UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

DANIEL SALVI WUNDERLICH

SIMULAÇÃO E ANÁLISE DO PROTOCOLO DE ROTEAMENTO ANÔNIMO ANODR PARA REDES SEM FIO AD HOC MÓVEIS

FLORIANÓPOLIS - SC, 2009

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

DANIEL SALVI WUNDERLICH

SIMULAÇÃO E ANÁLISE DO PROTOCOLO DE ROTEAMENTO ANÔNIMO ANODR PARA REDES SEM FIO AD HOC MÓVEIS

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação.

Orientador:
Prof. Dr. João Bosco Mangueira Sobral

Membros da Banca:
Prof. Fernando Augusto da Silva Cruz
Lucas Guardalben

FLORIANÓPOLIS - SC, 2009

Daniel Salvi Wunderlich

SIMULAÇÃO E ANÁLISE DO PROTOCOLO DE ROTEAMENTO ANÔNIMO ANODR PARA REDES SEM FIO AD HOC MÓVEIS

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: _____
Prof. Dr. João Bosco Mangueira Sobral

Banca examinadora

_____
Prof. Fernando Augusto da Silva Cruz

_____
Lucas Guardalben

It is never too late to give up our prejudices - Thoreau

AGRADECIMENTOS

A minha família, sem a qual nada faria sentido.

Aos meus amigos quase irmãos, por acreditarem – e não acreditarem também.

Alexandra Lima, por existir.

RESUMO

Redes sem fio ad hoc móveis (MANET – *Mobile Ad Hoc Network*) são estruturas caracterizadas por equipamentos que se comunicam entre si sem o uso de uma estrutura central responsável pelo seu gerenciamento. Seu uso comercial nos últimos anos vem crescendo significativamente, principalmente devido à flexibildade e ao baixo custo envolvido. Entretanto, vários aspectos de segurança ainda se apresentam como uma ameaça no sentido de se adotar um padrão comercial de um protocolo de roteamento anônimo. Esse estudo procurou levantar alguns aspectos de segurança e simular discretamente um dos protocolos descritos e comparados por Tamashiro (2007). Em termos gerais, a pesquisa consistiu em estudar o funcionamento do protocolo anônimo ANODR e verificar as propriedades de anonimato utilizando o simulador de redes QualNet. A simulação permitiu assim analisar as propriedades teóricas com as saídas do simulador.

PALAVRAS-CHAVE: *Redes Ad hoc, Anonimato, MANET, ANODR.*

ABSTRACT

Mobile Ad Hoc Networks (MANET) are structures formed by equipments that communicate with each other without using a central unit for its management. The commercial use in the last few years has been remarkably increased, mainly due to its flexibility and low cost involved. However, many security aspects still stands as a threat toward adopting a commercial standard for an anonymous routing protocol. This paper showed some related security topics and informally simulated a previously choosen protocol, based on a comparative study by Tamashiro (2007). To sum up, this research was intended to study the anonymous protocol ANODR and to verify its anonimity properties by running some simulations over Qualnet Simulator. Then some theorical properties could have been analysed in comparison with Simulator outputs.

KEYWORDS: *Ad Hoc Networks*, *Anonymity*, *MANET*, *ANODR.*

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE ABREVIATURAS E ACRÔNIMOS

AD HOC – Do latim, “para isto”

ANODR – Anonymous On Demand Routing

AODV – Ad-hoc On demand Distance Vector

CARP – Certificate Free Routing Protocol

CDMA – Code Division Multiple Access

DSR – Dynamic Source Routing

IP – Internet Protocol

MANET – Mobile Ad-Hoc Network

ODAR – On-Demand Anonymous Routing

SDAR – Secure Distributed Routing Protocol

TCP – Transmission Control Protocol

TDMA – Time Division Multiple Access

TORA – Temporally Ordered Routing Algorithm

UDP – User Datagram Protocol

WLAN – Wireless Local Area Network

WPAN – Wireless Personal Area Network

SUMÁRIO

1 INTRODUÇÃO

Redes sem fio ad hoc móveis (MANET – *Mobile ad-hoc Networks*) são estruturas caracterizadas por equipamentos que se comunicam entre si sem o uso de uma estrutura central responsável pelo seu gerenciamento. Esses dispositivos possuem uma habilidade de roteamento que possibilita uma rápida reorganização da topologia da rede e uma liberdade de locomoção que independe de estruturas fixas. Essa flexibilidade, portanto, habilita seu uso em diversas aplicações, como operações militares e de resgate. Embora a dispensa de um ou mais equipamentos fixos de controle seja um ponto altamente positivo, a sua ausência incorpora uma preocupação grande com a segurança dos protocolos de roteamento utilizados, já que abordagens tradicionais não garantem os requisitos mínimos de anonimato (identidade, localização e rota).

Com base no estudo comparativo de diversos protocolos apresentado por TAMASHIRO (2007), pretende-se realizar uma simulação discreta de um protocolo com intuito de validar as comparações teóricas apresentadas no estudo. A escolha do protocolo foi definida principalmente pela combinação positiva dos aspectos de segurança. Assim, o protocolo escolhido foi o ANODR (KONG & HONG, 2003). Este foi incorporado ao simulador de redes Qualnet (SNT, 2009), onde foi possível utilizar componentes móveis de rede sem-fio para criar um cenário de simulação. Ainda referente aos aspectos de segurança, será apresentada uma fundamentação teórica das redes propostas e suas aplicações. Por fim, será possível fazer um estudo comparativo entre a proposta teórica (um tanto intuitiva) e o resultado da simulação.

1.1 Objetivos

1.1.1 Objetivos Gerais

- ✓ Apresentar estudos recentes e conceitos importantes sobre redes ad-hoc móveis;
- ✓ Simular um protocolo de roteamento anônimo.

1.1.2 Objetivos Específicos

- ✓ Apresentar uma contextualização teórica das redes ad-hoc móveis e dos protocolos de roteamento anônimo;
- ✓ Apresentar com detalhes um protocolo de roteamento anônimo;
- ✓ Simular um protocolo usando uma ferramenta de simulação discreta;
- ✓ Analisar os resultados da simulação.

1.2 Delimitação do Escopo

De forma geral, esse trabalho será baseado no estudo comparativo dos protocolos de roteamento anônimo realizado por TAMASHIRO (2007). Existe uma limitação quanto ao tipo de ataque, sendo que a análise realizada leva em conta apenas ataques passivos. Pretende-se ainda analisar e simular apenas um protocolo avaliado, assim sua escolha pode não ser a melhor entre as disponíveis (pelo menos não necessariamente).

Ainda com relação aos aspectos de segurança, o estudo não irá questionar possíveis ataques ou falhas de segurança em outras camadas (enlace, transporte e aplicação), já que o foco é a camada de rede (roteamento).

1.3 Metodologia

Para a realização desse trabalho, procurou-se seguir a seguinte metodologia:

- ✓ Pesquisar conceitos e aplicações referentes às redes MANET;
- ✓ Apresentar os conceitos e os detalhes de funcionamento do protocolo escolhido;
- ✓ Estudar a ferramenta para simulação e apresentar conceitos relacionados;
- ✓ Simular um protocolo previamente definido;
- ✓ Apresentar os resultados.

2 REDES SEM FIO AD HOC MÓVEIS

Este capítulo apresenta um estudo teórico de algumas definições relevantes para a compreensão das redes sem fio ad-hoc móveis. É feita uma breve caracterização dessas redes, relacionado tais asptectos com as aplicações e implicações de segurança. Por fim, alguns protocolos tradicionais de redes ad-hoc móveis são brevemente caracterizados.

2.1 Visão Geral

Redes sem fio ad hoc móveis são redes formadas por equipamentos móveis que de alguma forma mantém uma comunicação entre si, sem o uso de estruturas fixas (*switches, backbones*, etc.) e sem o uso de enlaces com fio. Além da estrutura, a topologia também não é fixa, ou seja, todos os nós da rede são responsáveis por organizar e controlar a rede, resultando em uma topologia dinâmica. Os nós então são responsáveis por descobrir, dinamicamente, com quais podem se comunicar diretamente e por encaminhar pacotes, cujos destinos não estão no raio de alcance de suas origens (TAMASHIRO, 2007 *apud* HAAS *et al.*, 1999). Na figura 1 é possível perceber essa idéia. Enquanto que os nós A, B e C podem se comunicar diretamente, a comunicação com D depende do roteamento através de C.

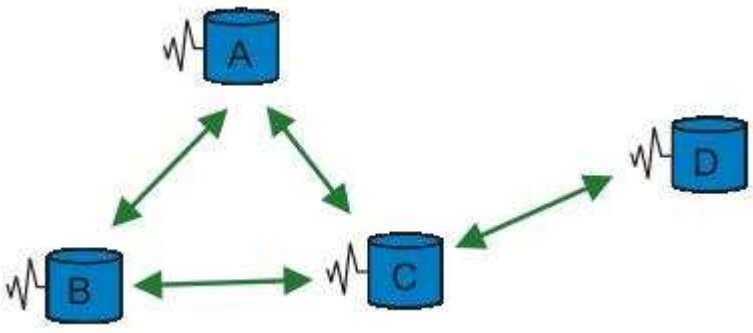

Figura 1: Rede Sem Fio Ad Hoc Móvel

Ainda em relação à natureza de operação da rede, pode-se dizer que o modelo é um sistema autônomo de nós móveis. Segundo CORSON & MACKER (1999), "*uma rede MANET é constituída por nós que são livres para mover-se arbitrariamente*", e ainda que "*o sistema pode operar isoladamente, ou pode ter gateways ou interfaces com uma rede fixa*".

2.2 Características e Aplicações

As principais características e aplicações, segundo CORSON & MACKER (1999), estão relacionadas com a capacidade de mobilidade e dinamismo da topologia. Os nós são equipados com transmissores e receptores variados, o que caracteriza um modelo altamente assimétrico, já que a capacidade de transmissão de um nó pode variar em relação a outro dentro de uma mesma área de alcance. Isso indica que um nó A poderia comunicar-se diretamente com C, mas o nó C, devido ao seu transmissor de curto alcance, necessitaria de um nó intermediário para fazer o roteamento. As principais características apontadas são:

- ✓ **Topologia dinâmica**: se os nós são livres para mover-se de forma arbitrária, a topologia deve se adequar dinamicamente de tempos em tempos;

- ✓ **Largura de banda reduzida e condições do enlace variáveis:** as redes sem fio naturalmente possuem menor capacidade que estruturas físicas, levando em conta ainda que outros efeitos do meio (ruídos, interferências, etc.) limitam a capacidade de uso ótimo das taxas de transmissão sem fio. E como redes sem fio demandam normalmente pelas mesmas aplicações que as utilizadas em redes cabeadas, deve-se considerar normal operar em piores condições (não deve ser uma exceção);

- ✓ **Restrição de energia**: alguns nós na rede podem não ter as mesmas condições de energia que outros, implicando na adoção de critérios para simplificar as operações e assim diminuir o consumo de energia;

- ✓ **Limite na segurança física:** as redes sem fio ad hoc móveis são mais vulneráveis aos ataques passivos e ativos, ao contrário das redes estruturadas, onde os equipamentos fixos centralizam a segurança. É necessário aplicar técnicas para diminuir essas ameaças. Porém uma vantagem dessa estrutura é a redução dos pontos de falha.

Outras características são citadas por TAMASHIRO (2007) *apud* CHLAMTAC *et al*. (2003) e MURTHY & MANOJ (2004), mais específicas em relação a instalação:

- ✓ **Autonomia e ausência de infra-estrutura**: são redes auto-organizadas e não dependem de estrutura física centralizada para gerenciamento e manutenção. Os pontos de falham diminuem, mas sua detecção se torna mais difícil;

- ✓ **Roteamento de múltiplos saltos:** Os nós funcionam como roteadores quando o pacote para certo destino está em seu raio de alcance, mas o de origem não (encaminhamento de pacotes);

- ✓ **Instalação rápida e baixo custo:** Não há necessidade de equipamentos centrais, cabos e manutenção.

Esse conjunto de características habilita uma grande variedade de aplicações. Inicialmente o propósito eram as operações militares, estendo-se um pouco depois às de resgate. A própria tecnologia das redes sem fio ad hoc móveis, segundo CORSON & MACKER (1999), é de alguma forma associada com o termo *Mobile Packet Radio Networking* (PRNet), referente às pesquisas militares nos Estados Unidos nas décadas de 70 e 80, e com as redes WMN (Wireless Mesh Networks), organizadas conforme uma topologia MESH (WIKIPEDIA – MESH NETWORKS, 2008).

Com a popularização das redes sem fio, alguns novos propósitos incluem aplicações de cunho industrial e comercial. O amadurecimento do padrão 802.11 (IEEE 802.11, 2009) contribuiu de forma significativa para essa ampliação comercial. De certa forma pode-se estender sua utilidade para incontáveis novos propósitos (principalmente devido a sua flexibilidade), ou para melhorias em operações de resgate usando satélites e outras topologias autônomas de comunicação.

Em [TAMASHIRO, 2007] é apresentado um quadro que categoriza alguns cenários atuais e futuros das aplicações. O quadro é apresentado na tabela 1.

Tabela 1: Aplicações das Redes Sem Fio Ad Hoc Móveis

Aplicações	**Descrição**
Redes Táticas	Comunicação em operações militares Batalhas automatizadas
Redes de Sensores	Monitoramento de residências Medição de parâmetros (radiação, calor, etc.) Monitoramento de dados (ex. atividade sísmica)
Emergência	Operações de busca e resgate
Ambientes comerciais	Comércio eletrônico: serviços como pagamento em qualquer lugar Negócios: acesso dinâmico a arquivos armazendos em uma localização central, escritório móvel Veículos: transmissão de notícias, condição das estradas, tempo e música, formação de redes entre veículos próximos
Redes caseiras e corporativas	Redes sem fio locais (WLAN) Redes sem fio pessoais (WPAN)
Aplicações educacionais	Configuração de salas virtuas e de videoconferência Criação de uma rede para comunicação rápida em conferências, encontros e palestras
Redes Mesh	Zonas residenciais: acesso à Internet Auto-estradas: comunicação para os automóveis Zonas comerciais: alternativa à rede de celulares Campus universitário: rede de baixo custo
Entretenimento	Acesso à Internet em ambientes abertos Jogos entre múltiplos jogadores
Localização de serviços	Serviços de informação: localização de serviços como postos de gasolina

Fonte: TAMASHIRO (2007) *apud* CHLAMTAC *et al* (2003) e MURTHY & MANOJ (2004)

Apesar do conjunto de aplicações ser potencialmente infinito, os problemas com segurança podem retardar ou até impedir que certas características das redes ad hoc sejam realizadas na prática. Uma nova abordagem deve levar em conta requisitos fundamentais de segurança. Uma delas é o anonimato, que garante que nós participantes no roteamento não são capazes de identificar as partes envolvidas. Contudo, as pesquisas nessa área vêm crescendo significativamente nos últimos anos, e até o momento várias propostas foram apresentadas no meio acadêmico. Com a popularização de dispositivos móveis, a adoção de um padrão seguro de roteamento será necessária para estender essas funcionalidades para outras áreas, como redes interligadas de rodovias e carros, e comunicação entre eletrodomésticos.

2.3 Protocolos de Roteamento

Existem diversas abordagens tradicionais nos protocolos de roteamento para redes cabeadas que são utilizadas para redes sem fio ad-hoc móveis. Algumas dessas caraterísticas são listadas abaixo:

- ✓ **Link State:** Cada nó mantém uma visão completa da topologia da rede, calculando uma certa função-custo para cada rota. Periodicamente o nó envia por *broadcast* essa função-custo para toda a rede, onde então cada nó irá atualizar sua topologia aplicando um alogritmo para determinar o caminho mais curto de roteamento;

- ✓ **Distance Vector:** Cada nó monitora somente os links de saída, reenviando peridocamente somente para seus vizinhos. Esse método em termos computacionais é mais eficiente, mais fácil de implementar e requer menos espaço de armazenamento;

- ✓ **Source Routing**: As decisões de roteamento são tomadas na origem, onde cada pacote leva junto o caminho da rota que devem seguir.

De acordo com as características acima, dois algoritmos de roteamento vêm sendo amplamente utilizados: AODV, seguindo o padrão *distance vector*, e DSR, seguindo o padrão

source routing. Suas principais características são verificadas abaixo:

- ✓ **AODV:** Esse protocolo não requer que os nós mantenham rotas para destinos que não são usados. Utiliza mensagens para manter e monitorar rotas (*Route Request*, *Route Reply* e *Route Error)*, sendo que todas mensagens são enviadas via UDP sobre cabeçalho IP comum. Mensagens de requisição são enviadas via *broadcast* quando o nós deseja encontrar uma rota. Quando mais de uma rota está disponível, vários identificadores seqüenciais únicos são mantidos, e normalmente o maior é considerado a rota mais "nova". Os nós da rota ainda mantem informações do status de conectividade do seu link, e assim que deixam de enviar essa informação o protocolo considera o link perdido;

- ✓ **DSR:** Esse protocolo foi proposto principalmente para redes ad-hoc móveis com grande quantidade de nós e muita mobilidade, pois ao trabalhar "sob demanda" não é necessário manter informações custosas da topologia da rede. Basicamente o protocolo é composto pela descoberta de rota e em sua manutenção. A solicitação de rota é feita quando uma transmissão é solicitada (sob demanda), e quando possível são mantidos *cachês* das rotas estabelecidas. Os protocolos anônimos apresentados no próximo capítulo seguem a idéia desse algoritmo;

No próximo capítulo alguns aspectos de segurança são apresentados, bem como a caracterização de redes anônimas e uma visão geral de possíveis soluções para prover anonimato. Embora muitos estudos tenham sido realizados até o momento, percebe-se que a nível comercial nenhum padrão vem sendo adotado.

3 PROTOCOLOS ANÔNIMOS AD HOC MÓVEIS

Esse capítulo tem como propósito apresentar conceitos gerais de segurança de redes computacionais relativas ao anonimato no roteamento dos pacotes. Apresenta ainda algumas propostas teóricas de anonimato para redes estruturadas, que são usadas – na maioria dos casos – como base no processo de roteamento de pacotes e descoberta/manutenção de rotas nas redes ad-hoc móveis.

Mais adiante o anonimato em redes ad-hoc móveis é abordado. Em seguida alguns protocolos que tentam prover anonimato são brevemente explicados.

3.1 Aspectos de Segurança e Anonimato

Ainda que o foco do estudo não se refira diretamente aos aspectos de segurança em si e no anonimato nas camadas de enlace, transporte e aplicação, é importante apresentar de forma rápida alguns conceitos.

Segundo TAMASHIRO (2007), "*segurança em redes sem fio ad hoc móveis consiste em: disponibilidade, autenticidade, confidencialidade, integridade, não-repúdio, privacidade, anonimato, robustez, entre outros requisitos*". Assim, tais redes são mais vulneráveis a ataques, devido a sua topologia dinâmica, vulnerabilidade dos nós e dos enlaces, ausência de gerenciamento e monitoramento centralizado e limitação de recursos.

Por anonimato, segundo a especificação ISO/IEC 15408 (TAMASHIRO, 2007), são apontados quatro requisitos relacionados com a privacidade: *Anonimato*, que assegura a não revelação da identidade do usuário; *Pseudo-anonimato*, que assegura Anonimato, porém associa a identidade ao serviço; *Não-correlação*, que assegura a não correlação de um ou mais serviços ao usuário; e *Não-observação*, que assegura o uso de um serviço sem que outros usuários percebam que o serviço está sendo usado.

Outras definições foram aperfeiçoadas no sentido de detalhar e preencher uma eventual falta de entendimento. PFITZMANN & HANSEN (2008) vem trabalhando desde o ano de 2000 com propostas de adotar conceitos relativos ao anonimato e privacidade. É indicado ainda que a adoção de uma terminologia padrão pode contribuir para o avanço das pesquisas, evitando que cada novo pesquisador invente sua própria definição. Nesse sentido, esse estudo propõe conceitos mais amplos para definir privacidade, sendo que o anonimato não está ligado somente ao usuário, mas a qualquer *entidade* – definida como alguém ou algo

que executa uma ação. Na definição de não-correlação o atacante não consegue saber se um *item de interessente* (sob a perspectiva do atacante, sendo mais abrangente que usuário e serviço) está relacionado ou não. Um item de interesse, dessa forma, pode ser qualquer componente de interesse no ataque (pessoas, processos, mensagens, etc.). O anonimato pode ainda ser qualificado e relacionado com a não-correlação, apresentando assim: *Anonimato de origem*, quando a mensagem não pode ser associada com a origem, e vice-versa; e *Anonimato de destino*, quando a mensagem não pode ser associada com o destino, e vice-versa.

3.1.1 Roteamento anônimo

Para se consolidar redes de roteamento anônimo sem fio é necessário estabelecer e manter rotas que garantam o anonimato. Além disso, é desejável que a localização não seja revelada e que a rota de cada mensagem não seja conhecida pelos nós intermediários.

Essa abordagem tem sido bastante estudada em redes cabeadas, tendo suas origens nas redes *Mix-net* onde, segundo *BANERJEE et al.* (2006), pacotes enviados de uma origem para um destino devem passar um conjunto de misturadores (*mixes*). Um misturador, ao reordenar e encriptar a mensagem, evita que haja uma correlação entre a mensagem de entrada e a de saída. Muitas vezes são usadas redes de misturadores, cada um cifrando a mensagem com a chave de um misturador.

Já o esquema *Onion Routing* (*GOLDSCHLAG* et al., 1999) propõe uma solução semelhante para prover conexões anônimas (resistentes a espionagem e análise de tráfego) em tempo real. As mensagens são encriptadas repetidamente e enviadas através de diversas camadas de roteadores chamados *Onions*. Cada roteador remove uma camada de encriptação para saber como enviar para o próximo roteador, prevenindo que nós intermediários saibam a origem, destino e conteúdo da mensagem. A figura 2 ilustra esse modelo.

Desde 2006 tem sido desenvolvido um projeto de software (TOR PROJECT, 2006) baseado no esquem *Onion Routing* que busca aumentar o nível de anonimato na Internet, apesar de não impedir diversos tipos de ataques.

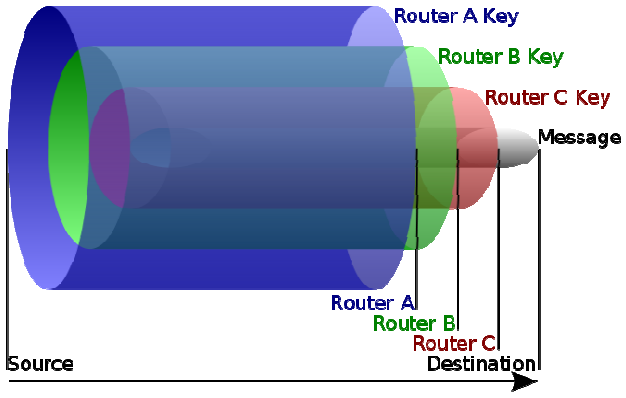

Figura 2: Esquema das camadas no *Onion Routing*

3.1.2 Ataques de Análise de Tráfego

Segundo TAMASHIRO (2007), através dos ataques de análise de tráfego um atacante pode inferir informações úteis sobre os nós da rede, localização, topologia, frequência de comunicação e padrões de movimento. São destacados os seguintes ataques:

- ✓ **Ataque de análise de tempo**: se os pacotes são processados e encaminhados na mesma ordem em que são recebidos, um atacante pode inferir quais pertencem à mesma rota;

- ✓ **Ataque de conteúdo do pacote**: quando o conteúdo do pacote permanece inalterado durante uma transmissão, o atacante pode seguir o pacote;

- ✓ **Ataque de volume do pacote**: quando pacotes possuem o mesmo tamanho ou se o tamanho muda de forma padronizada, o atacante pode seguir o pacote;

- ✓ **Ataque de reconhecimento de fluxo**: diz respeito à capacidade que um atacante tem de identificar pacotes referentes à mesma rota, através de análise de tempo por exemplo.

3.2 Anonimato em Redes Ad-Hoc Móveis

Em um nó de uma rede ad-hoc móvel o roteamento ocorre na camada de rede (assim como em outras redes), que é responsável por manter rotas entre diversos nós e possibilitar assim a troca de mensagens. Como a topologia é dinâmica (pode haver mudança de localização, topologia, nós, etc.), os algoritmos tradicionais não podem ser usados nesses modelos de rede.

Em KUOSMANEM (2007) são avaliados diversos protocolos para redes sem fio ad-hoc móveis segundo alguns critérios e taxonomias. Dessa forma, ele caracteriza os protocolos da seguinte forma:

- ✓ **Modelo de comunicação**: divisão feita (*CDMA*, *TDMA*, etc.) de acordo com o modelo de comunicação para qual o protocolo foi projetado (*multi-channel* ou *single-channel*);

- ✓ **Estrutura**: os nós na rede podem estar separados uniformemente ou de acordo com alguma hierarquia. Protocolos uniformes tratam os nós da mesma maneira, enquanto que os protocolos não-uniformes aperfeiçoam as rotas de acordo com o nó. O segundo modelo é ainda divido em *neighbor selection* e *partition*, de acordo com a maneira que as rotas são gerenciadas.

- ✓ **Informação do estado**: protocolos baseados em topologia (*Topology-based*) levam em conta que os nós possuem as informações da topologia. Protocolos baseados no destino (*Destination-based*) mantêm informações apenas dos vizinhos próximos.

- ✓ **Agendamento**: classificação feita de acordo com a periodicidade na busca de informações sobre a rota. Os modelos pró-ativos verificam periodicamente a descoberta de novas rotas, enquanto os modelos sob demanda calculam a informação da rota conforme sua necessidade.

- ✓ **Tipo de transmissão**: essa divisão localiza-se no topo das taxonomias anteriores,

> já que para os modelos *unicast* tais classificações podem ser são aplicadas, e para outros não. Modelos *unicast* são comuns em redes ad-hoc, onde um pacote é endereçado sempre a um único destinatário. Os modelos *multicast* lidam com a necessidade de enviar um pacote para múltiplos destinatários. Os modelos *geocast* são semelhantes, porém enviam pacotes para destinatários localizados em regiões geograficamente distantes.

Ainda em relação a esse estudo, diversos protocolos conhecidos (AODV, DSR, TORA, OLSR, CEDAR) são avaliados segundo essa classificação. Para isso é definida uma função custo-benefício. Não se pode considerar apenas o número de nós, mas outros fatores determinantes, como o tamanho da banda, a capacidade do enlace de rede (link), a latência, etc.

O intuito aqui é destacar a importância de considerar a natureza de aplicação da rede, que pode ser determinante na escolha do modelo. Em uma rede militar, por exemplo, fatores como baixo ruído, interferência e anonimato devem ser levados em conta. Já em uma rede de sensores doméstica o protocolo deve lidar com a baixa capacidade de energia dos dispositivos.

Embora os diversos protocolos existentes sejam adequados para várias aplicações, um anonimato eficiente na comunicação permanece um tópico aberto. Abordagens semelhantes ao roteamento *Onion* vem sendo empregadas, porém a manutenção sobre a topologia da rede permanece complexa, ainda mais tratando-se de equipamentos sem muitos recursos (em alguns casos) situados em topologias dinâmicas. É muito comum usar um mecanismo de *trapdoor* para coletar a rota de origem entre dois pares de comunicação. Um *trapdoor* é um conceito amplo em criptografia, onde é definida uma função unidirecional entre dois grupos. Se tratando de redes ad-hoc móveis, o anonimato da rota é mantido usando-se um trapdoor global, onde apenas o nó de origem e destino possui as chaves secretas para visualizar o conjunto de nós na rota. Os nós intermediários, por sua vez, vão adicionando as camadas durante a rota e desconhecem assim o caminho completo da mensagem.

Embora a maioria dos protocolos apresentados a seguir tenta garantir o anonimato, nem sempre alguns requisitos são completamente cumpridos. São apresentadas resumidamente as técnicas adotadas por cada autor e por fim uma tabela comparativa, baseada no estudo de TAMASHIRO (2007).

3.3 Visão Geral dos Protocolos

Algumas soluções para anonimato em redes sem fio ad-hoc móveis foram desenvolvidas ao longo dos últimos anos, sem que nenhuma tenha conseguido obter completo sucesso. Ainda que muitas dessas soluções sejam parecidas na utilização de mecanismos de encriptação e roteamento, alguns detalhes tornam algumas mais apropriadas para certos dispositivos ou não. A tabela 2 mostra um resumo comparativo entre os protocolos apresentados a seguir, levando em conta o anonimato de identidade, localização e padrão de movimento, de rota e de *venue* (se os nós sabem a sua distância em relação aos nós origem e destino, ou seja, se os pacotes possuem contadores de saltos ou se o seu tamanho indica o número de saltos percorridos; se os pacotes podem ser seguidos até o nó origem ou destino (TAMASHIRO, 2007)).

3.3.1 ANODR

O protocolo ANODR (*Anonymous On-Demand Routing*) foi proposto por KONG & HONG (2003), com objetivo de garantir anonimato de rota e privacidade de localização em um ambiente de roteamento sob demanda.

O anonimato de rota é alcançado por uma abordagem de pseudônimos nos nós com informações globais de *trapdoor*. No processo de descoberta da rota um pacote é enviado por *broadcast* com a solicitação *Route Request*. A rota é formada por um conjunto de nós intremediários, em um esquema *Onion Routing*. Uma mensagem *Route Reply* é enviada pelo destinatário, quando esse consegue abrir a mensagem encriptada. A partir desse ponto, as mensagens entre origem e destino são trocados através dessa rota. O autor sugere ainda a utlização de técnicas *mixing* para evitar ataques que analisam os pacotes, e ainda o envio de pacotes falsos. Para garantir a criptografia das mensagens são usadas chaves públicas e chaves simétricas.

3.3.2 SDAR

O protocolo SDAR, proposto por EL-KHATIB *et al*. (2004), foi modelado segundo níveis de confiança entre os vizinhos em uma rede sem fio ad-hoc móvel. Esse modelo de confiança atualiza periodicamente cada nó na rota conforme seu comportamento ao longo do tempo. Assim, três níveis foram estabelecidos: baixo, médio e alto.

O nó de origem inicia um processo de estabelecimento da rota após enviar uma mensagem *broadcast* para a rede, com certo nível de confiança. Os nós intermediários que satisfazem esse nível de confiança incluem seu ID e uma *session key* na mensagem, reenviando em seguida para os nós vizinhos. Cada nó intermediário encripta a mensagem antes de adicionar seu ID. Quando o nó destino recebe a mensagem é montada uma mensagem multi-camadas (*Onion Routing*) de retorno com todo o caminho reveso da rota.

Segundo os autores o protocolo se mostra eficiente contra alguns tipos de ataques passivos e ativos, e garante, sobretudo, o anonimato dos nós de origem e destino. Usando o mecaniso de confiança, o protocolo consegue ainda garantir certa confiabilidade na rota da mensagem.

3.3.3 MASK

A idéia básica do protocolo MASK (ZHANG *et al*., 2004) é a autenticação anônima entre vizinhos baseada em um sistema dinâmico de pseudônimos (ao invés de seus endereços ou identificadores reais), e no processo anônimo de descoberta de rota e envio de mensagens, baseado na autenticação compartilhada entre vizinhos.

O protocolo procura atender cinco objetivos: anonimato de origem, destino e sua relação; *untraceability* e *unlocability* (adversários não conseguem seguir um pacote); autenticação anônima e segura entre vizinhos; baixo processamento criptográfico; e proteção contra diversos ataques, como reconhecimento de fluxo e análise de tempo.

O modelo de criptografia adotado é uma função de mapeamento bilinear com curva elíptica (BDHP – *Bilinear Diffie-Hellman Problem*). O protocolo é constiutido por quatro fases: inicialização, autenticação, descoberta da rota e encaminhamento das mensagens. As análises da simulação indicam desempenho razoável e cumprimento dos objetivos de anonimato.

3.3.4 CARP

BANERJEE *et al.* (2006) propõe o *Certificate Free Anoymous Routing Protocol* (CARP), que busca prover anonimato de identidade, rota e localização. Um dos propósitos do CARP é cumprir esses requisitos sem causar grandes custos computacionais, limitando assim o uso de criptografia com chaves simétricas e evitando verificações baseadas em autoridades certificadoras. Para isso optou por usar outra abordagem de encriptação durante a descoberta

da rota: *Identity Based Encryption (IBE)*, um algoritmo assimétrico que utiliza a própria identificação do nó como sua chave publica. Dessa forma, o nó ao enviar a mensagem não precisa compartilhar a chave pública com o destinatário, uma vez que essa é gerada de acordo com os parâmetros do algoritmo IBE (que são compartilhados na inicialização) e o próprio ID do destino.

A estrutura de cada nó inclue diversas tabelas que mantem informações do roteamento: *Route Request Token Table*, que mantem um token das rotas recebidas no nó, evitando um reenvio desnecessário, e um *timestamp* para apagar rotas inativas; *Pseudonym Table,* que guarda os pseudônimos (únicos para cada requisição de rota) dos nós e um *timestamp* para apagar pseudônimos inativos; *Routing Data Table*, responsável por armazenar as rotas do nó com outros destinos; *Route Reply Data Table*, onde são gerenciados os envios de *route reply* (indicação de encaminhamento da requisição), para finalizar o reenvio periódico de descoberta da rota; e *Data Packet Table*, mecanismo semelhante a tabela de Route Reply onde são armazenadas informações recentes de pacotes de dados passados por esse nó.

3.3.5 ODAR

Assim como no protocolo MASK, o protocolo ODAR (*On-Demand Anonymous Routing*), proposto por SY *et al*. (2006), utiliza *Diffie-Hellman* na geração das chaves secretas na origem e destino. Da mesma forma, o protocolo busca manter anonimato de identidade, rota e localização. O protocolo ODAR propõe o uso de *Bloom Filters*, estruturas para armazenamento de dados onde é possível testar se um dado elemento pertence ao grupo.

Tabela 2: Comparativo de anonimato entre protocolos

Anonimato de Identidade			
	Nó Origem	Nó Destino	Nós intermediários
ANODR	Sim	Não	Sim
SDAR	Parcialmente	Sim	Não
MASK	Sim	Não	Sim
CARP	Sim	Sim	Sim
ODAR	Sim	Sim	Parcialmente
Anonimato de *venue* dos nós origem e destino			
	Nó Origem	Nó Destino	
ANODR	Não	Sim	
SDAR	Parcialmente	Parcialmente	
MASK	Parcialmente	Sim	
CARP	Não	Não	
ODAR	Não	Não	
Anonimato de Localização e Padrão de Movimento			
	Localização	Padrão de Movimento	
ANODR	Forte	Forte	
SDAR	Parcialmente fraca	Parcialmente fraca	
MASK	Fraca	Forte	
CARP	Forte	Fraca	
ODAR	Parcialmente fraca	Parcialmente fraca	
Anonimato de Rota			
	Em relação a nós que não pertencem à rota	Em relação a nós que pertencem à rota	
ANODR	Sim	Parcialmente	
SDAR	Não	Não	
MASK	Parcialmente	Parcialmente	
CARP	Não	Não	
ODAR	Não	Não	

Fonte: TAMASHIRO (2007).

4 SIMULAÇÃO DO PROTOCOLO

Nesse capítulo será apresentada a ferramenta utilizada durante a simulação, o detalhamento do protocolo selecionado, a metodologia utilizada e os resultados da simulação.

4.1 Escolha do simulador de redes

Para realizar a simulação do protocolo optou-se pelo simulador de rede *QualNet 4.5.1*, desenvolvido pela empresa *Scalable Networks* (SNT, 2009). Antes, entretanto, outros simuladores foram analisados em termos de facilidade de uso e incorporação do código do protocolo. Entre as opções disponíveis, o simulador OPNET e o Network-Simulator (ns-2) foram amplamente estudados.

A ferramenta OPNET é uma solução completa para simulação de redes, sendo possível, através dos editores, criar rapidamente novos pacotes, nós, camadas de rede, processos, etc. Entretanto, o software mostrou-se um pouco complexo na hora de implementar a especificação informal dos protocolos, devido a ausência de suporte e documentação fornecida pelo fabricante. Com uma licensa comercial e com o suporte adequado, possivelmente essa ferramenta possui o conjunto mais completo de opções para simulação.

O simulador de redes *Network Simulator* (ns-2) é um simnulador de eventos discretos para pesquisa na área de redes. Suporta desde componentes básicos, como TCP e UDP, até mecanismos mais complexos em redes wireless e via satélite. Ainda assim, por ser gratuito e código-fonte aberto, possui uma boa documentação e contribuição de códigos. É escrito em conjunto da linguagem C++ com uma versão orientada a objetos de TCL (WIKIPEDIA TCL, 2009): OTCL (OTCL, 2009). Apesar dos recursos visuais serem infinitamente inferiores ao OPNET, o simulador *ns-2* é capaz de representar desde redes locais simples até redes mais complexas. Além disso, uma grande vantagem desse simulador é sua integração nativa com biobliotecas C++. Seria, portanto, uma ótima escolha para simular o protocolo e verificar suas propriedades.

O simulador *QualNet* apresentou uma boa documentação, uma licensa educacional e, principalmente, uma implementação do protocolo anônimo ANODR. Assim, optou-se por adaptar essa implementação para verificar as propriedades deste protocolo. Na próxima seção são apresentados alguns conceitos técnicos relativos ao simulador escolhido para o trabalho.

4.1.1 Simulador QualNet 4.5.1

O simulador *QualNet* foi utilizado na versão 4.5.1, disponibilizado para testes com uma licensa educacional. Em termos gerais, é um simulador simples de usar que procura avaliar o desempenho de redes wireless, cabeadas e mistas. Ele é dividido em módulos, cada um responsável por agrupar diferentes características de redes. Alguns deles são: Wireless, Multimídia, Celular e Satélite, Segurança e Rede de Sensores. Além das bibliotecas, o simulador é divido em componentes de simulação.

O **Editor de Cenários** permite que o usuário configure distribuições geográficas, modelos de mobilidade, conexões físicas e os parâmetros de funcionamento dos nós e da rede. Todos os parâmetros são facilmente configurados em uma estrutura hierárquica.

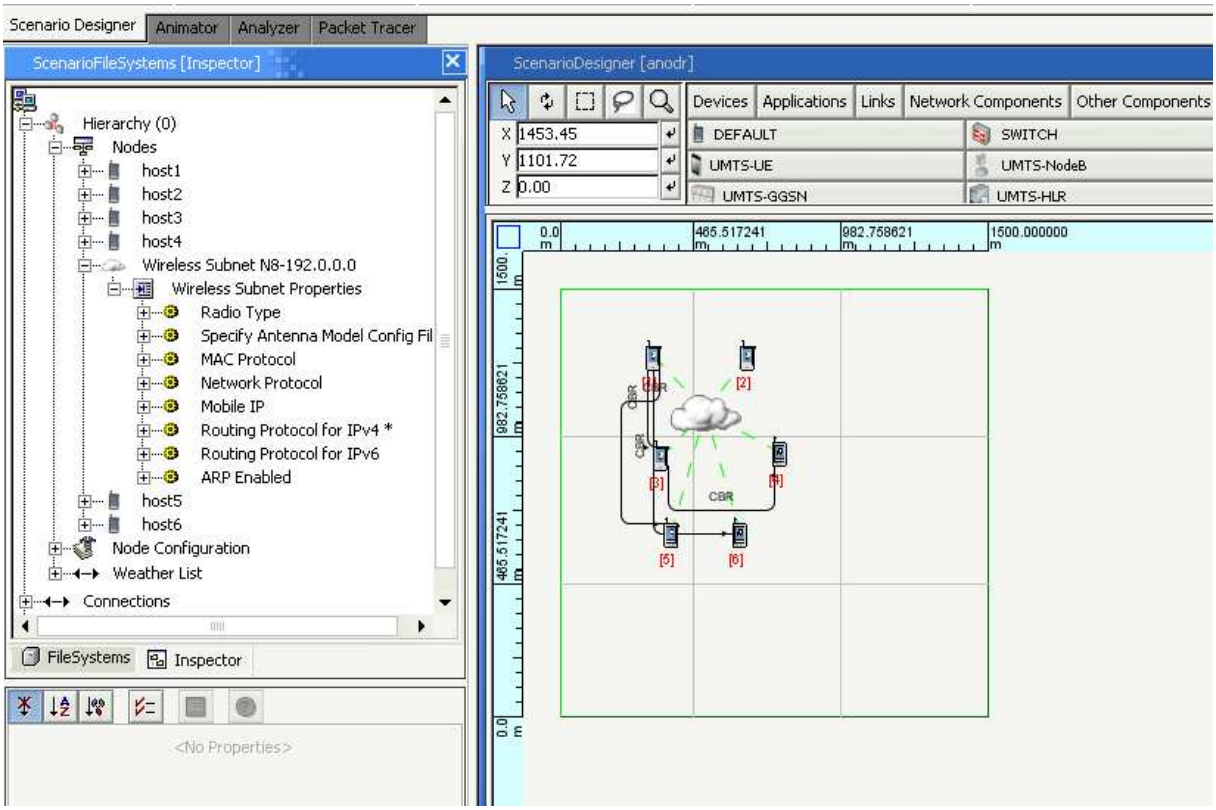

Figura 3: QualNet - Editor de Cenários do Simulador

O **Visualizador de Animação** permite visualizar de forma interativa os pacotes e faixas de alcance dos nós durante a simulação. É possível ainda usar o modo de visualização em 3D.

O **Analisador de Simulação** é uma ferramenta gráfica para exibição de centenas de

métricas estatísticas. É possível ainda criar métricas que atendam a necessidade de cada aplicação. Como opção, os gráficos podem ser visualizados em tempo real, no momento em que os dados são coletados nas interfaces dos nós.

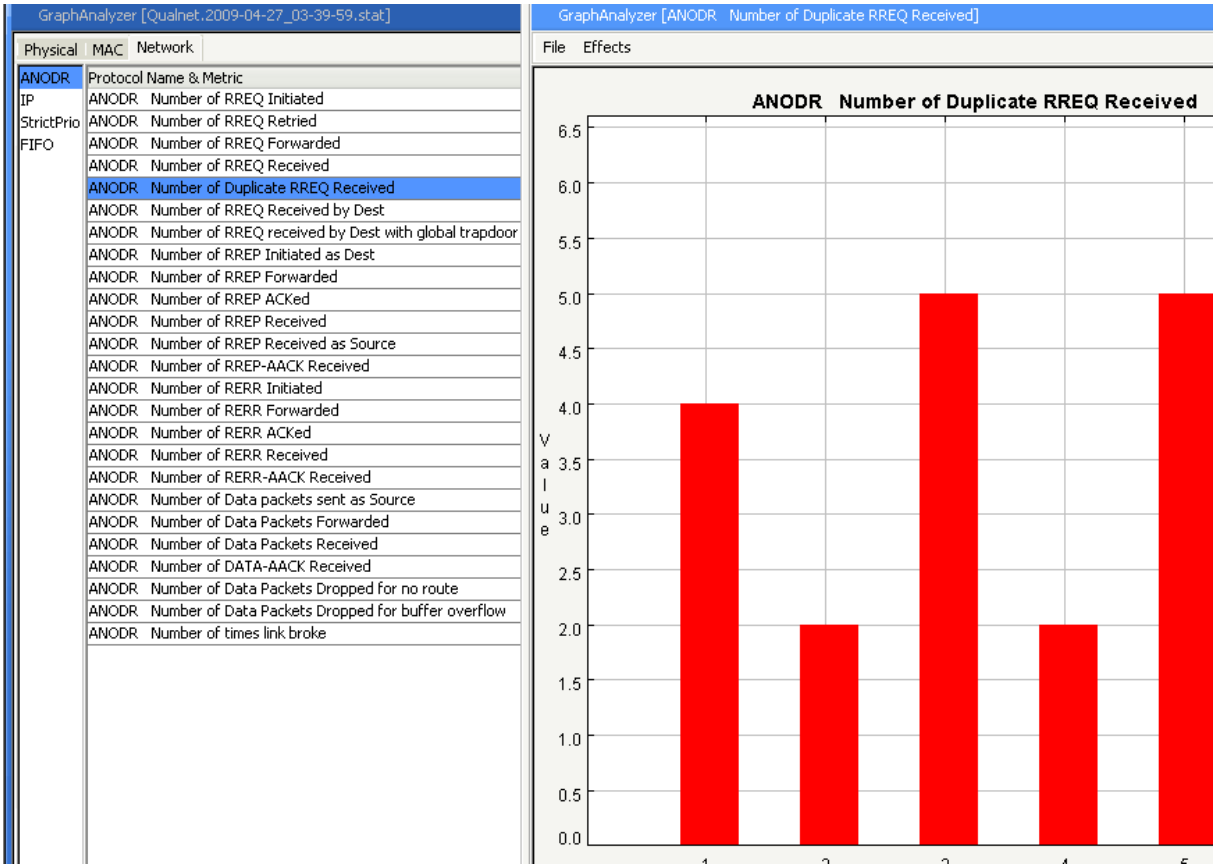

Figura 4: QualNet - Analisador de Simulação

O simulador ainda conta com o **Analisador de Pacotes,** que permite ao usuário visualizar e analisar os detalhes de cada pacote nas camadas de rede. Funciona de forma semelhante a um capturador de pacotes real, onde todo o tráfego que passa pelas interfaces de rede é salvo de acordo com o formato e campos do pacote.

Mesmo possuindo diversos recursos e bibliotecas prontas para simulação de diferentes redes, o usuário pode modificar certos comportamentos (adicionando métricas para estatísticas, por exemplo), alterando o código-fonte e compilando o simulador novamente.

O protocolo escolhido para simulação está disponível dentro da biblioteca de Segurança, sendo possível criar o cenário da rede ad-hoc e então escolher esse protocolo como roteamento de dados. O simulador disponibiliza diversas estatísticas para análise dos pacotes do ANODR, como número de requisições e número de respostas. Essas estatísticas

são exibidas na forma de gráficos dentro do simulador (figura 4). São, entretanto, insuficientes para completar o estudo comparativo e validar as propriedades de anonimato. Para tanto, certas modificações foram necessárias no código-fonte do protocolo para habilitar um conjunto mais detalhado de saídas. Na realidade, as estatísticas já disponibilizadas pelo simulador são mais úteis para análise de performance. Para validar as propriedades de anonimato o código-fonte foi alterado durante o trabalho, possibilitando assim visualizar os campos dos pacotes transitando na rede. Os detalhes de funcionamento do protocolo são mostrados na seção 4.2. A simulação e as alterações no seu código são detalhadas na seção 4.3.

4.2 Escolha do Protocolo

O protocolo escolhido para avaliação e simulação é o ANODR – *Anonymous On Demand Routing*. A escolha do protocolo foi fundamentada na (1) disponiblidade dentro do simulador e (2) nas características apresentadas, onde se observou que alguns outros protocolos posteriores a esse são apenas modificações para proteger outros ataques, ou para melhorar o desempenho quando se trata de equipamentos móveis de pouco poder computacional. Será apresentada uma visão geral do protocolo e os detalhes específicos de funcionamento.

4.2.1 Visão Geral do Protocolo

O protocolo ANODR, proposto por KONG & HONG (2003), foi o primeiro protocolo para roteamento anônimo em redes ad-hoc móveis. Em termos gerais, procura prover anonimato de rota e privacidade de localização, garantindo que os adversários não possam descobrir a identidade real das partes envolvidas em uma transmissão de rede. O planejamento do protocolo é baseado em *broadcast* com informações de *trapdoor*.

O propósito do protocolo é desenvolver um esquema de rotas que não possam ser traçadas, dentro de um ambiente de roteamento sob demanda. Esse objetivo é, substancialmente, diferente de outras propostas de roteamento seguro, onde procura-se prevenir outros tipos de ataque, como o de negação de serviço. Dentro do ambiente que o protocolo é projetado, os atacantes tentam agir passivamente em silêncio, procurando ficar o mais invisível possível.

Dessa forma, a contribuição do trabalho é tentar apresentar um protocolo de roteamento que previne um atacante associar os participantes da rede com suas identidades, e impedir que o fluxo de um pacote possa ser seguido no destino ou origem. E embora os adversários possam detectar a existência de transmissões de rede sem fio, fica difícil saber o número de participantes e os padrões de transmissão. O anonimato não é referenciado, pois, em termos de não-observância.

O protocolo é constituído por três fases: descoberta de rota, encaminhamento de dados e manutenção de rota. Para facilitar o entendimento dos conceitos apresentados nas próximas seções, a tabela 3 mostra um conjunto genérico de notações e variáveis.

Tabela 3: Notação utilizada pelo protocolo ANODR

PK_A	Chave pública do nó A
SK_A	Chave privada do nó A correspondente a PK_A
$\{M\}_{PKA}$	Encriptação/verificação da mensagem M usando a chave PK_A
$[M]_{SKA}$	Decriptação/assinatura da mensagem M usando a chave privada SK_A
K(M)	Encriptação/decriptação da mensagem M usando a chave simétrica K
src	Tag especial que denota a origem
dest	Tag Especial que denota o destino
K_A	Chave de encriptação conhecida somente pelo nó A
K_{AB}	Chave de encriptação compartilhada entre o nó A e o nó B
N_A, N^i_A	Nonce (*Number once* – número utilizado uma única vez) ou nonces criados pelo nó A
RREQ	Identificador do pacote de solicitação de rota
RREP	Identificador do pacote de resposta de rota
REER	Identificador de erro do pacote de manutenção de rota

4.2.2 Modelos criptográficos do protocolo

O conceito de misturadores (**MIX-net**) é utilizado por vários protocolos que buscam prover anonimato no roteamento de mensagens. Supondo que a mensagem *m* precisa ser enviada da origem *S* até o destino *D* através de um misturador *M*, a entrada para a rede será da seguinte forma:

$$\{ D, N^1_S, \{m, N^0_S\}PK_D\}PK_M \}$$

Dessa forma, somente M consegue decriptar a mensagem e encaminhar a mensagem

para o destino D. Quando a mensagem passa por um conjunto de misturadores a estrutura passa a funcionar em camadas (*Onion Routing*), cada misturador retira uma camada externa e encaminha para o próximo misturador até que a mensagem chegue ao destino.

A idéia de **broadcast com *trapdoor*** funciona como uma chave para abertura da mensagem somente pelo destino. Como explicado anteriormente, *trapdoor* é uma função matemática aplicada na criptografia que é facilmente calculada em uma direção, denotada por uma função $f : X > Y$. Calcular $f(x)$ se torna trivial conhecendo a chave secreta y. No contexto do protocolo, a implementação da função de *trapdoor* depende das definições de criptografia da rede e no impacto do desempenho e *overhead* no processamento. Utilizando, por exemplo, uma chave secreta compartilhada K_T entre o nó de origem e destino, o *trapdoor* é definido como $K_T(dest, K_C)$. O *nonce* K_C é utilizado posteriormente na resposta (RREP) como uma prova de abertura do trapdoor pelo destino (*proof*). Durante a fase da descoberta da rota, a latência causada por *overheads* pode ser contornada ao configurar o nó para primeiro encaminhar a mensagem e depois tentar abrir o *trapdoor*.

4.2.3 Funcionamento do Protocolo

Algumas propostas são utilizadas como base para a definição posterior dos componentes de funcionamento:

- ✓ **Broadcasting com trapdoor**: Ao enviar o broadcast para a rede, a informação de trapdoor conhecida apenas pelo destino é associada à mensagem.

- ✓ **Privacidade de localização e não rastreabilidade**: Normalmente o roteamento de mensagens em uma rede ad-hoc móvel ocorre através de múltiplos saltos, passando por vários nós intermediários até atingir o destino. Objetiva-se, assim, garantir privacidade de localização para cada nó responsável por encaminhar mensagens, e ainda entre os nós origem/destino.

O protocolo divide o processo de roteamento em duas partes principais: **Descoberta de rota**, **manutenção de rota** e **encaminhamento de dados**.

4.2.3.1 Descoberta de Rota

A descoberta anônima de rota é um processo que ocorre no momento que um determinado nó na rede deseja enviar dados para outro nó. As rotas não são conhecidas previamente, por esse motivo o roteamento é chamado sob demanda. Durante a fase inicial da requisição o seguinte pacote é criado no nó de origem:

{ RREQ, *seqnum*, tr_{dest}, *onion* }

O primeiro campo indica o tipo de pacote que será criado. O campo *seqnum* é um identificador global único do pacote. O terceiro campo é a chave criptográfica de *trapdoor*, que pode ser aberta somente pelo destinatário. O campo *onion* representa as camadas criptográficas (*onion routing*) utilizadas pelos nós intermediários, garantindo dessa forma a rota anônima de pseudônimos. Esse campo é essencial no funcionamento do protocolo, e sua arquitetura influencia decisivamente no desempenho e garantia de anonimato. Para tal, o autor propõe três abordagens, partindo de um esquema simples de misturadores (Mix-NET), e finalmente chegando ao modelo final mais eficiente.

O primeiro modelo utiliza uma simples adaptação do esquema da rede de misturadores e chaves assimétricas. É chamado, por isso, de **ANODR-PO** (*Public key Onion*).

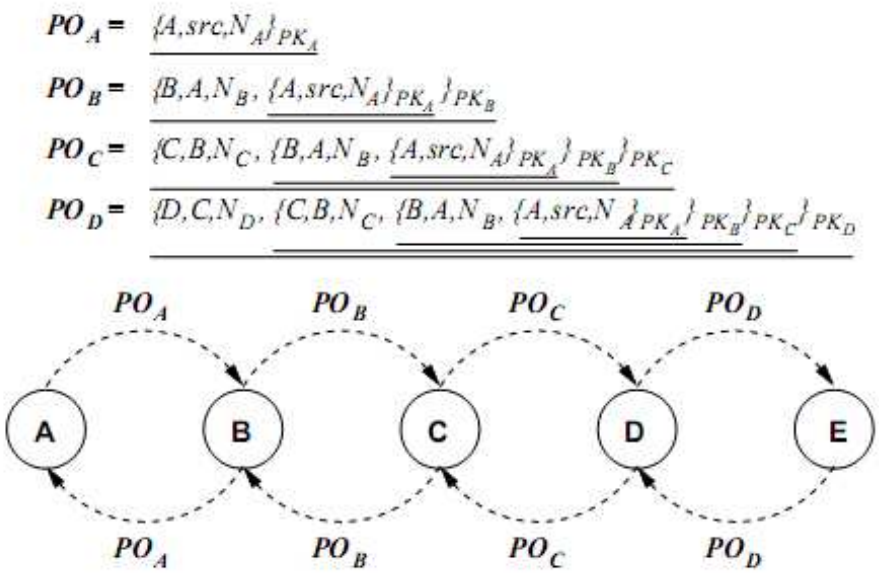

Figura 5: Descoberta anônima da rota usando o esquema ANODR-PO

Durante a fase de requisição (RREQ), cada nó na rota adiciona uma camada referente ao nó anterior e envia por *broadcast* a mensagem. O resultado da mensagem encaminhada é sempre encriptado com a chave pública do nó. Ao chegar no destino (sabe-se pela informação de *trapdoor* do pacote de requisição) a estrutura *onion* representa uma rota anônima de volta ao nó de origem. A mensagem de resposta possui o seguinte formato:

{ RREP, *N*, pr_{dest}, *onion* }

O primeiro campo identifica um pacote de resposta de rota. O campo *N* é um número randômico único que representa o pseudônimo da rota. O terceiro campo é uma prova de abertura do *trapdoor* pelo destino. O *onion* desse pacote é uma cópia exata do onion que chega ao nó de destino.

Como mostrado na figura 5, durante a resposta de rota o nó tenta decriptar o *onion* utilizando sua chave privada. Quando não encontra sua identificação na mensagem decriptada (primeiro campo), o nó não faz parte da rota e a mensagem é descartada. Ao perceber que a rota deve passar pelo nó, é gerado um *nonce* N^i que substitui o nonce *N* da mensagem. Uma correspondência interna da relação N pra N^i é mantida na tabela interna de encaminhamento. Em seguida, a camada externa do *onion* é retirada e a mensagem é encaminhada.

O esquema de chaves assimétricas do modelo ANODR-PO prejudica o desempenho da rede em termos de latência e *overhead*. Imaginando que as mensagens de RREQ e RREP são enviadas via *broadcast*, toda a rede ficará comprometida em executar operações computacionais custosas para montar e verificar os *onions* das rotas. Assim, é proposto um esquema com chaves simétricas, chamado de **ANODR-BO** (*Boomerang Onion*) – em referência ao uso da mesma chave simétrica usada na requisição e depois na resposta.

Na figura 6 é possível verificar o esquema de roteamento usando a idéia das chaves simétricas. Quando o nó B intermediário recebe uma requisição para encaminhamento, ele adiciona uma camada ao *Boomerang Onion* e encripta o resultado com uma chave simétrica

randômica K_B. Quando a resposta retorna ao nó, a mesma chave é utilizada para remover uma camada do esquema. Assim é garantido que a latência não será grande, pelo menos se depender das operações criptográficas executadas durante a requisição e resposta. O uso de chaves simétricas tem sem mostrado muito eficiente em equipamentos móveis de reduzido poder computacional.

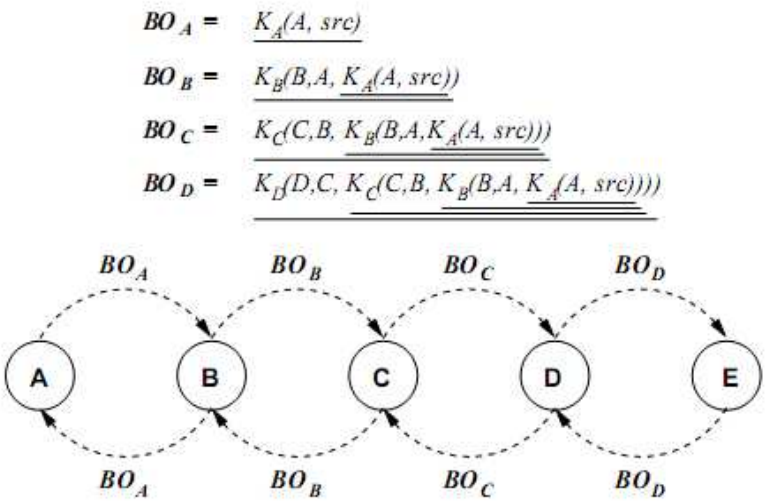

Figura 6: Descoberta anônima da rota usando o esquema ANODR-BO

Uma modificação é proposta ao esquema do ANODR-BO para chegar ao modelo final. Como a identidade dos nós intermediários pode ser encontrada nas mensagens, o anonimato não é alcançado entre os vizinhos da rota. É adicionado, então, um *trapdoor* nas camadas. Quando um nó intermediário B recebe a mensagem de requisição de rota, ele gera um *nonce* N_B e adiciona na camada. Da mesma forma do esquema anterior, a mensagem é encriptada usando uma chave simétrica antes de ser enviada via *broadcast* pela rede. A informação de *trapdoor* é a chave simétrica do nó e o *nonce* gerado pelo nó. Durante a resposta de rota o nó será capaz de decriptar a mensagem e confirmar o *nonce* com seu registro interno. A figura 7 ilustra esse processo, chamado de **ANODR-TBO** (*Trapdoor Boomerang Onion*).

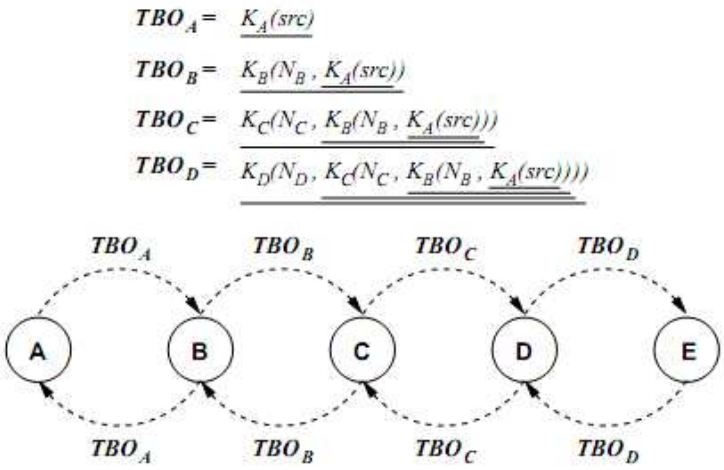

Figura 7: Descoberta anônima da rota usando o esquema ANODR-TBO

4.2.3.2 Manutenção de Rota

A manutenção de rota no protocolo ANODR segue um esquema de atualização das tabelas de roteamento. Após certo número de tentativas de retransmissões de dados, o nó verifica o pseudônimo N' que está associado com o pseudônimo N do provável nó defeituoso (desligado ou com defeito). Assim, uma mensagem de erro é enviada no seguinte formato: {REER, N'}. Os nós que recebem via broadcast essa mensagem e verificam que estão usando essa rota, realizam o mesmo procedimento em cascata para notificar os vizinhos.

4.2.3.3 Encaminhamento de Dados

Após receber a resposta de rota, o nó de origem encripta os dados utilizando o pseudônimo de rota da sua tabela de encaminhamento. O pacote é então enviado via *broadcast* para a rede e, para cada nó intemediário, verifica-se a tabela de encaminhamento para encontrar uma correpondência com o pseudônimo. Caso pertença à rota, o nó altera o pacote e inclui o seu pseudônimo. Esse procedimento é repetido até o nó chegar ao destino.

4.2.4 Considerações Sobre a Implementação

Algumas questões são levantadas pelo autor em relação a analise de pacotes feita por um atacante (*eavesdropper*, por exemplo), que pode associar requisições com respostas de rota. Em termos computacionais, dificilmente alguém que não possui a chave de abertura do *onion* consegue relacionar a estrutura de um *onion* (os pseudônimos das rotas, por exemplo)

encriptado com o correspondente sem encriptação. Entretanto, algumas associações são possíveis em outros campos: (1) pacotes de resposta de rota (RREP) com a mesmo prova de abertura do *trapdoor* (pr_{dest}) podem facilmente serem associados com a mesma rota. (2) Pacotes de requisição de rota (RREQ) com o mesmo sequencial (*seqnum*) e *trapdoor* podem pertencer à mesma rota. Ainda assim, caso a rede esteja comprometida, a estrutura do onion entre RREP e RREQ pode ser igualada, permitindo associar os dois pacotes à mesma rota.

Para contornar alguns desses problemas, é proposto o uso de chaves assimétricas temporárias durante a fase de requisição nos nós intermediários. Assim, o nó intermediário gera um par de chaves pública/privada para toda requisição de rota encaminhada pelo nó:

$$\{ \text{RREQ}, seqnum, pk_{one}, tr_{dest}, TBO \}$$

Quando o nó de destino inicializa a resposta de rota, uma chave simétrica K_{seed} é gerada para proteger a prova de abertura do *trapdoor* e o *onion* TBO. Incluíndo a metodologia de *trapdoor* mencionada anteriormente, os pacotes de requisição e resposta de rota (já protegendo a chave simétrica K_{seed} com a chave pública) são:

$$\{ \text{RREQ}, seqnum, pk_{one}, K_T(dest, K_C), K_C(dest), TBO \}$$

$$\{ \text{RREP}, \{K_{seed}\}pk_{one}, K_{seed}\{K'_C, \text{TBO}\} \}$$

Posteriormente nos nós intermediários, durante a resposta, a prova de abertura do *trapdoor* pelo destino é realizada verificando-se a igualdade: $K_C(dest) = K'_C(dest)$.

Durante os pacotes de RREP e RREQ, e no encaminhamento de dados, são utilizados ACK's (*acknowledgments*) anônimos para confirmar certas operações (como abertura do *trapdoor*) e cancelar assim o *re-broadcast* de pacotes. O anonimato é alcançado utilizando como referência o pseudônimo do nó que se deseja informar o ACK.

4.3 Simulação do Protocolo

Para simular informalmente o protocolo selecionado, a seguinte metodologia foi adotada:

- ✓ Estudar os detalhes de funcionamento do protocolo selecionado;
- ✓ Adaptar o protocolo conforme as necessidades de verificação das propriedades de anonimato;
- ✓ Compilar o simulador com as adaptações;
- ✓ Criar o cenário de simulação e analisar os resultados;

De acordo com a metodologia adotada, foi possível criar e simular de forma rápida e eficiente o protocolo selecionado. Como a implementação do protocolo está bem completa dentro do simulador, nenhuma modificação essencial foi necessária em termos de funcionalidade. Foi necessário, entretanto, configurar o ambiente de simulação para ser possível adaptar o protocolo e capturar informações pertinentes para verificação de suas propriedades.

As modificações no código-fonte do protocolo não interferiram na forma que o protocolo opera. Foram incluídas linhas de código na camada de roteamento para ser possível verificar a estrutura dos pacotes durante o período de simulação, em diversas etapas distintas (RREQ e RREP, por exemplo). As principais modificações foram incluídas para capturar os pacotes e imprimir as saídas no simulador. Abaixo um exemplo de como essas alterações foram incluídas:

```
    if (*pktPtr == ANODR_RREQ)
    {
        // Imprime o conteudo do pacote de requisicao
        AnodrRreqPacket* rreqPkt = (AnodrRreqPacket *) pktPtr;

            printf("\n ======= PACOTE RREQ [Simulação: %s segundos] [Nó %u]
[Send/Receive %c] ======= \n",
                                clockStr,
                                node->nodeId,
                                sendOrReceive);
            if (true) {

                  // PKone
                  printf(" PkOne: ");
                  for (int i = 0; i < sizeof(rreqPkt->onetimePubKey); i++) {
                        printf("%02X",(int)rreqPkt->onetimePubKey[i]);
```

```
                    }
                    printf("\n");

                    // Kt(dest,Kc)
                    printf(" Kt(dest,Kc): ");
                    for (int i = 0; i < sizeof(rreqPkt-
>globalTrapdoor.bits.ciphertext); i++) {
                          printf("%02X",(int)rreqPkt-
>globalTrapdoor.bits.ciphertext[i]);
                    }
                    printf("\n");

                    // Kc(dest)
                    printf(" Kc(dest): ");
                    for (int i = 0; i < sizeof(rreqPkt-
>globalTrapdoor.commitment); i++) {
                          printf("%02X",(int)rreqPkt-
>globalTrapdoor.commitment[i]);
                    }
                    printf("\n");

                    // onion
                    printf(" TBO: ");
                    for (int i = 0; i < sizeof(rreqPkt->onion.bits); i++) {
                          printf("%02X",(int)rreqPkt->onion.bits[i]);
                    }
                    printf("\n");
              }


              printf(" ======= PACOTE RREQ ======= \n");

                                        }
```

A seção 4.3.1 detalha a configuração do ambiente. Em seguida, na seção 4.3.2, o cenário de simulação é apresentado, para então, na seção 4.3.3, ser possível analisar e comparar os resultados. Os resultados são mostrados de forma iterativa, apresentando-se as adaptações do protocolo, a referência com o cenário e a verificação da propriedade de anonimato.

4.3.1 Configuração do ambiente

A plataforma utilizada para rodar o simulador foi um notebook equipado com processodor *Intel Core 2 Duo* de 32 bits e sistema operacional *Microsoft Windows XP Sp3*. É possível compilar o simulador em outras plataformas, sendo necessário baixar os códigos fontes específicos.

A versão utilizada do simulador inclui, em parte, os códigos-fonte na linguagem C++

para alteração e posterior recompilação. Na realidade, o instalador vem com a biblioteca de binários pré-compilada, o que é suficiente para a grande maioria dos usuários. Quando alterações são necessárias, o ambiente *Microsfot Visual C++* é indicado pelo fabricante.

O *Microsoft Visual C++* foi utilizado durante todo o processo de simulação do protocolo. A cópia foi disponibilizada de forma gratuita na versão *2005 Express*. De forma geral, a instalação e configuração do ambiente é realmente muito simples e não requer nenhum esforço adicional. Todos os procedimentos foram seguidos conforme a documentação detalhada do fabricante [QUALNET, 2008]. A figura 5 ilustra o ambiente de desenvolvimento utilizando o Visual C++.

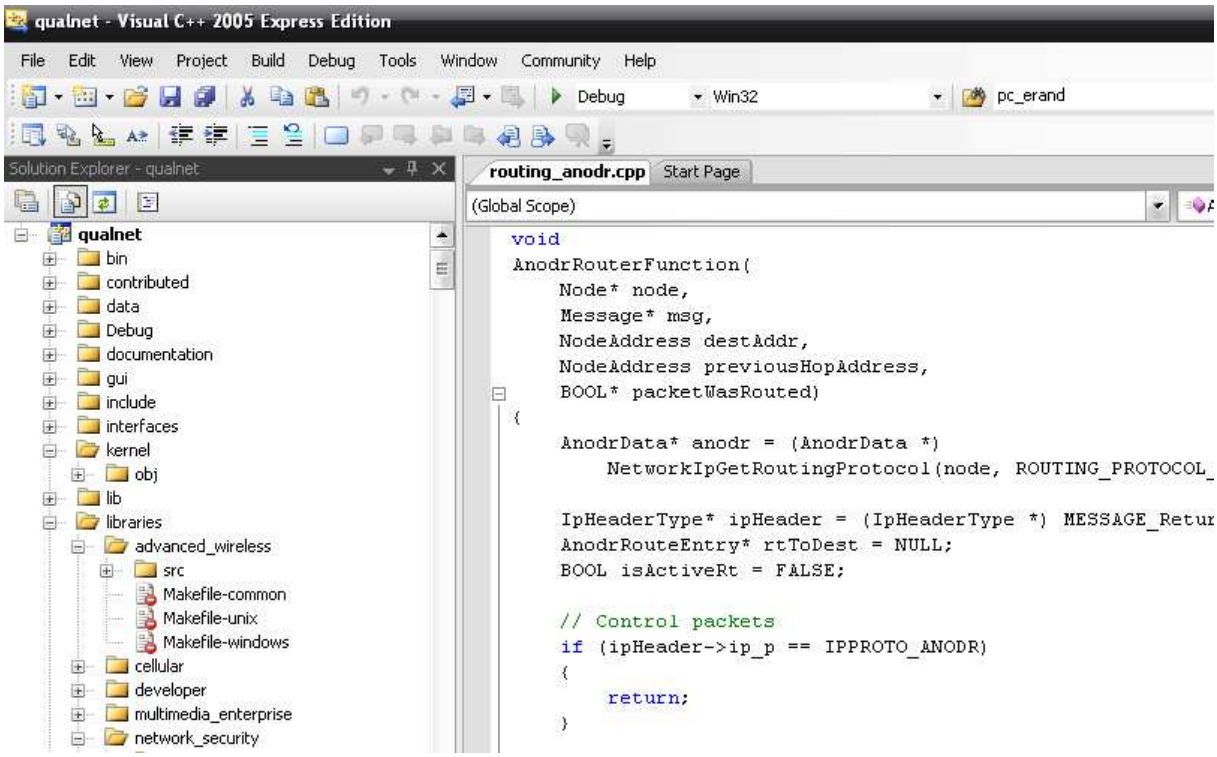

Figura 8: Ambiente de desenvolvimento Visual C++

4.3.2 Simulação do cenário

Para simular o protocolo uma rede ad-hoc móvel foi criada dentro de um campo de 600 metros x 600 metros, onde 5 nós foram dispostos uniformemente. Todos os nós são considerados simétricos, ou seja, se um determinado nó A consegue se comunicar diretamente com B, então B consegue se comunicar diretamente com A. O resumo das configurações do

cenário pode ser visto na tabela 4. A figura 9 ilustra o cenário utilizado durante a simulação.

Tabela 4: Detalhes de configuração do cenário de simulação

Parâmetro	Valor
Tempo de simulação	1 minuto
Dimensões do cenário	600 metros X 600 metros
Número de nós	5
Alcance dos nós	250 metros
Velocidade do canal	2 Mbits/seg

A estratégia para geração do tráfego de dados da simulação foi montada de acordo com as verificações desejadas. Para simular os dados foi utilizado o gerador padrão de bytes (CBR – *Constant Bit Rate*). As ligações entre os nós foram ajustadas diamicamente de acordo com a necessidade de validação de determinado parâmetro. Em termos gerais, o modelo se assemelha ao mostrado pela figura 10.

Na próxima seção os resultados de cada simulação são analisados de acordo com o parâmetro de anonimato. Dessa forma, os passos são executados da seguinte maneira: (1) verificação do parâmetro de anonimato; (2) alteração da geração de tráfego entre os nós (caso necessário); (3) adaptação do protocolo (código-fonte) para geração de dados relevantes durante a simulação (caso necessário); e (4) análise e interpretação da saída do simulador.

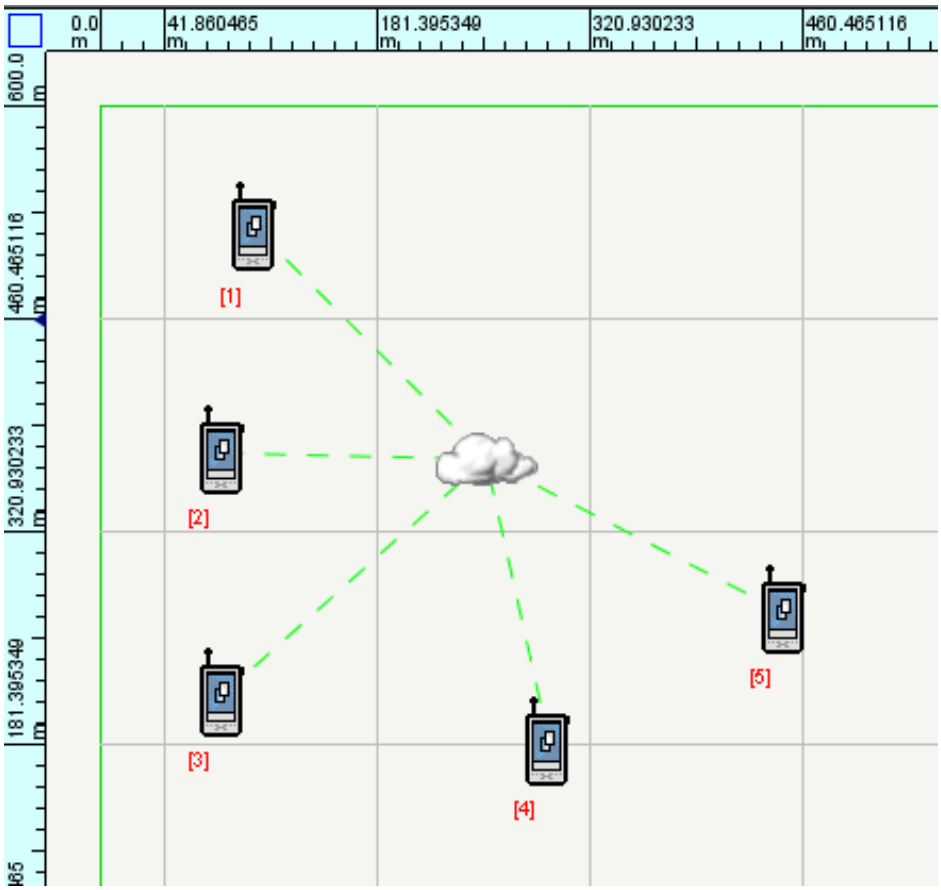

Figura 9: Cenário de simulação do protocolo

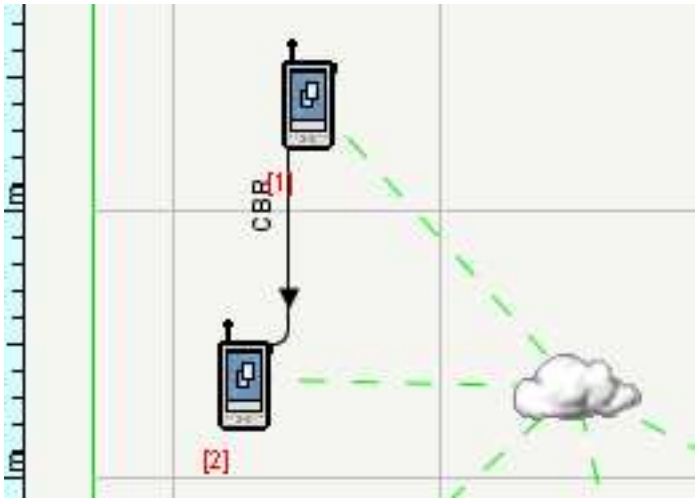

Figura 10: Gerador de tráfego CBR entre o nó 1 e 2

4.3.3 Resultados

As propriedades adotadas para verificação de anonimato são baseadas no estudo comparativo realizado por TAMASHIRO (2007). São elas:

- ✓ **Anonimato da identidade do nó origem (destino)**: um atacante não pode relacionar um pacote à identidade do nó origem (destino);

- ✓ **Anonimato da identidade dos nós intermediários**: um atacante não pode relacionar um pacote às identidades dos nós emissor e destinatário;

- ✓ **Anonimato do venue do nó origem (destino)**: um atacante não pode correlacionar o nó origem (destino) e seu *venue*;

- ✓ **Privacidade de localização ou privacidade do venue**: um atacante não pode relacionar a identidade de um nó à sua localização exata ou aproximada - *venue* (1), nem correlacionar pacotes transmitidos por um nó em sua localização atual (2). Se um protocolo garante (1), mas não (2), há privacidade fraca. Se ambas as propriedades são válidas, há privacidade forte;

- ✓ **Privacidade do padrão de movimento**: um atacante não pode relacionar a identidade de um nó aos pacotes que encaminha ou recebe (1), nem correlacionar pacotes transmitidos por um nó ou grupo de nós em suas localizações atuais e anteriores (2). Se um protocolo garante (1), mas não (2), há privacidade fraca. Se ambas as propriedades são válidas, há privacidade forte;

- ✓ **Anonimato da rota**: um atacante não pode correlacionar os nós pertencentes à rota (origem, intermediários e destino) e as transmissões, ao longo da rota, referentes ao mesmo pacote.

4.3.3.1 Anonimato de Identidade

Ao verificar as propriedades de anonimato de identidade (TAMASHIRO, 2007), o protocolo ANODR apresentou as seguintes características:

- ✓ **Anonimato no nó de origem**: Sim
- ✓ **Anonimato no nó de destino**: Não
- ✓ **Anonimato nos nós intermediários**: Sim

Para verificar essas características do protocolo, uma simulação de transmissão de dados foi configurada entre o nó [1] e o nó [4]. Pela disposição geográfica dos mesmos, os pacotes de requisição e respostas precisam passar pelos nós intermediários [2] e [3] para alcançarem o destino. A figura 11 ilustra a configuração dos nós e o envio dos pacotes durante a simulação.

Segundo TAMASHIRO (2007), o anonimato da identidade de origem é garantido, pois durante a requisição e resposta de rota a identidade da origem [1] está cifrada nas camadas *onion.* Além disso, a chave é conhecida somente pelo nó de origem. Essa propriedade é verificada ao analisar os pacotes de requisição e resposta que são recebidos da camada física (*Wireless MAC*) para a camada de roteamento. Algumas linhas foram adicionadas ao código-fonte do protocolo para imprimir os detalhes do pacote de requisição e resposta (Figura 12).

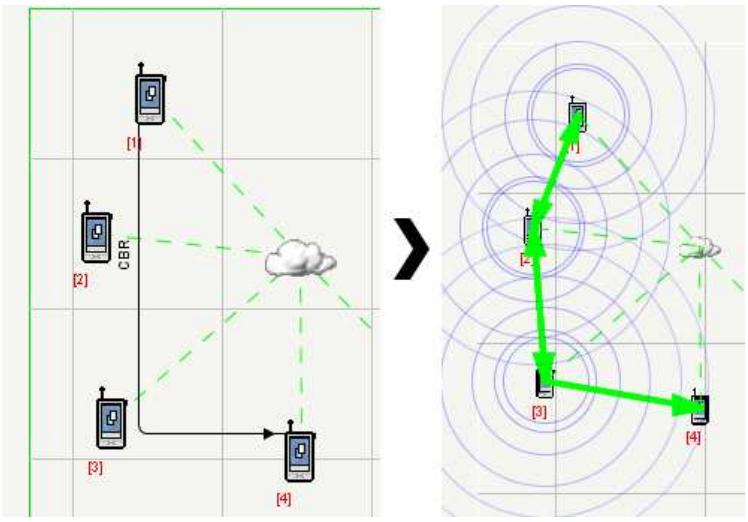

Figura 11: Envio de pacotes entre o nó [1] e [4]

```
printf("\n ======= PACOTE RREQ [Simulação: %s segundos] [Nó %u] [Send/Receive %c] ======= \n",
            clockStr,
            node->nodeId,
            sendOrReceive);
// PKone
char pkone[MAX_STRING_LENGTH];
DebugPrint128BitInHex(pkone, rreqPkt->onetimePubKey);
printf(" PKone: %s \n", pkone);

// Kt(dest,Kc)
char trapdoor[MAX_STRING_LENGTH];
DebugPrint128BitInHex(trapdoor, rreqPkt->globalTrapdoor.bits.ciphertext);
printf(" Kt(dest,Kc): %s \n", trapdoor);

// Kc(dest)
char proof[MAX_STRING_LENGTH];
DebugPrint128BitInHex(proof, rreqPkt->globalTrapdoor.commitment);
printf(" Kc(dest): %s \n", proof);

// onion
char onion[MAX_STRING_LENGTH];
DebugPrint128BitInHex(onion, rreqPkt->onion.bits);
printf(" TBO: %s \n", onion);


printf(" ======= PACOTE RREQ ======= \n");
```

Figura 12: Detalhes do código para gerar a saída do pacote de requisição

Um nó intermediário ao receber uma requisição de rota não tem como verificar a origem do pacote. A captura dos pacotes RREQ e RREP no nó intermediário [2] pode ser observada na tabela 5.

Tabela 5: Captura dos pacotes RREQ e RREP

```
======= PACOTE RREQ [Simulação: 1.000000000 segundos] [Nó 1] [Send/Receive S]

 PkOne:
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000

 Kt(dest,Kc):
040000C06172652074686520646573741462FC6C8D2F9520EE8A02232A9673290E2CA11FC59A8C62B07
B9156DAC6846800000000000000028000000B61C0000000002018047120054 9DD2D434FF120020E990
```

```
7CE001917CFFFFFFFFDB01917C26D1750000000201000000001C00000048500E0100000000

 Kc(dest): 596F7520617265207468652064657374

 TBO: 5562C6711AB9EC164DA52964779A574E

 ======= PACOTE RREQ =======



 ======= PACOTE RREQ [Simulação: 4.451334460 segundos] [Nó 2] [Send/Receive R]

 PkOne:
00000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000

 Kt(dest,Kc):
040000C06172652074686520646573741462FC6C8D2F9520EE8A02232A9673290E2CA11FC59A8C62B07
B9156DAC68468000000000000000028000000B61C00000000020180471200549DD2D434FF120020E990
7CE001917CFFFFFFFFDB01917C26D1750000000201000000001C00000048500E0100000000

 Kc(dest): 596F7520617265207468652064657374

 TBO: 5562C6711AB9EC164DA52964779A574E

 ======= PACOTE RREQ =======



 [RREQ - Processamento interno] [Nó 2] [Time 4.451374460 seg.]

 ONION de entrada: 5562C6711AB9EC164DA52964779A574E

 Chave do ONION: AE2F145EA4300C66902C9B7E8E74E514

 ONION de saída: FB4DD22FBE89E070DD89B21AF9EEB25A



 ======= PACOTE RREQ [Simulação: 4.451374460 segundos] [Nó 2] [Send/Receive S]

 PkOne:
00000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000

 Kt(dest,Kc):
040000C06172652074686520646573741462FC6C8D2F9520EE8A02232A9673290E2CA11FC59A8C62B07
B9156DAC68468000000000000000028000000B61C00000000020180471200549DD2D434FF120020E990
7CE001917CFFFFFFFFDB01917C26D1750000000201000000001C00000048500E0100000000

 Kc(dest): 596F7520617265207468652064657374

 TBO: FB4DD22FBE89E070DD89B21AF9EEB25A

 ======= PACOTE RREQ =======



 ======= PACOTE RREP [Simulação: 11.354123638 segundos] [Nó 4] [Send/Receive S]
```

```
 {Kseed}PKone:
040000C06172652074686520646573741462FC6C8D2F9520EE8A02232A9673290E2CA11FC59A8C62B07
B9156DAC6846800000000000000028000000B61C0000000002018047120054 9DD2D434FF120020E990
7CE001917C

 Kseed{PRdest, TBO}:
1462FC6C8D2F9520EE8A02232A9673294C0C88151597BD3FC6EF6074EF3CE805

 ======= PACOTE RREP [FIM]    =======


 [RREP - Processamento interno] [Nó 3] [Time 11.395042235 seg.]

 Kseed: 00000000AD91BE31173D7728E3F9B620

 ONION: 4C0C88151597BD3FC6EF6074EF3CE805

 K'c: 1462FC6C8D2F9520EE8A02232A967329
```

Na primeira captura, o *onion* TBO está cifrado com a chave simétrica do nó [1] e não possui nenhuma informação do nó de destino. Como o *onion* do pacote RREP é o mesmo da requisição, o anonimato é garantido da mesma forma durante a resposta. Analisando informações do processamento interno dos (considerado assim um nó malicios), nenhuma informação permite associar o pacote à sua origem.

Em referência ao anonimato da identidade do nó de destino, TAMASHIRO (2007) indica que nós intermediários podem conhecer o nó de destino, que está presente na confirmação de abertura do *trapdoor* durante o processamento da resposta do pacote de rota (na verificação de igualdade $K_C(dest) = K'_C(dest)$). Para verificar essa propriedade a função que trata o processamento da resposta foi analisada (Figura 13).

```
//--------------------------------------------------------------------
// FUNCTION: AnodrHandleReply
// PURPOSE:  Processing procedure when RREP is received
// ARGUMENTS: node, the node received reply
//            msg,  Message containing rrep packet
//            interfaceIndex, the interface through which reply has been received
// RETURN:   TRUE: my RREP;  FALSE: none of my business
//--------------------------------------------------------------------

static BOOL
AnodrHandleReply(Node* node,
                 Message* msg,
                 int interfaceIndex)
{
```

Figura 13: Função no protocolo que trata do processamento de RREP

Ao verificar a estrutura interna do código responsável por fazer a verificação da prova de abertura de *trapdoor*, nenhuma informação do destino fica disponível para leitura no nó intermediário. Os seguintes passos são executados para verificação da prova de abertura do *trapdoor*:

- ✓ A tag *dest* (que aqui não representa o ID do nó ou o endereço IP) é cifrada utilizando a chave simétrica K'_C;
- ✓ O resultado da operação anterior é verificado com a entrada K_C(*dest*) da tabela de rota do nó;
- ✓ Se não for igual o pacote é descartado, pois um possível ataque de injeção de RREP foi executado;

Na simulação do protocolo a tag *dest* é uma constante definida no cabeçalho da aplicação: `#define ANODR_DEST_TAG ((byte *)"You are the dest").` O que realmente indica quem deve receber a mensagem é a chave previamente compartilhada entre a origem e o destino. No momento em que o destino recebe a mensagem, a chave compartilhada permite abrir o conteúdo da tag *dest* cifrada e verificar com a constante definida no protocolo. Portanto, o protocolo **consegue manter o anonimato** do destino nos nós intermediários.

Em relação ao anonimato de identidade dos nós intermediários, TAMASHIRO (2007) afirma que na fase de requisição e resposta são utilizadas chaves públicas temporárias e ainda assim são estabelecidos pseudônimos nas rotas entre os nós. Durante a captura dos pacotes (tabela 5) de RREQ e RREP verifica-se que os nós intermediários da rota estão cifrados dentro da estrutra *onion*. Mesmo assim, são usados *nonce*'s (pseudônimos) para identificar cada nó em uma determinada sessão. Verificando a estrutura de rotas do nó intemediário [2] durante uma transmissão de dados, é possível notar que a associação entre nós é realizada através dos pseudônimos:

```
[DATA] Nó [2] encaminha um DATA packet
 (00000000.4b2920f4.73ae53ad.16735255. ==> 00000000.6edd0c61.353d2248.63ac4158.)
```

4.3.3.2 Anonimato de Venue na Origem e Destino

O conceito de *venue* refere-se ao menor lugar espacial onde um adversário pode detectar uma transmissão de rede sem-fio. Assim, o anonimato de *venue* deve evitar que um atacante ou um nó intermediário comprometido saiba sua localização. Não deverá ser possível reconhecer o número de saltos percorridos, ou ainda saber se o tamanho do pacote indica uma localização aproximada. Segundo essa definição, o protocolo ANODR é avaliado da seguinte forma (TAMASHIRO, 2007):

- ✓ **Anonimato de *venue* no nó de origem:** Não provê. Um atacante interno ao receber uma mensagem de requisição RREQ sabe que é vizinho da origem pelo tamanho do *onion* recebido. Um atacante externo não consegue identificar o tamanho, pois são adicionados *paddings* para parecer que o pacote possui o mesmo tamanho;

- ✓ **Anonimato de *venue* no nó de destino:** Provê. Nos pacotes de requsição de rota (RREQ) o tamanho muda a cada nó intermediário, mas não é possível inferir quando o nó chega ao destino. Da mesma forma com os pacotes de resposta, que diminuem de forma padronizada de tamanho a cada salto, não é possível definir o *venue* do destino, pois o tamanho inicial da resposta depende da distância da origem.

Para verificar o anonimato de *venue* no nó de origem, foi utilizada a mesma configuração disposta na figura 11. O nó de origem [1] deseja se comunicar com o nó de destino [4]. Os nós [2] e [3] são os nós intermediários. Primeiro analisou-se os tamanhos dos pacotes sob o ponto de vista de um atacante externo. Observando-se novamente a tabela 5, os tamanhos dos onions permanecem os mesmos a cada salto, sendo impossível de determinar a localização aproximada da origem. Ao analisar o tamanho do onion na estrutura interna do protocolo, observou-se o seguinte:

Tabela 6: Tamanho da camada *onion* durante o processamento interno do pacote

```
[RREQ - Processamento interno] [Nó 3] [Time 7.902749041 seg.]
```

```
 ONION de entrada: FB4DD22FBE89E070DD89B21AF9EEB25A
 Chave do ONION: B7415A3AAB1E5D4F1B66D26E16D25A5F
 ONION de saída: 4C0C88151597BD3FC6EF6074EF3CE805


[RREQ - Processamento interno] [Nó 2] [Time 4.451374460 seg.]
 ONION de entrada: 5562C6711AB9EC164DA52964779A574E
 Chave do ONION: AE2F145EA4300C66902C9B7E8E74E514
 ONION de saída: FB4DD22FBE89E070DD89B21AF9EEB25A


[RREP - Processamento interno] [Nó 3] [Time 11.479423399 seg.]
 Kseed: 00000000F420294BAD53AE7355527316
 ONION: 5562C6711AB9EC164DA52964779A574E
 K'c: 00000000000000000000000000000000

[RREP - Processamento interno] [Nó 2] [Time 11.437322817 seg.]
 Kseed: 00000000610CDD6E48223D355841AC63
 ONION: FB4DD22FBE89E070DD89B21AF9EEB25A
 K'c: 00000000000000000000000000000000
```

Analisando a estrutura interna do protocolo é possível perceber que o *onion* não se modifica de tamanho conforme os saltos, permanecendo constante em 16 bytes (a saída do simulador utiliza valores hexadecimais – para cada par de caractereres temos a representação de 1 byte). Assim que um nó intermediário encaminha um pacote de requisição, o pseudônimo encriptado do *onion* de entrada é armazenado na tabela interna de roteamento. O *onion* de saída é associado com o *onion* de entrada para se utilizado no pacote de resposta ($N < > N^i$). Assim, o tamanho da estrutura das camadas do *onion* não é alterado conforme os saltos aumentam, pois cada nó, para uma determinada rota, possui uma relação 1 para 1 com o nó vizinho. Um adversário que realiza um ataque de volume do pacote (analisando se os tamanhos dos campos se alteram de forma padronizada) não consegue determinar, mesmo se tatando de um atacante interno, o *venue* de origem através de uma comparação de tamanho do primeiro *onion* recebido com os demais *onions* da rota. Um ataque de conteúdo (quando algum campo permanece inalterado), por outro lado, consegue determinar quem enviou o

pacote pela primeira vez, no caso do atacante possuir uma visão global da rede (quando o atacante conhece todos os nós da rota). Isso é verificado ao se observar a tabela 5, onde o *seqnum*, que nessa implementação é denotado pelo *trapdoor*, permanece inalterado. O pacote de resposta não é sensível ao ataque de volume, pois nenhum campo se altera de tamanho de forma padronizada, e de conteúdo, pois o *seqnum* nesse caso é protegido pela chave temporária *Kseed*.

4.3.3.3 Privacidade de Localização e Padrão de Movimento

A privacidade de localização em redes sem fio ad-hoc móveis é uma preocupação crítica nesse tipo de rede (GERLA, 2006). Para isso, um atacante não deve possuir informações relevantes que possam identificar a origem ou o destino, e ainda os pacotes de transmissões distintas (em um mesmo nó) não poderão ser correlacionados. Da mesma maneira, o padrão de movimento de um nó deve ser protegido por uma possível análise do histórico de transmissões de um nó. Se o nó move-se em determinada direção, um atacante não deve conhecer esse padrão de movimento, a menos que consiga correlacionar pacotes transmitidos da posição atual com pacotes transmitidos a partir de posições anteriores.

O protocolo ANODR, segundo TAMASHIRO (2007), foi classificado da seguinte maneira:

- ✓ **Privacidade de localização:** Forte;
- ✓ **Padrão de movimento:** Forte.

A identificação da origem e do destino é protegida pelo anonimato de identidade do protocolo. Para verificar a correlação de pacotes foram realizadas duas transmissões distintas no cenário da figura 14. Nesse caso foi adicionado um gerador de tráfego CBR entre o nó [5] até o nó [2].

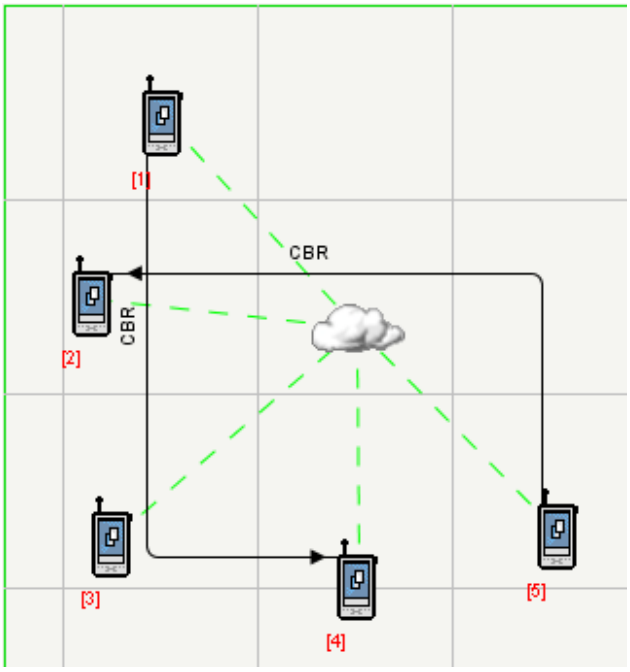

Figura 14: Cenário com 2 geradores de tráfego

O resultado da simulação demonstrou que os *trapdoors* e *onions* não podem ser correlacionados entre transmissões de nós distintos, mesmo quando uma segunda sessão foi iniciada. Isso ocorre porque os *onions* são números aleatórios usados uma única vez (*nonce*`s) e o *trapdoor* é cifrado com a chave simétrica K_C (que é um *nonce*). Abaixo (tabela 7), os resultados das sessões:

Tabela 7: Comparação do *trapdoor* e *onion* entre duas sessões diferentes

```
 ======= PACOTE RREQ [Simulação: 1.000000000 segundos] [Nó 1] [Send/Receive S]

Kc(dest): 596F752061726520746865206465737374
TBO: 5562C6711AB9EC164DA52964779A574E

======= PACOTE RREQ [Simulação: 1.000000000 segundos] [Nó 1] [Send/Receive S]

Kc(dest): 6865726520796F752061726500000000
TBO: 1144213412EDDA12543223454544E77

 ======= PACOTE RREQ =======
```

4.3.3.4 Anonimato de Rota

O anonimato de rota é verificado através da análise de vulnerabilidade dos ataques de reconhecimento de fluxo, que diz respeito à capacidade que o atacante tem de identificar pacotes referentes à mesma rota. TAMSHIRO (2007) avalia o protocolo ANODR da seguinte maneira:

- ✓ **Anonimato de rota em relação a nós fora da rota:** Não são capazes de associar os nós às rotas, pois há anonimato de identidade e ataques de tempo, conteúdo e volume não são executados;

- ✓ **Anonimato de rota em relação a nós pertencentes à rota:** Como a identidade de destino é revelada, nós intermediários internos (trabalhando em conluio) podem descobrir que pertencem à mesma rota. Ainda assim, a distância relativa é revelada pelo ataque de volume.

A garantia do anonimato de identidade foi verificada nas comparações anteriores. O anonimato em relação aos nós fora da rota é garantido, pois o ataque de volume não pode ser executado (de acordo com as comparações realizadas na seção anterior). O ataque de conteúdo (campo *seqnum*, conforme comparações anteriores) também não compromete o anonimato, pois um atacante que não pertence à rota não consegue determinar o caminho exato do pacote, pois ataques de análise de tempo não podem ser executados. Para verificar o ataque de análise de tempo, foi realizada uma nova simulação. Assim, foi observado que o protocolo adiciona um atraso (*delay*) aleatório antes de fazer o *broadcast* na rede (figura 15).

```
for (i = 0; i < node->numberInterfaces; i++)
{
    IpInterfaceInfoType* intfInfo = ip->interfaceInfo[i];
    if (intfInfo->routingProtocolType == ROUTING_PROTOCOL_ANODR)
    {
        clocktype delay = (clocktype) RANDOM_erand(node->globalSeed) * ANODR_BRO
```

Figura 15: Atraso adicionado no envio de pacotes

O atraso no envio dos pacotes pode ser visualizado na tabela 8. Foram executadas duas simulações com *seeds* globais distintos. Ao remover o código que adiciona o *delay*, o atraso entre envio e recebimento permaneceu constante em 1.33 ms.

Tabela 8: Atraso verificado entre o envio e recebimento de mensagens

```
SIMULAÇÃO 01: (seed = 3535)

======= PACOTE RREQ [Simulação: 1.000000000 segundos] [Nó 1] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.001755132 segundos] [Nó 2] [Send/Receive R]
delay: 1.75 ms

======= PACOTE RREQ [Simulação: 1.001795132 segundos] [Nó 2] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.012804374 segundos] [Nó 3] [Send/Receive R]
delay: 11.01 ms

======= PACOTE RREQ [Simulação: 1.012844374 segundos] [Nó 3] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.021248526 segundos] [Nó 4] [Send/Receive R]
delay: 8.04 ms

======= PACOTE RREQ [Simulação: 1.021288526 segundos] [Nó 4] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.030986021 segundos] [Nó 5] [Send/Receive R]
delay: 9.69 ms


SIMULAÇÃO 02: (seed = 23)

======= PACOTE RREQ [Simulação: 1.000000000 segundos] [Nó 1] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.008634181 segundos] [Nó 2] [Send/Receive R]
delay: 8.6 ms

======= PACOTE RREQ [Simulação: 1.008674181 segundos] [Nó 2] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.014545871 segundos] [Nó 3] [Send/Receive R]
delay: 5.8 ms

======= PACOTE RREQ [Simulação: 1.014585871 segundos] [Nó 3] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.019443794 segundos] [Nó 4] [Send/Receive R]
delay: 4.85 ms

======= PACOTE RREQ [Simulação: 1.019483794 segundos] [Nó 4] [Send/Receive S]
======= PACOTE RREQ [Simulação: 1.030240469 segundos] [Nó 5] [Send/Receive R]
delay: 10.75 ms
```

Em relação aos nós pertencentes à rota, o anonimato é parcialmente garantido. A identidade de destino não é revelada (conforme análise anterior). Entretanto, o protocolo é

suscetível ao ataque de conteúdo, permitindo determinar a localização aproximada do primeiro envio. Em um ataque coordenado, a rota da mensagem pode ser descoberta. Se um nó X, por exemplo, estiver comprometido, um adversário consegue fazer uma ligação entre dois pseudônimos de rota (para cada rota que passa pelo nó). Se *n* nós estão comprometidos - e fazem parte de uma rota consecutiva - então *n* + 1 nós são relacionados. Se os nós não forem consecutivos, o adversário pode criar segmentos, mas será difícil relacionar vários segmentos para formar a rota completa. Por exemplo, se o nó [1] é a origem e o nó [5] o destino (dispostos conforme a figura 9), e os nós [1], [2], [4], [5] estão comprometidos, então o adversário consegue criar dois segmentos ([1],[2],[3] e [3],[4],[5]), porém não consegue associa-los à mesma rota.

5 CONCLUSÃO

Em uma rede ad-hoc móvel dois nós podem comunicar-se diretamente se estiverem nos seus alcances de freqüência. Quando isso não acontece, nós intermediários servem como roteadores das mensagens, e como não existe uma estrutura centralizada (como os tradicionais roteadores), todos nós da topologia devem trabalhar de forma cooperativa para manter o bom funcionamento da rede. Em ambientes hostis, onde falhas de segurança podem comprometer decisivamente a rede, mecanismos tradicionais não conseguem garantir anonimato de identidade, localização e rota.

Em ambientes críticos, uma solução eficiente, que considere fatores internos (capacidade dos nós, consumo de energia, etc.) e externos (mobilidade, ataques passivos, etc.), se torna fundamental para operacionalização da rede e segurança das operações. Da mesma forma, o crescente uso de redes ad-hoc móveis em soluções comerciais e redes públicas também busca prover anonimato. Assim, uma solução ótima nunca será viável, principalmente devido à característica altamente heterogênia das aplicações. É necessário fazer um balanceamento de necessidades, procurando priorizar as características de cada rede e a viablidade comercial na adoção de um protocolo. Levando-se também em consideração o atual nível de pesquisa na área, dificilmente um protocolo será padronizado nos próximos anos e dificilmente, se assim for, apenas um será comercializado em equipamentos.

O presente estudo procurou apresentar um panorama na área de redes sem fio ad-hoc móveis, e ainda verificar e simular os levantamentos e comparativos realizados por TAMASHIRO (2007). Alguns conceitos foram apresentados no sentido de situar redes ad-hoc móveis em um contexto mais genérico. Foi detalhado o funcionamento e estrutura interna do protocolo ANODR, um protocolo do tipo *source-routing* sob demanda, que procura prover anonimato usando uma combinação de pseudônimos, chaves públicas e privadas, redes MIX-net e operações com chaves simétricas e assimétricas.

O resultado da simulação comprovou o anonimato de identidade do protocolo. No estudo comparativo realizado por TAMASHIRO (2007) é dito que a identidade de destino pode ser revelada. O resultado, por outro lado, demonstrou que isso não é possível. Os outros parâmetros de anonimato foram comprovados.

Em relação ao futuro dos protocolos que tentam prover anonimato nas redes sem fio ad-hoc móveis, é um pouco difícil prever o comportamento ideal de tais redes. A própria

concepção dessas redes viabiliza um número grande de aplicações – que se beneficiam principalmente da mobilidade e da facilidade de instalação e configuração. Contudo, novas ameaças à segurança da rede devem ser tratadas de forma criteriosa, e isso depende diretamente do tipo de aplicação. O anonimato é apenas um dos itens de segurança. Ataques ativos e segurança em outras camadas devem ser estudados da mesma forma. O conjunto de soluções de segurança para atender os requisitos vai depender do contexto da aplicação (energia, mobilidade, poder computacional, etc.). Assim, nenhuma solução poderá ser considerada ideal, já que a abrangência de aplicações é potencialmente infinita. Seria interessante, entretanto, incluir questões de segurança nos protocolos mais estudados atualmente (DSR, AODV).

5.1 Trabalhos Futuros

Como forma de verificar os protocolos anônimos apresentados, segue como sugestão a implementação do protocolo em aplicações reais de redes sem fio ad-hoc móveis. Assim, será possível montar uma rede entre dispositivos e simular ataques utilizando adversários físicos. Em um segundo momento, os vários protocolos disponíveis devem ser estudos no sentido de unificar os requisitos de anonimato, para então ser formalizado um protocolo de roteamento anônimo padrão.

REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, Paulo Dias de. **Simulação Computacional para Redes de Computadores**. Rio de Janeiro: Editora Ciência Moderna, 2009.

BANERJEE, Amal; JULIEN, Christine; SHMATIKOV, Vitality. **Certificate Free Anonymous Routing for Mobile Ad Hoc Networks**. 2006. The University of Texas at Austin.

BONEH, D.; FRANKLIN, M. **Identity Based Encryption From Weil Pairing**. 2003. SIAM Journal of Computing: 586-615.

CHLAMTAC, I.; CONTI, M.; LIU, J. J.-N. **Mobile Ad Hoc Networking: Imperatives and Challenges**. Ad Hoc Networks, v. 1, p. 13–64, July 2003.

CORSON, S.; MACKER, J. **Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations**. IETF, January 1999. RFC 2501 (Informational). (Request for Comments, 2501). Disponível em: <http://www.ietf.org/rfc/rfc2501.txt>. Acesso em: Nov. 2008

EL-KHATIB, K. et al. **Secure Dynamic Distributed Algorithm for Ad Hoc Wireless Networks**. In: International Conference on Parallel Processing Workshops. [S.l.: s.n.], 2003. p. 359–366.

GERLA, Mario; SANADIDI, M.Y.; HONG, Xiaoyan; KONG, Jiejun. **Mobility Changes Anonymity: Mobile Ad Hoc Networks Need Efficient Anonymous Routing**. 2006. University of Alabama, University of California.

GOLDSCHLAG, D.; REED, M., SYVERSON, P. **Onion Routing for anonymous and private internet connections**. Communications of the ACM, 42(2):39C4, 1999.

JIMENEZ T.; ALTMAN E. **NS Simulator for Beginners**. Univ. de Los Andes, Venezuela e

Sophie-Antopolis, França. 2003-2004. Disponível em:
< http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>. Acesso em: mar. 2008.

KONG, J.; HONG, X. ANODR: **ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks**. In: 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing. [S.l.: s.n.], 2003. p.291–302.

KUOSMANEM, Petteri. **Classification of Ad Hoc Routing Protocols**. 2007. Artigo publicado. Naval Academy – Helsinki – Finland.

MANET WG. **Mobile Ad-hoc Networks (manet) Charter**. Disponível em:
<http://www.ietf.org/html.charters/manet-charter.html>. Acesso em: out. 2008.

MURTHY, C. S. R.; MANOJ, B. S. Ad **Hoc Wireless Networks. Architectures and Protocols**. [S.l.]: Prentice Hall Professional Technical Reference, 2004.

OTCL. **Object TCL Extension.** Disponível em:
< http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl/> . Acesso em: Mar. 2009

PFITZMANN, A.; HANSEN, M. **Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology**. Feb 2008. Version 0.31. Disponível em: <http://dud.inf.tu-dresden.de/Anon Terminology.shtml>

QUALNET. **Qualnet 4.5.1 Installation Guide**. Jul 2008. Disponível em:
< http://www.scalable-networks.com/publications/documentation/>. Acesso em: Mar. 2009.

SY, D.; CHEN, R.; BAO, L. ODAR: **On-Demand Anonymous Routing in Ad Hoc Networks**. In: Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems. [S.l.: s.n.], 2006. p. 267–276

SNT. **Scalable Networks.** 2009. Disponível em:
< http://www.scalable-networks.com/>. Acesso em: Mar. 2009

TAMASHIRO, Clytia Higa. **Uma Análise de Protocolos de Roteamento Anônimo para Redes Sem Fio Ad Hoc Móveis**. 2007. Tese. (Mestrado Ciências da Computação) UFSC, Florianópolis.

TOR PROJECT, **The Tor Project**. 2006. Disponível em < https://www.torproject.org/ >. Acesso em: Mar. 2009

WIKIPEDIA – TCL. **Tool Comand Language.** Disponível em:
<http://en.wikipedia.org/wiki/TCL> . Acesso em: Mar. 2009

WIKIPEDIA – MESH NETWORKS. **Mesh Networks.** Disponível em:
<http://en.wikipedia.org/wiki/Mesh_network> . Acesso em: Nov. 2008

ZHANG, Y.; LIU, W.; LOU, W. **Anonymous Communications in Mobile Ad Hoc Networks.** In: Annual Joint Conference of the IEEE Computer and Communications Societies. [S.l.: s.n.], 2005. v. 3, p. 1940–1951.

ANEXO I – Cabeçalho do código-fonte do protocolo

```
// Copyright (c) 2001-2005, Scalable Network Technologies, Inc.  All
Rights Reserved.
//                          6701 Center Drive West
//                          Suite 520
//                          Los Angeles, CA 90045
//                          sales@scalable-networks.com
//
// This source code is licensed, not sold, and is subject to a written
// license agreement.  Among other things, no portion of this source
// code may be copied, transmitted, disclosed, displayed, distributed,
// translated, used as the basis for a derivative work, or used, in
// whole or in part, for any program or purpose other than its intended
// use in compliance with the license agreement as part of the QualNet
// software.  This source code and certain of the algorithms contained
// within it are confidential trade secrets of Scalable Network
// Technologies, Inc. and may not be used as the basis for any other
// software, hardware, product or service.

#ifndef ANODR_H
#define ANODR_H

//-----------------------------------------------------
// ASR (Anonymous Secure Routing) protocol is a variant of ANODR.
//
// @inproceedings{ZhuWKBD04,
//    author = {Bo Zhu and Zhiguo Wan and Mohan S. Kankanhalli and Feng
Bao and Robert H. Deng},
//    title = {{Anonymous Secure Routing in Mobile Ad-Hoc Networks}},
//    booktitle = {29th IEEE International Conference on Local Computer
Networks (LCN'04)},
//    pages = {102-108},
//    year = {2004}}
//-----------------------------------------------------
// Search the string "#ifdef ASR_PROTOCOL" in anodr.cpp for differences.
// ASR uses one-time pad, while ANODR uses AES.
// This means ASR uses onion key (one-time only) per RREQ flood,
//    while ANODR uses per-node onion key (AES key) during network
lifetime.
// This is the only protocol-wise difference between ANODR and ASR.
// Thus ASR is a variant of ANODR.
//
// Uncomment the line below (make clean and make) to simulate ASR in
QualNet.
#define ASR_PROTOCOL

#include "crypto.h"

typedef struct struct_network_anodr_str AnodrData;

class D_AnodrPrint : public D_Command
{
    private:
        AnodrData *anodr;
```

```
    public:
        D_AnodrPrint(AnodrData *newAnodr) { anodr = newAnodr; }

        virtual void ExecuteAsString(const char *in, char *out);
};

// # of bits ==> # of Bytes
#define b2B(x)    (((int)x+7)/8)

#define ANODR_INVALID_PSEUDONYM   ((byte
*)"\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0")
#define isInvalid128Bit(x) (!memcmp(x, ANODR_INVALID_PSEUDONYM, 128/8))
#define ANODR_BROADCAST_PSEUDONYM ((byte
*)"\377\377\377\377\377\377\377\377\377\377\377\377\377\377\377\377")
// 128-bit long source tag, can be any 128-bit long well-known message
#define ANODR_SRC_TAG     ((byte *)"I am the source ")
// 128-bit long destination tag, can be any 128-bit long well-known
message
#define ANODR_DEST_TAG    ((byte *)"You are the dest")
#define ANODR_TEST_VECTOR ((byte *)"1234567890ABCDEF")

#define ANONYMOUS_IP    0xfffffffe

// Notation:
//
// {M}_{PK} means using public key PK to encrypt message M.
// K(M)     means using symmetric key K to encrypt message M.

#define AES_BLOCKLENGTH 128
#define ANODR_AES_KEYLENGTH   AES_BLOCKLENGTH
#define ANODR_PSEUDONYM_LENGTH      AES_BLOCKLENGTH
#define ANODR_ECC_KEYLENGTH   192   // 192 for ECC
#define ANODR_ECC_PUBLIC_KEYLENGTH  (3*ANODR_ECC_KEYLENGTH) // Q.x Q.y Q.z
// Note that in GPG's ECC, any ECC ciphertext length is
// "AES ciphertext length + 3*192-bit R", that is,
// the corresponding AES ciphertext length plus extra 576 bits for ECC's
R.
#define ANODR_ECC_BLOCKLENGTH(PLAINTEXTLENGTH)
((PLAINTEXTLENGTH)+3*ANODR_ECC_KEYLENGTH)

// Similar to the 802.11 link layer unicast retransmission count.
// This is for unicast control packets RREP and RERR
#define ANODR_UNICAST_RETRANSMISSION_COUNT     8

// Anodr default timer and constant values ref:
// draft-ietf-manet-anodr-08.txt section: 12

#define ANODR_DEFAULT_ACTIVE_ROUTE_TIMEOUT      (5000 * MILLI_SECOND)
#define ANODR_DEFAULT_NET_DIAMETER              (35)
#define ANODR_DEFAULT_NODE_TRAVERSAL_TIME       (40 * MILLI_SECOND)
#define ANODR_DEFAULT_RREQ_RETRIES              (2)
#define ANODR_DEFAULT_ROUTE_DELETE_CONST        (5)
#define ANODR_DEFAULT_MESSAGE_BUFFER_IN_PKT     (100)
#define ANODR_ACTIVE_ROUTE_TIMEOUT              (anodr-
>activeRouteTimeout)
#define ANODR_DEFAULT_MY_ROUTE_TIMEOUT   (2 * ANODR_ACTIVE_ROUTE_TIMEOUT)
#define ANODR_NET_DIAMETER              (anodr->netDiameter)
#define ANODR_NODE_TRAVERSAL_TIME       (anodr->nodeTraversalTime)
#define ANODR_RREQ_RETRIES              (anodr->rreqRetries)
```

```
#define ANODR_ROUTE_DELETE_CONST        (anodr->rtDeletionConstant)
#define ANODR_MY_ROUTE_TIMEOUT          (anodr->myRouteTimeout)
#define ANODR_NEXT_HOP_WAIT             (ANODR_NODE_TRAVERSAL_TIME + 10)

// This is what is stated in the ANODR spec for Net Traversal Time.
#define ANODR_NET_TRAVERSAL_TIME_NEW   (2 * ANODR_NODE_TRAVERSAL_TIME *
ANODR_NET_DIAMETER)
#define ANODR_NET_TRAVERSAL_TIME   (3 * ANODR_NODE_TRAVERSAL_TIME *
ANODR_NET_DIAMETER / 2)
#define ANODR_FLOOD_RECORD_TIME  (3 * ANODR_NET_TRAVERSAL_TIME)
#define ANODR_DELETE_PERIOD  (ANODR_ROUTE_DELETE_CONST *
ANODR_ACTIVE_ROUTE_TIMEOUT)
#define ANODR_REV_ROUTE_LIFE            (ANODR_NET_TRAVERSAL_TIME)
#define ANODR_BROADCAST_JITTER          (10 * MILLI_SECOND)

// Performance Issue
#define ANODR_MEM_UNIT                    100
#define ANODR_SENT_HASH_TABLE_SIZE       20

#define AES_DEFAULT_DELAY     (40 * MICRO_SECOND)
#define ECC_DEFAULT_DELAY     (40 * MILLI_SECOND)

// Anodr Packet Types
#define ANODR_DUMMY           0   // dummy
#define ANODR_RREQ            1   // route request packet type
#define ANODR_RREQ_SYMKEY     2   // route request packet with global
                                    // trapdoor encrypted in symmetric key
#define ANODR_RREP            3   // route reply packet type
#define ANODR_RERR            4   // route error packet type
#define ANODR_CONTROL_AACK    5   // anonymous ack to unicast packets
#define ANODR_DATA_AACK       6   // anonymous data messages

// ANODR's trapdoor boomerang onion(TBO) is uniformly 128-bit long
#define ANODR_LOCALTRAPDOOR_LENGTH AES_BLOCKLENGTH

// ANODR's route pseudonym: 128-bit
// Also ANODR's per-hop Local Trapdoor:  128-bit TBO - Trapdoored
Boomerang Onion
typedef struct
{
    byte bits[b2B(ANODR_PSEUDONYM_LENGTH)];
} AnodrPseudonym;
#define AnodrOnion      AnodrPseudonym

// ANODR's global trapdoor from source A to destination E is:
//       {DEST_TAG, K_reveal, K_AE}_{PK_E}, K_reveal(DEST_TAG)
// For GPG's ECC encryption:
// 1.2.3. R.x R.y R.z: In 192-bit ECC, each one is 192 bits,
//                     thus 3*192 = 576 bits
// 4. 384-bit plaintext: DEST_TAG, K_reveal, K_AE.
//    The AES ciphertext C is also 384 bits.
// Thus the ECC ciphertext is totally 576 + 384 = 960 bits = 120 bytes.
// Plus 128-bit trapdoor commitment K_reveal(DEST_TAG), 1088 bits = 136
bytes.
#define ANODR_GLOBALTRAPDOOR_LENGTH
(ANODR_ECC_BLOCKLENGTH(3*AES_BLOCKLENGTH)+AES_BLOCKLENGTH)

// ANODR's Global Trapdoor:
// Suppose the source is node A, the destination is node E.
```

```
// "tr_{dest} = {DEST_TAG, K_{reveal},  K_{AE}}_{PK_E},
K_{reveal}(DEST_TAG)"
//              | 128-bit   128-bit    128-bit|            128-bit
//              |  960 bits in ECC ciphertext |
// where K(M) means using symmetric key K to encrypt message M using AES,
// and "," simply means concatenation.
// K_{AE} is the to-be-shared symmetric key between A and E used later.
// K_{reveal} is a nonce key for the purpose of trapdoor commitment.
//
// For new contacts, the global trapdoor is 1088 bits long.
typedef struct
{
    union
    {
      // "DEST_TAG, K_{reveal}, K_{AE}": 3*128 = 384 bits in plaintext.
      byte plaintext[b2B(3*AES_BLOCKLENGTH)];
      // "{DEST_TAG, K_{reveal}, K_{AE}}_{PK_E}" encrypted in
      // destination E's ECC public key PK_E:
      // 384 + 3*192 = 960 bits long in ECC ciphertext.
      byte ciphertext[b2B(ANODR_ECC_BLOCKLENGTH(3*AES_BLOCKLENGTH))];
    } bits;

    // the commitment "K_{reveal}(DEST_TAG)"
    byte commitment[b2B(AES_BLOCKLENGTH)];
} AnodrGlobalTrapdoor;

// Symmetric key based global trapdoor for RREQ floods with established
K_AE
typedef struct
{
    union
    {
      // "DEST_TAG, K_{reveal}": 2*128 = 256 bits in plaintext.
      byte plaintext[b2B(2*AES_BLOCKLENGTH)];
      // "K_AE(DEST_TAG, K_{reveal})" encrypted in the shared
      // destination E's ECC public key PK_E:
      // 384 + 3*192 = 960 bits long in ECC ciphertext.
      byte ciphertext[b2B(2*AES_BLOCKLENGTH)];
    } bits;

    // the commitment "K_{reveal}(DEST_TAG)"
    byte commitment[b2B(AES_BLOCKLENGTH)];
} AnodrGlobalTrapdoorSymKey;

//------------------------
// ANODR packet structures
//------------------------

// ANODR route request message format
// See page 70, page 94 and page 96 of Jiejun Kong's Ph.D. dissertation.
//
// Assuming we are using 192-bit ECC and 128-bit Pseudonym,
// ANODR's RREQ packets (for 1st-time contact) are of a uniform length
// 8 + 1088 + 128 + 3*192 = 1800 bits = 225 bytes
typedef struct
{
    unsigned char  type;  // ANODR_RREQ for request

    // seqNum = globalTrapdoor (encrypted in ECC)
```

```
    AnodrGlobalTrapdoor globalTrapdoor;

    // TBO: the Trapdoored Boomerang Onion
    AnodrOnion  onion;

    // pk_{onetime}, only comprised of Q.x Q.y Q.z in ECC
    byte onetimePubKey[b2B(ANODR_ECC_PUBLIC_KEYLENGTH)];
} AnodrRreqPacket;

// Symmetric key based RREQ packet for RREQ floods with established K_AE
typedef struct
{
    unsigned char  type;  // ANODR_RREQ_SYMKEY for request

    // seqNum = globalTrapdoor (encrypted in ECC)
    AnodrGlobalTrapdoorSymKey globalTrapdoor;

    // TBO: the Trapdoored Boomerang Onion
    AnodrOnion  onion;

    // pk_{onetime}, only comprised of Q.x Q.y Q.z in ECC
    byte onetimePubKey[b2B(ANODR_ECC_PUBLIC_KEYLENGTH)];
} AnodrRreqSymKeyPacket;

// ANODR route reply message format
// See page 75, page 94 and page 96 of Jiejun Kong's Ph.D. dissertation
//
// Assuming we are using 192-bit ECC and 128-bit Pseudonym,
// ECC encryption of the 128-bit pseudonym is of 128+576=704 bits.
// Thus ANODR's RREP packets are of a uniform length
// 8 + 704 + 128 + 128 = 968 bits = 121 bytes
typedef struct
{
    unsigned char type; // ANODR_RREP for reply

    union
    {
      // 128 bits plaintext
      // nym.pseudonym is unencrypted route pseudonym (i.e., in plaintext)
      AnodrPseudonym pseudonym;

      // 128+576 = 704 bits ECC ciphertext
      // this is nym.pseudonym encrypted by pk_{upstream} (using ECC)
      // pk_{upstream} is the one-time public key of the intended receiver
      // of the RREP packet).  Later this pseudonym functions as
      // a per-hop seed key K_{seed}.
      byte
encryptedPseudonym[b2B(ANODR_ECC_BLOCKLENGTH(AES_BLOCKLENGTH)a)];
    } nym;

    // 256-bit (pr_dest = K_reveal, TBO) encrypted by K_{seed} (using AES)
    union
    {
      struct
      {
          byte anonymousProof[b2B(AES_BLOCKLENGTH)];
          AnodrOnion  onion;
      } aesPlaintext;
      byte aesCiphertext[b2B(2*AES_BLOCKLENGTH)];
```

```
    } rrepPayload;

    //AnodrPseudonym test;
} AnodrRrepPacket;

//
// Assuming we are using 128-bit Pseudonym,
// ANODR's RRER, AACK packets are of a uniform length
// 8 + 128 = 129 bits = 17 bytes
//

// ANODR route error message format
// See page 77 of Jiejun Kong's Ph.D. dissertation
typedef struct
{
    unsigned char type;  //  ANODR_RERR for route error
    AnodrPseudonym pseudonym;  // Anonymous VCI
    // AnodrGlobalTrapdoor seqNum;  // This field is no longer needed
} AnodrRerrPacket;

// ANODR anonymous acknowledgement message format
typedef struct
{
    unsigned char type; // ANODR_CONTROL_AACK or ANODR_DATA_AACK
    AnodrPseudonym pseudonym; // Anonymous VCI
} AnodrAackPacket;

//-------------------------------------
// Anodr Routing table structure
//-------------------------------------
// Anodr route entry of an anonymous virtual circuit of a connection
typedef struct str_anodr_route_table_row
{
    // This field is only meaningful at the source node
    NodeAddress   destAddr;

    // Table entries described in Jiejun Kong's Ph.D. dissertation.
    // Like the dissertation, here `input' and `output',
    // `upstream' and `downstream' are for the direction source ->
destination.
    AnodrGlobalTrapdoorSymKey seqNum;    // seqNum = globalTrapdoor

    // onion_{old}, incoming RREQ TBO = outgoing RREP TBO
    AnodrOnion inputOnion;
#ifdef ASR_PROTOCOL
    // 128-bit TBO key.
    // In ANODR, this key is per-node based
    // (each node only has 1 such key to use in the entire network
lifetime)
    // In ASR, this key is per-flood based
    // (each onion/RREQ flood must have a different such key: a one-time
pad)
    AnodrOnion onionKey;
#endif
    // onion_{new}, outgoing RREQ TBO = incoming RREP TBO
    AnodrOnion outputOnion;

    // the commitment K_reveal(DEST_TAG)
    byte commitment[b2B(AES_BLOCKLENGTH)];
```

```
    // the committed K_reveal
    byte committed[b2B(AES_BLOCKLENGTH)];

    // pk_{upstream}, only need to know Q.x Q.y Q.z in ECC
    byte upstreamOnetimePublicKey[b2B(ANODR_ECC_PUBLIC_KEYLENGTH)];
    // sk_{me}, only need to store Q.x Q.y Q.z in ECC
    byte myOnetimePublicKey[b2B(ANODR_ECC_PUBLIC_KEYLENGTH)];

    // Anonymous VCI towards the source
    AnodrPseudonym pseudonymUpstream;
    // Anonymous VCI towards the destination
    AnodrPseudonym pseudonymDownstream;

    // The interface through which inputOnion comes in during RREQ
    int inputInterface;
    // The interface through which outputOnion comes in during RREP
    int outputInterface;

    // The expire moment of the route entry
    clocktype          expireTime;
    // Whether the route is active, i.e., RREP ACKed
    BOOL               activated;
    // If as the destination, this is the corresponding end-to-end
    // symmetric key K_{AE} in the 1st RREQ packet's global trapdoor
    // encrypted by my ECC public key, which can only be decrypted and
    // know by me
    byte e2eKey[b2B(ANODR_AES_KEYLENGTH)];

    // UnRREPed RREQ retransmission count.
    int rreqRetry;
    // UnACKed unicast retransmission count.
    // Note that all unicast control flow follows the direction src <-
dest
    // Note that all unicast data flow follows the direction src -> dest
    int unicastRrepRetry;
    int unicastRerrRetry;
    int unicastDataRetry;

    struct str_anodr_route_table_row* next;
    struct str_anodr_route_table_row* prev;
} AnodrRouteEntry;

typedef struct
{
    AnodrRouteEntry* head;
    int   size;
} AnodrRoutingTable;

//
// Structure to store packets temporarily until one route to the
destination
// of the packet is found or the packets are timed out to find a route
//
typedef struct str_anodr_fifo_buffer
{
    NodeAddress destAddr;       // Destination address of the packet
    AnodrPseudonym pseudonym; // If not null, need to store and query this
    Message *msg;               // The packet to be sent
    struct str_anodr_fifo_buffer *next; // Pointer to the next message.
```

```
} AnodrBufferNode;

// Link list for message buffer
typedef struct
{
    AnodrBufferNode *head;
    int size;                   // buffer size in # of packets
    int numByte;        // buffer size in # of bytes
} AnodrMessageBuffer;

#if 0
// Structure to store information about messages for which RREQ has been
sent
// These information are necessary until a route is found for the
destination

typedef struct str_anodr_sent_node
{
    NodeAddress destAddr;  // Destination for which the RREQ has been sent
    int ttl;               // Last used TTL to find the route
    int times;             // Number of times RREQ has been sent
    struct str_anodr_sent_node* hashNext;
} AnodrRreqSentNode;

// structure for Sent node entries
typedef struct
{
    AnodrRreqSentNode* sentHashTable[ANODR_SENT_HASH_TABLE_SIZE];
    int size;
} AnodrRreqSentTable;
#endif

// Memory Pool
typedef struct str_anodr_mem_poll
{
    AnodrRouteEntry routeEntry ;
    struct str_anodr_mem_poll* next;
} AnodrMemPollEntry;

// Structure to store the statistical informations of Anodr
typedef struct
{
    D_UInt32 numRequestInitiated;
    unsigned int numRequestResent;
    unsigned int numRequestRelayed;

    unsigned int numRequestRecved;
    unsigned int numRequestDuplicate;
    unsigned int numRequestTtlExpired;
    unsigned int numRequestRecvedAsDest;
    unsigned int numRequestRecvedAsDestWithSymKeyGlobalTrapdoor;

    unsigned int numReplyInitiatedAsDest;
    unsigned int numReplyForwarded;
    unsigned int numReplyAcked;
    unsigned int numReplyRecved;
    unsigned int numReplyRecvedAsSource;

    unsigned int numAackRecved;
```

```
    unsigned int numRerrInitiated;
    unsigned int numRerrForwarded;
    unsigned int numRerrAcked;
    unsigned int numRerrRecved;

    unsigned int numDataInitiated;
    unsigned int numDataForwarded;

    unsigned int numDataRecved;
    unsigned int numDataDroppedForNoRoute;
    unsigned int numDataDroppedForOverlimit;
    unsigned int numMaxHopExceed;
    unsigned int numHops;
    unsigned int numRoutes;

    unsigned int numBrokenLinks;

    unsigned int numMaxSeenTable;   // Added to track table's max entries
    unsigned int numLastFoundHits;  // Added to track LastFound matches

} AnodrStats;

// Anodr main structure to storee all necessary informations for Anodr
typedef struct struct_network_anodr_str
{
    // 128-bit TBO key.
    // In ANODR, this key is per-node based
    // (each node only has 1 such key to use in the entire network
lifetime)
    AnodrOnion onionKey;

    // set of user configurable parameters
    int          netDiameter;
    clocktype    nodeTraversalTime;
    clocktype    myRouteTimeout;
    clocktype    activeRouteTimeout;
    int          rreqRetries;
    int          rtDeletionConstant;

    // set of anodr protocol dependent parameters
    AnodrRoutingTable routeTable;
    AnodrMessageBuffer msgBuffer;

    int               bufferSizeInNumPacket;
    int               bufferSizeInByte;

    AnodrStats stats;
    BOOL statsCollected;
    BOOL statsPrinted;
    BOOL processAck;
    BOOL biDirectionalConn;
    int  ttlStart;
    int  ttlIncrement;
    int  ttlMax;
    clocktype lastBroadcastSent;

    // Performance Issue
    AnodrMemPollEntry* freeList;
```

```
    BOOL isExpireTimerSet;
    BOOL isDeleteTimerSet;

    // TRUE: do real crypto in the headers; FALSE: just simulate
    BOOL doCrypto;
      #ifdef DO_ECC_CRYPTO
    // Elliptic Curve Cryptosystem keys
    // [0] E.p
    // [1] E.a
    // [2] E.b
    // [3] E.G.x
    // [4] E.G.y
    // [5] E.G.z
    // [6] E.n
    // [7] Q.x
    // [8] Q.y
    // [9] Q.z
    // [10] d
    // From [0] to [9] is public, [10] is secret
    // I added placeholder [11] for the random nonce k (for future
signing)
    MPI eccKey[12];
#endif // DO_ECC_CRYPTO
    // for destination's pk
    byte destPubKey[b2B(3*ANODR_ECC_KEYLENGTH)];

    // debug purpose
    unsigned int sendCounter, recvCounter;
} AnodrData;

typedef struct
{
    char packetType;    // RREQ or RREP
    int interface;      // incoming interface
    int ttl;            // In ANODR, ttl is not used
} AnodrCryptoOverheadType;

void AnodrInit(
    Node *node,
    AnodrData **anodrPtr,
    const NodeInput *nodeInput,
    int interfaceIndex);

void AnodrFinalize(Node *node);

void AnodrRouterFunction(
    Node *node,
    Message *msg,
    NodeAddress destAddr,
    NodeAddress previousHopAddress,
    BOOL *packetWasRouted);

void
AnodrMacLayerStatusHandler(
    Node *node,
    const Message* msg,
    const NodeAddress nextHopAddress,
    const int incommingInterfaceIndex);
```

```
void AnodrHandleProtocolPacket(
    Node *node,
    Message *msg,
    NodeAddress srcAddr,
    NodeAddress destAddr,
    int ttl,
    int interfaceIndex);

void
AnodrHandleProtocolEvent(
    Node *node,
    Message *msg);

BOOL
IsMyAnodrDataPacket(Node *node, Message* msg);

BOOL
IsMyAnodrUnicastControlFrame(Node *node, Message* msg, MAC_PROTOCOL
protocol);

BOOL
IsMyAnodrForwardDataFrame(Node *node, Message* msg, MAC_PROTOCOL
protocol);

BOOL
IsMyAnodrBroadcastFrame(Node *node, Message* msg, MAC_PROTOCOL protocol);
#endif
```

APÊNDICE

Simulação e Análise do Protocolo de Roteamento Anônimo ANODR para Redes Sem Fio Ad-Hoc Móveis

DANIEL SALVI WUNDERLICH[1]

[1]UFSC – Universidade Federal de Santa Catarina
Departamento de Informática e Estatística

doutorx@inf.ufsc.br

Abstract: *Mobile Ad Hoc Networks (MANET) are structures formed by equipments that communicate with each other without using a central unit for its management. This paper showed some related security topics and informally simulated a previously choosen protocol, based on a comparative study by Tamashiro (2007). To sum up, this research was intended to study the anonymous protocol ANODR [Kong & Hong, 2003] and to verify its anonimity properties by running some simulations over Qualnet Simulator [SNT, 2009]. Then some theorical properties could have been analysed in comparison with Simulator outputs.*

Resumo: *Redes sem fio ad-hoc móveis (MANET – Mobile Ad Hoc Network) são estruturas caracterizadas por equipamentos que se comunicam entre si sem o uso de uma estrutura central responsável pelo seu gerenciamento. Esse estudo procurou levantar alguns aspectos de segurança e simular discretamente um dos protocolos descritos e comparados por Tamashiro (2007). Em termos gerais, a pesquisa consistiu em estudar o funcionamento do protocolo anônimo ANODR [Kong & Hong, 2003] e verificar as propriedades de anonimato utilizando o simulador de redes QualNet [SNT, 2009]. A simulação permitiu assim analisar as propriedades teóricas com as saídas do simulador.*

1. Introdução

Redes sem fio ad-hoc móveis (MANET – *Mobile ad-hoc Networks*) são estruturas caracterizadas por equipamentos que se comunicam entre si sem o uso de uma estrutura central responsável pelo seu gerenciamento. Esses dispositivos possuem uma habilidade de roteamento que possibilita uma rápida reorganização da topologia da rede e uma liberdade de locomoção que independe de estruturas fixas. Essa flexibilidade, portanto, habilita seu uso em diversas aplicações, como operações militares e de resgate.

Com base no estudo comparativo de diversos protocolos apresentado por Tamashiro (2007), foi realizada uma simulação discreta do protocolo ANODR [Kong & Hong, 2003] com intuito de validar as comparações teóricas apresentadas no estudo. O artigo foi estruturado da seguinte maneira: A seção 2 apresenta definições, características e aplicações das redes sem fio ad-hoc móveis. A seção 3 apresenta os

protocolos anônimos estudados. A seção 4 descreve sucintamente o protocolo ANODR. A seção 5 mostra a simulação e os resultados. Por fim, a seção 6 apresenta as conclusões.

2. Redes Sem Fio Ad-Hoc Móveis

Redes sem fio ad hoc móveis são redes formadas por equipamentos móveis que de alguma forma mantém uma comunicação entre si, sem o uso de estruturas fixas (*switches*, *backbones*, etc.) e sem o uso de enlaces com fio. Além da estrutura, a topologia também não é fixa, ou seja, todos os nós da rede são responsáveis por organizar e controlar a rede, resultando em uma topologia dinâmica. Os nós então são responsáveis por descobrir, dinamicamente, com quais podem se comunicar diretamente e por encaminhar pacotes, cujos destinos não estão no raio de alcance de suas origens. Na figura 1 é possível perceber essa idéia. Enquanto que os nós A, B e C podem se comunicar diretamente, a comunicação com D depende do roteamento através de C.

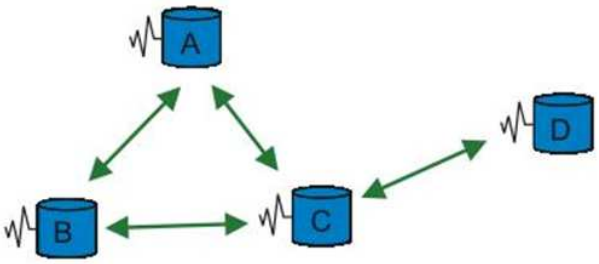

Figura 1. Exemplo de uma rede ad-hoc móvel

2.1 Características

As principais características e aplicações, segundo Corson & Macker (1999), estão relacionadas com a capacidade de mobilidade e dinamismo da topologia. Os nós são equipados com transmissores e receptores variados, o que caracteriza um modelo altamente assimétrico, já que a capacidade de transmissão de um nó pode variar em relação a outro dentro de uma mesma área de alcance. Isso indica que um nó A poderia comunicar-se diretamente com C, mas o nó C, devido ao seu transmissor de curto alcance, necessitaria de um nó intermediário para fazer o roteamento. As principais características apontadas são:

- Autonomia e ausência de infra-estrutura;
- Roteamento de múltiplos saltos;
- Topologia dinâmica;
- Largura de banda reduzida;
- Restrição de energia;
- Limite na segurança física;
- Instalação rápida e baixo custo.

2.2 Aplicações

Inicialmente as aplicações das redes ad-hoc móveis estavam associadas com operações militares e de resgate. Com a popularização das redes sem fio, alguns novos propósitos incluem aplicações de cunho comercial. De certa forma pode-se estender sua utilidade para incontáveis novos propósitos. A tabela 1 exemplifica algumas aplicações.

Aplicações	**Descrição**
Redes Táticas	Comunicação em operações militares Batalhas automatizadas
Redes de Sensores	Monitoramento de residências
Emergência	Operações de busca e resgate
Ambientes comerciais	Comércio eletrônico: serviços como pagamento Veículos: transmissão de notícias, condição das estradas, formação de redes entre veículos próximos
Aplicações educacionais	Configuração de salas virtuais e de videoconferência
Redes Mesh	Zonas residenciais: acesso à Internet Auto-estradas: comunicação para os automóveis

Tabela 1. Aplicações das Redes Sem Fio Ad-Hoc Móveis

3. Protocolos Anônimos

3.1 Anonimato

Para se consolidar redes de roteamento anônimo sem fio é necessário estabelecer e manter rotas que garantam o anonimato. Além disso, é desejável que a localização não seja revelada e que a rota de cada mensagem não seja conhecida pelos nós intermediários. Os protocolos são analisados conforme as seguintes definições [Tamashiro, 2007]:

- **Anonimato da identidade do nó origem e destino:** um adversário não pode ser capaz de associar um pacote à identidade do nó de origem ou destino;
- **Anonimato da identidade dos nós intermediários:** um adversário não pode ser capaz de associar um pacote à identidade dos nós intermediários da rota;
- **Anonimato de *venue* do nó de origem e destino:** um adversário não pode correlacionar o nó de origem ou destino com o seu *venue* (menor área onde uma trasmissão de rede sem fio pode ser detectada);
- **Privacidade de localização ou de *venue*:** um adversário não pode relacionar a identidade de um nó à sua localização exata ou aproximada - venue (1), nem correlacionar pacotes transmitidos por um nó em sua localização atual (2). Se um

protocolo assegura (1), mas não (2), há privacidade fraca. Se ambas as propriedades são válidas, há privacidade forte;

- **Privacidade do padrão de movimento:** um adversário não pode relacionar a identidade de um nó aos pacotes que encaminha ou recebe (1), nem correlacionar pacotes transmitidos por um nó ou grupo de nós em suas localizações atuais e anteriores (2). Se um protocolo assegura (1), mas não (2), há privacidade fraca. Se ambas as propriedades são válidas, há privacidade forte;
- **Anonimato de rota:** um adversário não pode correlacionar os nós pertencentes à rota (origem, intermediários e destino) e as transmissões, ao longo da rota, referentes ao mesmo pacote.

3.2 Tipos de ataque

Segundo Tamashiro (2007), através dos ataques de análise de tráfego um atacante pode inferir informações úteis sobre os nós da rede, localização, topologia, frequência de comunicação e padrões de movimento. São destacados os seguintes ataques:

- **Ataque de análise de tempo:** se os pacotes são processados e encaminhados na mesma ordem em que são recebidos, um atacante pode inferir quais pertencem à mesma rota;
- **Ataque de conteúdo do pacote:** quando o conteúdo do pacote permanece inalterado durante uma transmissão, o atacante pode seguir o pacote;
- **Ataque de volume do pacote:** quando pacotes possuem o mesmo tamanho ou se o tamanho muda de forma padronizada, o atacante pode seguir o pacote;
- **Ataque de reconhecimento de fluxo:** diz respeito à capacidade que um atacante tem de identificar pacotes referentes à mesma rota, através de análise de tempo por exemplo.

3.3 Visão Geral dos Protocolos

Algumas soluções para anonimato em redes sem fio ad-hoc móveis foram desenvolvidas ao longo dos últimos anos, sem que nenhuma tenha conseguido obter completo sucesso. Ainda que muitas dessas soluções sejam parecidas na utilização de mecanismos de encriptação e roteamento, alguns detalhes tornam algumas mais apropriadas para certos dispositivos ou não. Os protocolos estudados são apresentados a seguir.

3.3.1 ANODR

O protocolo ANODR (*Anonymous On-Demand Routing*) foi proposto por Kong & Hong (2003), com objetivo de garantir anonimato de rota e privacidade de localização em um ambiente de roteamento sob demanda.

O anonimato de rota é alcançado por uma abordagem de pseudônimos nos nós com informações globais de *trapdoor*. No processo de descoberta da rota um pacote é enviado por *broadcast* com a solicitação *Route Request*. A rota é formada por um conjunto de nós intermediários, em um esquema *Onion Routing* [Goldschalg *et al*., 1999].

3.3.2 SDAR

O protocolo SDAR, proposto por El-Khatib *et al*. (2004), foi modelado segundo níveis de confiança entre os vizinhos em uma rede sem fio ad-hoc móvel. Esse modelo de confiança atualiza periodicamente cada nó na rota conforme seu comportamento ao longo do tempo. Assim, três níveis foram estabelecidos: baixo, médio e alto.

O nó de origem inicia um processo de estabelecimento da rota após enviar uma mensagem *broadcast* para a rede, com certo nível de confiança. Os nós intermediários que satisfazem esse nível de confiança incluem seu ID e uma *session key* na mensagem, reenviando em seguida para os nós vizinhos. Cada nó intermediário encripta a mensagem antes de adicionar seu ID. Quando o nó destino recebe a mensagem é montada uma mensagem multi-camadas (*Onion Routing*) de retorno com todo o caminho reveso da rota.

3.3.3 MASK

A idéia básica do protocolo MASK [Zhang *et al.,* 2004] é a autenticação anônima entre vizinhos baseada em um sistema dinâmico de pseudônimos (ao invés de seus endereços ou identificadores reais), e no processo anônimo de descoberta de rota e envio de mensagens, baseado na autenticação compartilhada entre vizinhos.

O protocolo procura atender cinco objetivos: anonimato de origem, destino e sua relação; *untraceability* e *unlocability* (adversários não conseguem seguir um pacote); autenticação anônima e segura entre vizinhos; baixo processamento criptográfico; e proteção contra diversos ataques, como reconhecimento de fluxo e análise de tempo. As análises da simulação indicam desempenho razoável e cumprimento dos objetivos de anonimato.

3.3.4 CARP

Banerjee *et al.* (2006) propõe o *Certificate Free Anoymous Routing Protocol* (CARP), que busca prover anonimato de identidade, rota e localização. Um dos propósitos do CARP é cumprir esses requisitos sem causar grandes custos computacionais, limitando assim o uso de criptografia com chaves simétricas e evitando verificações baseadas em autoridades certificadoras.

Para isso, optou por usar outra abordagem de encriptação durante a descoberta da rota: *Identity Based Encryption* (IBE) [Boneh & Fraklin, 2003], um algoritmo assimétrico que utiliza a própria identificação do nó como sua chave publica. Dessa forma, o nó ao enviar a mensagem não precisa compartilhar a chave pública com o destinatário, uma vez que essa é gerada de acordo com os parâmetros do algoritmo IBE (que são compartilhados na inicialização) e o próprio ID do destino.

3.3.5 ODAR

Assim como no protocolo MASK, o protocolo ODAR (*On-Demand Anonymous Routing*), proposto por Sy *et al.* (2006), utiliza *Diffie-Hellman* na geração das chaves secretas na origem e destino. Da mesma forma, o protocolo busca manter anonimato de identidade, rota e localização. O protocolo ODAR propõe o uso de *Bloom Filters*

estruturas para armazenamento de dados onde é possível testar se um dado elemento pertence ao grupo.

4. Detalhes do Protocolo ANODR

O protocolo ANODR foi o primeiro protocolo para roteamento anônimo em redes ad-hoc móveis. Em termos gerais, procura prover anonimato de rota e privacidade de localização, garantindo que os adversários não possam descobrir a identidade real das partes envolvidas em uma transmissão de rede. O planejamento do protocolo é baseado em *broadcast* com informações de *trapdoor*. A idéia de broadcast com *trapdoor* funciona como uma chave para abertura da mensagem somente pelo destino.

O propósito do protocolo é desenvolver um esquema de rotas que não possam ser traçadas, dentro de um ambiente de roteamento sob demanda. Dentro do ambiente que o protocolo é projetado, os atacantes tentam agir passivamente em silêncio, procurando ficar o mais invisível possível. Dessa forma, a contribuição do trabalho é tentar apresentar um protocolo de roteamento que previne um atacante associar os participantes da rede com suas identidades, e impedir que o fluxo de um pacote possa ser seguido no destino ou origem. E embora os adversários possam detectar a existência de transmissões de rede sem fio, fica difícil saber o número de participantes e os padrões de transmissão.

O protocolo é constituído por três fases: descoberta de rota, encaminhamento de dados e manutenção de rota.

4.1 Descoberta de Rota

A descoberta anônima de rota é um processo que ocorre no momento que um determinado nó na rede deseja enviar dados para outro nó. As rotas não são conhecidas previamente, por esse motivo o roteamento é chamado sob demanda. Durante a fase inicial da requisição o seguinte pacote é criado no nó de origem:

$$\{ \text{RREQ}, seqnum, tr_{dest}, onion \}$$

O primeiro campo indica o tipo de pacote que será criado. O campo *seqnum* é um identificador global único do pacote. O terceiro campo é a chave criptográfica de *trapdoor*, que pode ser aberta somente pelo destinatário. O campo *onion* representa as camadas criptográficas (*onion routing*) utilizadas pelos nós intermediários, garantindo dessa forma a rota anônima de pseudônimos. Esse campo é essencial no funcionamento do protocolo, e sua arquitetura influencia decisivamente no desempenho e garantia de anonimato. Para tal, o autor propõe três abordagens, partindo de um esquema simples de misturadores (*Mix-NET* - Banerjee *et al.* (2006)), e finalmente chegando ao modelo final mais eficiente.

O primeiro modelo utiliza uma simples adaptação do esquema da rede de misturadores e chaves assimétricas. É chamado, por isso, de **ANODR-PO** (*Public key Onion*). Durante a fase de requisição (RREQ), cada nó na rota adiciona uma camada referente ao nó anterior e envia por *broadcast* a mensagem. O resultado da mensagem encaminhada é sempre encriptado com a chave pública do nó. Ao chegar no destino (sabe-se pela informação de *trapdoor* do pacote de requisição) a estrutura *onion*

representa uma rota anônima de volta ao nó de origem. A mensagem de resposta possui o seguinte formato:

$$\{ \text{RREP}, N, pr_{dest}, onion \}$$

O primeiro campo identifica um pacote de resposta de rota. O campo *N* é um número randômico único que representa o pseudônimo da rota. O terceiro campo é uma prova de abertura do *trapdoor* pelo destino. O *onion* desse pacote é uma cópia exata do onion que chega ao nó de destino. Durante a resposta de rota o nó tenta decriptar o *onion* utilizando sua chave privada. Quando não encontra sua identificação na mensagem decriptada (primeiro campo), o nó não faz parte da rota e a mensagem é descartada. Ao perceber que a rota deve passar pelo nó, é gerado um *nonce* N^i que substitui o nonce *N* da mensagem. Uma correspondência interna da relação N pra N^i é mantida na tabela interna de encaminhamento. Em seguida, a camada externa do *onion* é retirada e a mensagem é encaminhada.

O esquema de chaves assimétricas do modelo ANODR-PO prejudica o desempenho da rede em termos de latência e *overhead*. Imaginando que as mensagens de RREQ e RREP são enviadas via *broadcast*, toda a rede ficará comprometida em executar operações computacionais custosas para montar e verificar os *onions* das rotas. Assim, é proposto um esquema com chaves simétricas, chamado de **ANODR-BO** (*Boomerang Onion*) – em referência ao uso da mesma chave simétrica usada na requisição e depois na resposta.

Quando um nó intermediário recebe uma requisição para encaminhamento, ele adiciona uma camada ao *Boomerang Onion* e encripta o resultado com uma chave simétrica randômica K_B. Quando a resposta retorna ao nó, a mesma chave é utilizada para remover uma camada do esquema. Assim é garantido que a latência não será grande, pelo menos se depender das operações criptográficas executadas durante a requisição e resposta. O uso de chaves simétricas tem sem mostrado muito eficiente em equipamentos móveis de reduzido poder computacional.

Uma modificação é proposta ao esquema do ANODR-BO para chegar ao modelo final. Como a identidade dos nós intermediários pode ser encontrada nas mensagens, o anonimato não é alcançado entre os vizinhos da rota. É adicionado, então, um *trapdoor* nas camadas. Quando um nó intermediário B recebe a mensagem de requisição de rota, ele gera um *nonce* N_B e adiciona na camada. Da mesma forma do esquema anterior, a mensagem é encriptada usando uma chave simétrica antes de ser enviada via *broadcast* pela rede. A informação de *trapdoor* é a chave simétrica do nó e o *nonce* gerado pelo nó. Durante a resposta de rota o nó será capaz de decriptar a mensagem e confirmar o *nonce* com seu registro interno. A figura 2 ilustra esse processo, chamado de **ANODR-TBO** (*Trapdoor Boomerang Onion*).

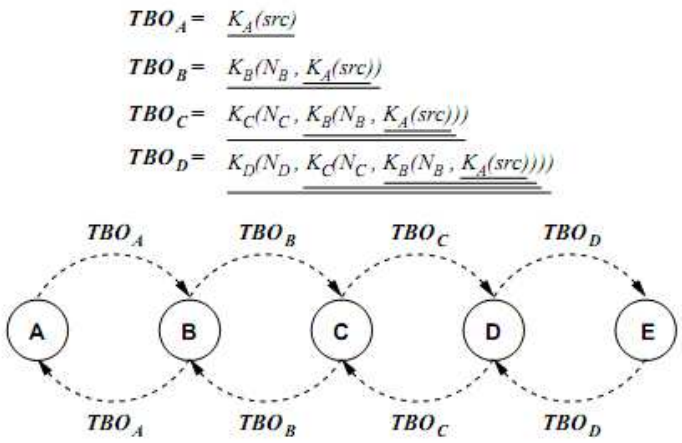

Figura 2. Descoberta anônima da rota usando o esquema ANODR-TBO

4.2 Manutenção de Rota

A manutenção de rota no protocolo ANODR segue um esquema de atualização das tabelas de roteamento. Após certo número de tentativas de retransmissões de dados, o nó verifica o pseudônimo N' que está associado com o pseudônimo N do provável nó defeituoso (desligado ou com defeito). Assim, uma mensagem de erro é enviada no seguinte formato: {REER, N'}. Os nós que recebem via broadcast essa mensagem e verificam que estão usando essa rota, realizam o mesmo procedimento em cascata para notificar os vizinhos.

4.3 Encaminhamento de Dados

Após receber a resposta de rota, o nó de origem encripta os dados utilizando o pseudônimo de rota da sua tabela de encaminhamento. O pacote é então enviado via *broadcast* para a rede e, para cada nó intemediário, verifica-se a tabela de encaminhamento para encontrar uma correpondência com o pseudônimo. Caso pertença à rota, o nó altera o pacote e inclui o seu pseudônimo. Esse procedimento é repetido até o nó chegar ao destino.

5. Simulação do protocolo

5.1 Ambiente de Simulação

O simulador *QualNet* foi utilizado na versão 4.5.1 [Qualnet, 2009], disponibilizado para testes com uma licensa educacional. Em termos gerais, é um simulador simples de usar que procura avaliar o desempenho de redes wireless, cabeadas e mistas. Ele é dividido em módulos, cada um responsável por agrupar diferentes características de redes. Alguns deles são: Wireless, Multimídia, Celular e Satélite, Segurança e Rede de Sensores.

O protocolo ANODR está disponível dentro da biblioteca de Segurança, sendo possível criar o cenário da rede ad-hoc e então escolher esse protocolo como roteamento de dados. O simulador disponibiliza diversas estatísticas para análise dos pacotes do ANODR, como número de requisições e número de respostas. Essas estatísticas são exibidas na forma de gráficos dentro do simulador. São, entretanto, insuficientes para

completar o estudo comparativo e validar as propriedades de anonimato. Para tanto, certas modificações foram necessárias no código-fonte do protocolo para habilitar um conjunto mais detalhado de saídas. Na realidade, as estatísticas já disponibilizadas pelo simulador são mais úteis para análise de performance. Para validar as propriedades de anonimato o código-fonte foi alterado durante o trabalho, possibilitando assim visualizar os campos dos pacotes transitando na rede. Todas alterações do código foram realizadas utilizando o compilador Microsoft Visual C++.

5.2 Cenário de Simulação

Para simular o protocolo uma rede ad-hoc móvel foi criada dentro de um campo de 600 metros x 600 metros, onde 5 nós foram dispostos uniformemente. Todos os nós são considerados simétricos, ou seja, se um determinado nó A consegue se comunicar diretamente com B, então B consegue se comunicar diretamente com A. O resumo das configurações do cenário pode ser visto na tabela 2.

5.3 Resultados

A simulação e os resultados foram agrupados conforme as propriedas de anonimato verificadas. Assim, os seguinte itens de anonimato são analisados: anonimato de identidade e *venue*, privacidade de localização e padrão de movimento e anonimato de rota.

Parâmetro	Valor
Tempo de simulação	1 minuto
Dimensões do cenário	600 metros X 600 metros
Número de nós	5
Alcance dos nós	250 metros
Velocidade do canal	2 Mbits/seg

Tabela 2. Detalhes de configuração do cenário

5.3.1 Anonimato de Identidade

Ao verificar as propriedades apresentadas por Tamashiro (2007), o protocolo ANODR apresentou as seguintes características:

- **Anonimato no nó de origem:** Sim
- **Anonimato no nó de destino:** Não
- **Anonimato nos nós intermediários:** Sim.

Nas capturas observadas no simulador, o *onion* TBO está cifrado com a chave simétrica do nó de origem e não possui nenhuma informação do nó de destino. Como o *onion* do pacote RREP é o mesmo da requisição, o anonimato é garantido da mesma forma durante a resposta. Nenhuma informação permite associar o pacote à sua origem.

Em relação ao anonimato de identidade dos nós intermediários, na fase de requisição e resposta são utilizadas chaves públicas temporárias e ainda assim são estabelecidos pseudônimos nas rotas entre os nós. Durante a captura dos pacotes verificou-se que os nós intermediários da rota estão cifrados dentro da estrutra *onion*.

Em referência ao anonimato da identidade do nó de destino, TAMASHIRO (2007) indica que nós intermediários podem conhecer o nó de destino, que está presente na confirmação de abertura do *trapdoor* durante o processamento da resposta do pacote de rota (na verificação de igualdade $K_C(dest) = K'_C(dest)$). Ao verificar a estrutura interna do código responsável por fazer a verificação da prova de abertura de *trapdoor*, nenhuma informação do destino fica disponível para leitura no nó intermediário. Na simulação do protocolo a tag dest é uma constante definida no cabeçalho da aplicação:

*#define ANODR_DEST_TAG ((byte *)"You are the dest").*

O que realmente indica quem deve receber a mensagem é a chave previamente compartilhada entre a origem e o destino. No momento em que o destino recebe a mensagem, a chave compartilhada permite abrir o conteúdo da tag dest cifrada e verificar com a constante definida no protocolo. Portanto, o protocolo consegue manter o anonimato do destino nos nós intermediários.

5.3.2 Anonimato de *Venue* na Origem e Destino

Ao verificar as propriedades apresentadas por Tamashiro (2007), o protocolo ANODR apresentou as seguintes características:

- **Anonimato de *venue* no nó de origem:** Não.
- **Anonimato de *venue* no nó de destino:** Sim.

Com o resultado da simulação foi verificado que o tamanho da estrutura das camadas do *onion* não é alterado conforme os saltos aumentam, pois cada nó, para uma determinada rota, possui uma relação 1 para 1 com o nó vizinho. Um adversário que realiza um ataque de volume do pacote não consegue determinar, mesmo se tatando de um atacante interno, o *venue* de origem através de uma comparação de tamanho do primeiro *onion* recebido com os demais onions da rota. Um ataque de conteúdo, por outro lado, consegue determinar quem enviou o pacote pela primeira vez, no caso do atacante possuir uma visão global da rede (quando o atacante conhece todos os nós da rota). Isso é pois *seqnum*, que nessa implementação é denotado pelo *trapdoor*, permanece inalterado. O pacote de resposta não é sensível ao ataque de volume, pois nenhum campo se altera de tamanho de forma padronizada, e de conteúdo, pois o seqnum nesse caso é protegido pela chave temporária *Kseed*.

5.3.3 Privacidade de Localização e Padrão de Movimento

Ao verificar as propriedades apresentadas por Tamashiro (2007), o protocolo ANODR apresentou as seguintes características:

- **Privacidade de localização:** Forte.
- **Padrão de movimento:** Forte.

O resultado da simulação demonstrou que os *trapdoors* e *onions* não podem ser correlacionados entre transmissões de nós distintos, mesmo quando uma segunda sessão foi iniciada. Isso ocorre porque os *onions* são números aleatórios usados uma única vez (*nonce*`s) e o *trapdoor* é cifrado com a chave simétrica K_C (que é um *nonce*).

5.3.4 Anonimato de Rota

Ao verificar as propriedades apresentadas por Tamashiro (2007), o protocolo ANODR apresentou as seguintes características:

- **Anonimato de rota em relação a nós fora da rota:** Não são capazes de associar os nós às rotas, pois há anonimato de identidade e ataques de tempo, conteúdo e volume não são executados;
- **Anonimato de rota em relação a nós pertencentes à rota:** Como a identidade de destino é revelada, nós intermediários internos (trabalhando em conluio) podem descobrir que pertencem à mesma rota. Ainda assim, a distância relativa é revelada pelo ataque de volume.

A garantia do anonimato de identidade foi verificada nas comparações anteriores. O anonimato em relação aos nós fora da rota é garantido, pois o ataque de volume não pode ser executado (de acordo com as comparações realizadas na seção anterior). O ataque de conteúdo (campo *seqnum*, conforme comparações anteriores) também não compromete o anonimato, pois um atacante que não pertence à rota não consegue determinar o caminho exato do pacote, pois ataques de análise de tempo não podem ser executados. Para verificar o ataque de análise de tempo, foi realizada uma nova simulação. Assim, foi observado que o protocolo adiciona um atraso (*delay*) aleatório antes de fazer o *broadcast* na rede.

Em relação aos nós pertencentes à rota, o anonimato é parcialmente garantido. A identidade de destino não é revelada (conforme análise anterior). Entretanto, o protocolo é suscetível ao ataque de conteúdo, permitindo determinar a localização aproximada do primeiro envio. Em um ataque coordenado, a rota da mensagem pode ser descoberta. Se um nó X, por exemplo, estiver comprometido, um adversário consegue fazer uma ligação entre dois pseudônimos de rota (para cada rota que passa pelo nó). Se *n* nós estão comprometidos - e fazem parte de uma rota consecutiva - então *n* + 1 nós são relacionados. Se os nós não forem consecutivos, o adversário pode criar segmentos, mas será difícil relacionar vários segmentos para formar a rota completa.

6 Conclusão

O presente estudo procurou apresentar um panorama na área de redes sem fio ad-hoc móveis, e ainda verificar e simular os levantamentos e comparativos realizados por Tamashiro (2007). O resultado da simulação do protocolo ANODR comprovou o anonimato de identidade do protocolo. No estudo comparativo realizado por Tamashiro (2007) é dito que a identidade de destino pode ser revelada. O resultado, por outro lado, demonstrou que isso não é possível. Os outros parâmetros de anonimato foram comprovados.

Em relação ao futuro dos protocolos que tentam prover anonimato nas redes sem fio ad-hoc móveis, é um pouco difícil prever o comportamento ideal de tais redes. A própria concepção dessas redes viabiliza um número grande de aplicações. O anonimato é apenas um dos itens de segurança. O conjunto de soluções de segurança para atender os requisitos vai depender do contexto da aplicação (energia, mobilidade, poder computacional, etc.). Assim, nenhuma solução poderá ser considerada ideal, já que a abrangência de aplicações é potencialmente infinita.

Como forma de verificar os protocolos anônimos apresentados, segue como sugestão a implementação do protocolo em aplicações reais de redes sem fio ad-hoc móveis. Assim, será possível montar uma rede entre dispositivos e simular ataques utilizando adversários físicos. Em um segundo momento, os vários protocolos disponíveis devem ser estudos no sentido de unificar os requisitos de anonimato, para então ser formalizado um protocolo de roteamento anônimo padrão.

Referências

Banerjee, Amal; Julien, Christine; Shmatikov, Vitality (2006) “Certificate Free Anonymous Routing for Mobile Ad Hoc Networks”, The University of Texas at Austin.

Boneh, D.; Franklin, M (2003) “Identity Based Encryption From Weil Pairing” em SIAM Journal of Computing: 586-615.

Corson, S.; Macker, J. (1999) “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”. IETF, January 1999. RFC 2501 (Informational). (Request for Comments, 2501). Disponível em: <http://www.ietf.org/rfc/rfc2501.txt>. Acesso em: Nov. 2008.

El-Khatib, K. *et al*; Franklin, M (2003) “Secure Dynamic Distributed Algorithm for Ad Hoc Wireless Networks”. In: International Conference on Parallel Processing Workshops. [S.l.: s.n.], 2003. p. 359–366.

Goldschlag, D.; Reed, M.; Syverson, P. (1999) “Onion Routing for anonymous and private internet connections. Communications of the ACM”, 42(2):39C4

Kong, J.; Hong, X. ANODR (2003) “ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks”. In: 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing. [S.l.: s.n.], p.291–302.

Qualnet (2008) “Qualnet 4.5.1 Guide”. Disponível em: < http://www.scalable-networks.com/publications/documentation/>. Acesso em: Mar. 2009.

SNT (2009) “Scalable Networks”. Disponível em:< http://www.scalable-networks.com/>. Acesso em: Mar. 2009

Tamashiro, Clytia Higa (2007) “Uma Análise de Protocolos de Roteamento Anônimo para Redes Sem Fio Ad Hoc Móveis”. Tese. (Mestrado Ciências da Computação) UFSC, Florianópolis..

Zhang, Y.; Liu, W.; Lou, W (2005) “Anonymous Communications in Mobile Ad Hoc Networks”. In: Annual Joint Conference of the IEEE Computer and Communications Societies. [S.l.: s.n.]. v. 3, p. 1940–1951.