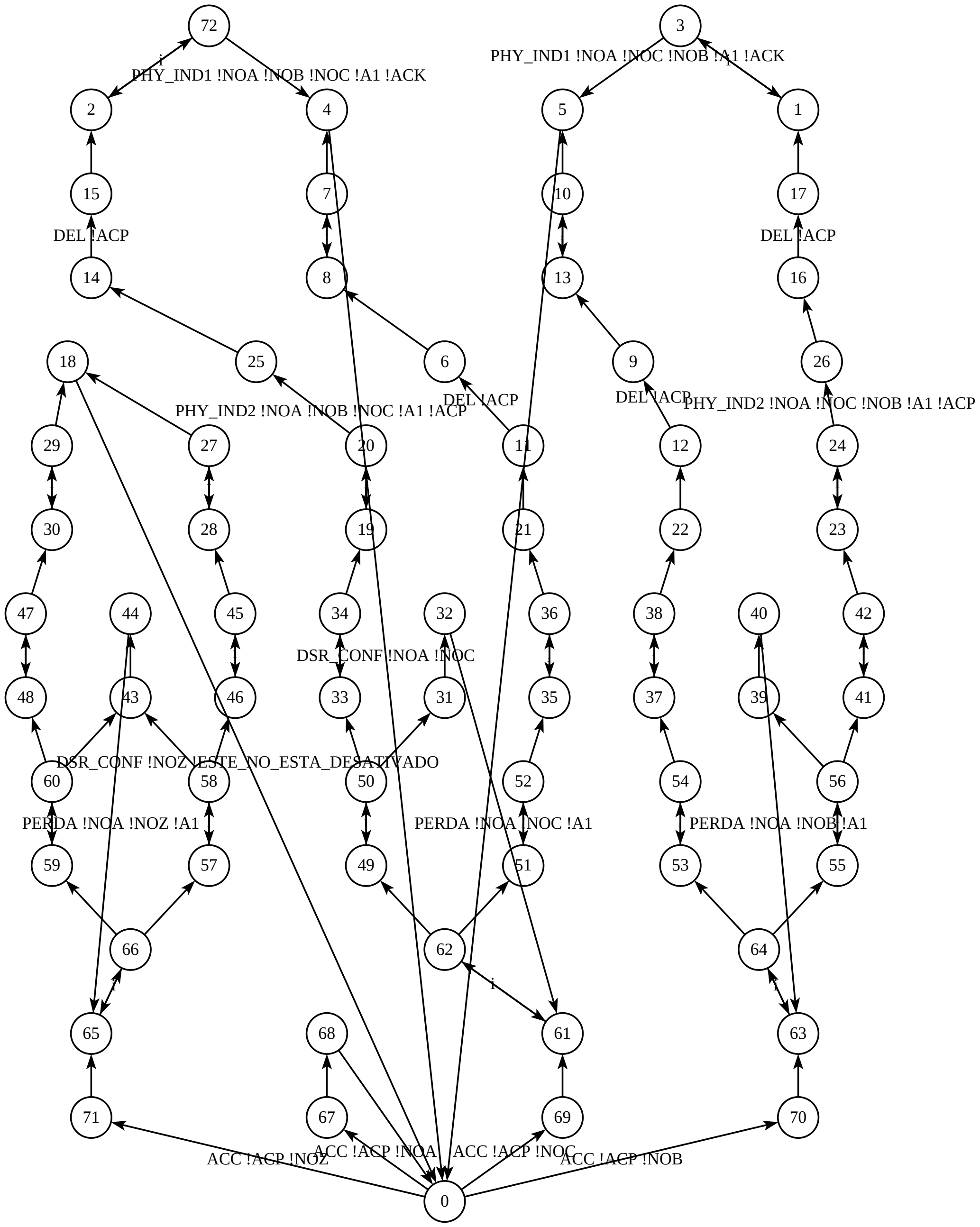

Figura 3. Sistema de Transições Rotuladas - Grafo Minimizado - 1 nó transmissor e 2 nós receptores

O identificador ID, diferentemente do referido na documentação da IETF, só será necessário na montagem das tabelas de rotas na memória dos nós participantes da rede, identificando cada entrada na tabela, em especial para o caso de haver mais de uma rota para um mesmo par nó origem/nó destino.

O aprendizado adquirido a partir desta série de experimentos é a apresentação da primeira metodologia de especificação e verificação de uma arquitetura de protocolos para redes móveis *ad-hoc* empregando técnicas de descrição formal, considerando, no melhor de nosso conhecimento, a inexistência de outras propostas com tal finalidade. Além disso, destaca-se o princípio incremental desta metodologia, que conduz, mediante inserção gradual de funcionalidades, à especificação de modelos completos de nós para tais aplicações.

7 Comentários Finais

Este trabalho apresenta a proposta de uma arquitetura completa de suporte à descoberta de serviços em redes móveis *ad hoc*, valendo-se de uma metodologia de especificação e verificação de propriedades que utiliza técnicas de descrição formal. Tais técnicas permitem uma detecção rápida de erros durante a concepção das aplicações, contribuindo para a obtenção de sistemas de melhor qualidade. Mecanismos são igualmente propostos no modelo para prover a redução da complexidade matemática por explosão de estados alcançáveis e a representação de restrições temporais.

A arquitetura obtida resultou de um estudo das características e restrições de redes móveis *ad hoc*, tendo sido adotados os protocolos SLP [2], ADTCP [3], DSR [4] e MACAW [5]. Em especial, apresentam-se as especificações em LOTOS e verificação de propriedades dos protocolos MACAW e DSR, o que mostrou as potencialidades da ferramenta utilizada, a viabilidade do experimento e a correção da especificação do protocolo frente ao modelo requerido para o serviço.

Como passos futuros, as especificações dos demais protocolos da arquitetura serão descritas e submetidas à mesma abordagem empregada aos protocolos estudados, não só reavaliando a forma de verificação, mas em especial a própria modelagem da rede. Outras ferramentas de verificação de propriedades, relacionadas à lógica temporal podem ser utilizadas. Assim, será possível avaliar a correção desta primeira arquitetura proposta ou mesmo sugerir alterações nos protocolos apresentados na literatura e utilizados neste trabalho.

Referências

1. S. Corson e J. Macker, "Mobile ad-hoc networking (manet): Routing protocol performance and issues and evalation considerations", *RFC 2501*, 1999.
2. E. Guttman, C. Perkins, J. Veizades e M. Day, "Service location protocol, version 2", 1999.
3. Z. Fu, X. Meng e S. Lu, "Design and implementation of a tcp-compatible transport protocol for ad hoc wireless networks", 2001.
4. D. B. Johnson, D. A. Maltz, Y.-C. Hu e J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks". trabalho em andamento.

5. V. Bharghavan, A. J. Demers, S. Shenker e L. Zhang, "MACAW: A media access protocol for wireless LAN's", in *SIGCOMM*, pp. 212–225, 1994.
6. ISO/IEC, "Is 8807: Information processing systems – open systems interconnection – lotos – a formal description technique based on the temporal ordering of observational behaviour", 1988.
7. J.C.Fernandez, H. Garavel, A. Kerbrat, L. Mounier, R. Mateescu e M. Sighireanu, "CADP: a protocol validation and verification toolbox", in *Proceedings of the Eighth International Conference on Computer Aided Verification CAV* (Rajeev Alur e Thomas A. Henzinger, eds.), vol. 1102, New Brunswick, NJ, USA, pp. 437–440, Springer Verlag, / 1996.
8. K. Obraczka e G. Tsudik, "Multicast routing issues in ad hoc networks", 1998.
9. Microsoft Corporation, "Universal plug and play device architecture reference specification", 1999.
10. K. Arnold, A. Wollrath, B. O'Sullivan, R. Scheifler e J. Waldo, *The Jini specification*. Reading, MA, USA: Addison-Wesley, 1999.
11. The Salutation Consortium Inc., "The salutation consortium inc. salutation architecture specification", 1999.
12. Bluetooth SIG, "Service discovery protocol", 2000.
13. V. D. Park e M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks", in *INFOCOM (3)*, pp. 1405–1413, 1997.
14. H. Balakrishnan, S. Seshan, E. Amir e R. Katz, "Improving tcp/ip performance over wireless networks", in *Mobicom95*, 1995.
15. P.Sinha, N.Venkitaraman, R.Sivakumar e V.Bharghavan, "Wtcp: A reliable transport protocol for wireless wide-area networks", in *Mobicom99*, 1999.
16. S. Floyd, "Tcp and explicit congestion notification", in *ACM CCR*, 1994.
17. H. Balakrishnan e R. Katz, "Explicit loss notification and wireless web performances", in *Globecom98*, 1998.
18. G. Holland e N.Vaidya, "Analysis of tcp performance over mobile ad hoc networks", in *Mobicom99*, 1999.
19. J.Liu e S.Singh, "Atcp: Tcp for mobile ad hoc networks", in *To appear in IEEE J-SAC in 2001*, 2001.
20. R. G. Ogier, F. L. Templin, B. Bellur e M. G. Lewis, "Topology broadcast based on reverse-path forwarding (tbrpf)". trabalho em andamento.
21. M. Gerla, X. Hong e G. Pei, "Fisheye state routing protocol (fsr) for ad hoc networks". trabalho em andamento.
22. C. E. Perkins, E. M. Belding-Royer e S. R. Das, "Ad hoc on-demand distance vector (aodv) routing". trabalho em andamento.
23. T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum e L. Viennot, "Optimized link state routing protocol". trabalho em andamento.
24. M. Gerla, X. Hong, L. Ma e G. Pei, "Landmark routing protocol (lanmar) for large scale ad hoc networks". trabalho em andamento.
25. R. Rozovsky e P. R. Kumar, "Seedex: A mac protocol for ad hoc networks", 2001.
26. E. Brinksma e T. Bolognesi, "Introduction to the ISO specification language LOTOS", *Computer Networks and ISDN Systems*, vol. 14, no. 1, no. 1, 1987.
27. R. Milner, "Communication and concurrency", 1989.
28. H. Ehrig e B. Mahr, "Fundamentals of Algebraic Specifications", in *Monographs on Theoretical Computer Science 1 (2)*, vol. Volume 6 (21), Springer-Verlag, 1985 (1990).
29. R. Mateescu, "Formal description and analysis of a bounded retransmission protocol", Tech. Rep. RR-2965, INRIA Rhône-Alpes, 1996.