Uma Rede Overlay Tolerante a Intrusões

Rafael R. Obelheiro* **e Joni da Silva Fraga†**

Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil
Email: rro@das.ufsc.br, fraga@das.ufsc.br

Resumo. *Este artigo apresenta ROTI, uma rede overlay tolerante a intrusões. A ROTI provê segurança para as mensagens e usa protocolos de encaminhamento de pacotes e roteamento tolerantes a faltas bizantinas. Quando da detecção de falhas, a rede overlay pode se reconfigurar, excluindo links faltosos e adicionando nós para preservar a segurança e a disponibilidade da rede.*

Abstract. *This paper introduces ROTI, an intrusion tolerant overlay network. ROTI provides message security and uses packet forwarding and routing protocols which tolerate Byzantine faults. When faults are detected, the overlay network can easily reconfigure itself, excluding faulty links while adding nodes in order to preserve network security and availability.*

1. Introdução

A crescente dependência da sociedade moderna em relação aos seus sistemas de informação impõe a esses sistemas requisitos cada vez mais severos de tolerância a faltas e segurança. Essa situação é exacerbada em sistemas ligados à Internet, que vem se tornando a cada dia que passa um ambiente mais hostil.

Os mecanismos de segurança tradicionais se concentram na prevenção de intrusões, enquanto os mecanismos de tolerância a faltas nem sempre são adequados para lidar com faltas maliciosas. Na confluência dessas duas correntes de pesquisa encontra-se a tolerância a intrusões, que se preocupa com incidentes de segurança sem depender da inviolabilidade dos componentes de segurança; ao contrário, admite-se que alguns desses componentes podem sofrer intrusões, e constrói-se o sistema de modo que ele desempenhe corretamente suas funções mesmo que essas intrusões se concretizem [9, 5, 17].

Uma característica importante em sistemas tolerantes a faltas ou a intrusões é a resiliência, que é capacidade do sistema de se ajustar a eventos adversos (como falhas de comunicação e intrusões) sem comprometer os seus requisitos de confiabilidade e segurança.

As redes *overlay*, que são redes lógicas construídas sobre redes físicas existentes, vêm sendo usadas para implementar serviços de rede com características diferenciadas, especialmente em relação ao que oferece a camada IP da Internet. Elas são interessantes

*Bolsista CAPES.
†Bolsista de Produtividade em Pesquisa do CNPq – Nível 2.

porque possibilitam que novos protocolos sejam implementados e testados em uma rede real, sem que seja necessário nenhum suporte especial por parte da infra-estrutura de rede já existente. Além disso, a flexibilidade inerente às redes *overlay* as torna ideais para a construção de serviços de rede resilientes. A literatura registra algumas propostas de redes *overlay* que atendem a requisitos de tolerância a faltas e de segurança [3, 2, 14]. Entretanto, não se conhece nenhuma proposta de rede *overlay* que seja tolerante a intrusões.

Este artigo propõe uma arquitetura de rede *overlay* tolerante a intrusões, que envolve propriedades de tolerância a faltas e de segurança. A propriedade de tolerância a faltas é mantida se entre dois nós corretos *A* e *B* do *overlay* houver pelo menos um caminho livre de faltas no *overlay* ligando os dois nós. Neste trabalho, a premissa em termos de semântica de falhas é o comportamento bizantino. As propriedades de segurança são mantidas se: (i) o conteúdo dos pacotes trocados entre *A* e *B* não puder ser examinado pelos nós intermediários (do *overlay* ou da rede física); (ii) o nó de destino *B* possa verificar que os pacotes recebidos são íntegros (não foram corrompidos em trânsito) e (iii) autênticos (foram efetivamente originados por *A*).

O restante do artigo está organizado da seguinte forma: a seção 2 apresenta os conceitos de intrusão e tolerância a intrusões, e a seção 3 apresenta conceitos de redes *overlay*. A rede *overlay* tolerante a intrusões proposta é apresentada na seção 4. A seção 5 discute trabalhos relacionados, e a seção 6 traz as conclusões e perspectivas futuras.

2. Tolerância a Intrusões

A pesquisa em segurança computacional tradicionalmente tem se concentrado em prevenir a ocorrência de violações de segurança. Devido a uma série de razões (como a aparentemente inevitável presença de vulnerabilidades), esta abordagem não tem sido muito bem sucedida. Em contrapartida, é possível supor como a abordagem de tolerância tipicamente aplicada a faltas acidentais poderia ser aplicada à segurança [17]:

- presumir que os sistemas permaneçam, até certo ponto, vulneráveis, independente das proteções que possuam;
- presumir que ataques contra componentes ou subsistemas podem acontecer, e que alguns desse ataques serão bem sucedidos;
- garantir que, a despeito destas hipóteses, o sistema como um todo permaneça seguro e capaz de desempenhar suas funções.

Uma intrusão pode então ser considerada como um evento que ocorre em um sistema, provocado por um indivíduo (ou por um processo automatizado iniciado por ele) e que resulta em, ou oportuniza, violação de uma ou mais das propriedades fundamentais da segurança (confidencialidade, integridade e disponibilidade). O conceito de intrusão pode ser, portanto, considerado um sinônimo para termos mais comumente usados em segurança computacional, tais como invasão e comprometimento de um sistema; o abuso de privilégios por parte de usuários autorizados também é uma intrusão. A tolerância a intrusões seria, por sua vez, a capacidade de um sistema de continuar a fornecer um serviço seguro a despeito de intrusões em um determinado número de seus componentes. Este conceito admite uma degradação na funcionalidade oferecida pelo sistema, desde que a sua segurança não seja violada.

3. Redes Overlay

Uma rede *overlay* é uma rede lógica construída sobre uma rede física existente [3, 1]. Redes *overlay* são interessantes porque permitem oferecer funcionalidades e qualidades de serviço diferenciadas sobre uma rede já estabelecida sem prejudicar a estabilidade e a robustez desta, causando um mínimo impacto sobre as aplicações existentes. Isto possibilita, por exemplo, que novos protocolos e idéias sejam testados com mais facilidade e possam amadurecer antes de serem transpostos para a rede real.

A figura 1 mostra um exemplo de como uma rede *overlay* se sobrepõe a uma rede física. Cada nó da rede *overlay* é também um nó da rede física. Uma conexão entre dois nós da rede *overlay* é chamada de *link* virtual, e corresponde à rota entre os respectivos nós na rede física. Cada nó é responsável por processar e rotear pacotes segundo critérios específicos da rede *overlay*; tais critérios normalmente dependem da aplicação a que a rede se destina. Cada rede *overlay* utiliza um esquema de endereçamento próprio, que pode ou não ser baseado no esquema de endereçamento da rede física. As conexões entre os nós do *overlay* são implementadas na rede física usando alguma forma de tunelamento (isto é, os pacotes da rede *overlay* são encapsulados em pacotes da rede subjacente), e não necessitam seguir nenhuma topologia predeterminada.

Assim como em uma camada de rede típica, as funções principais de uma rede *overlay* são o encaminhamento (*forwarding*) de pacotes, que determina como os elementos da rede (os roteadores) processam um pacote em trânsito para que ele chegue ao seu destino, e o roteamento, que é o processo através do qual o conhecimento sobre as diferentes rotas entre nós da rede é calculado, armazenado e disseminado.

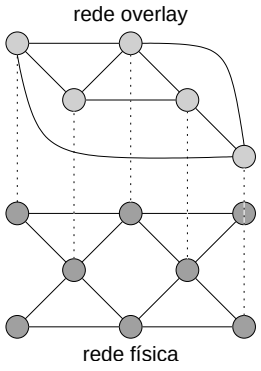

Figura 1: Rede *overlay* sobreposta a uma rede física

4. ROTI: Uma Rede Overlay Tolerante a Intrusões

Esta seção apresenta a ROTI, uma proposta de rede *overlay* tolerante a intrusões. Primeiramente coloca-se o modelo considerado e as premissas adotadas. Na seqüência, são discutidos aspectos de encaminhamento de pacotes, roteamento, detecção de falhas e reconfiguração da topologia da rede. A seção conclui com algumas considerações sobre a implementação da proposta.

4.1. Modelo e Premissas

O modelo da ROTI é mostrado na figura 2. Neste modelo, cada nó da rede *overlay* é um *host* Internet, sendo que cada *host* abriga apenas um nó *overlay*. Os nós do *overlay*

dividem-se em nós ativos e nós de reserva, inativos. A idéia é ter um *pool* de *hosts* Internet aptos a se tornarem nós da rede *overlay*. Em um dado momento, apenas um subconjunto destes *hosts* encontram-se ativos, formando a topologia corrente da rede; os demais *hosts* são considerados nós de reserva.

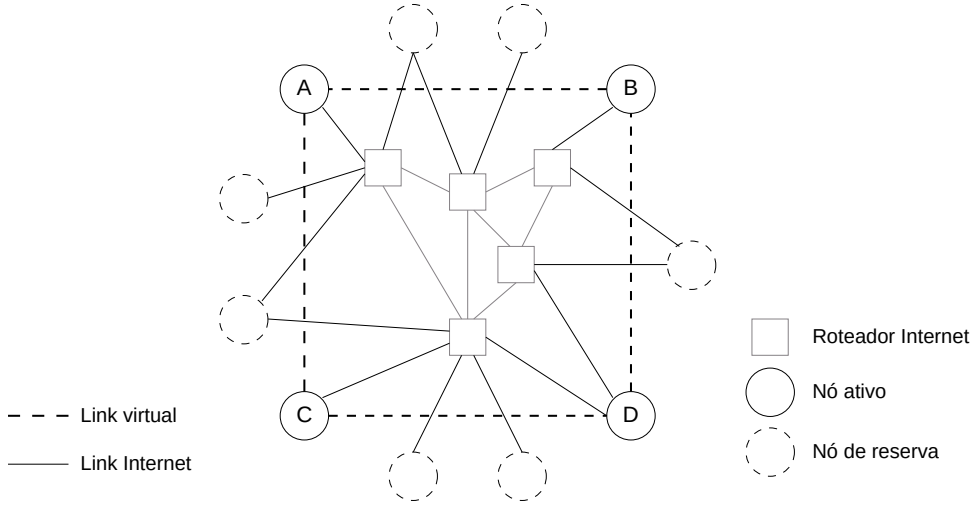

Figura 2: Modelo de rede *overlay* tolerante a intrusões

Os *links* virtuais são estabelecidos durante a ativação dos nós (a ativação de nós é discutida na seção 4.5). Em princípio, sempre que houver um *link* virtual ativo entre dois nós, a comunicação entre estes nós é feita diretamente através deste *link* virtual. A indireção, isto é, o envio de pacotes através de um ou mais nós intermediários, ocorre no caso de não existir um *link* virtual entre os nós de origem e destino. Por exemplo, na figura 2, o nó *A* tem que enviar mensagens para o nó *D* através dos nós *B* ou *C*, uma vez que não existe *link* virtual entre *A* e *D*.

Em uma dada comunicação entre nós do *overlay*, apenas os nós de origem e destino são considerados de confiança. Os demais nós, sejam eles do *overlay* ou da rede física subjacente, podem exibir comportamento bizantino, agindo sozinhos ou formando conluios. Um nó faltoso pode interceptar, modificar ou injetar pacotes, criar laços na rede, descartar pacotes de forma seletiva, rotear pacotes através de caminhos sub-ótimos ou fazer com que um caminho pareça mais curto ou mais longo do que realmente é.

Os mecanismos de detecção de falhas usados na ROTI (seção 4.4) identificam *links* virtuais com comportamento bizantino. Um *link* virtual pode exibir comportamento bizantino se pelo menos uma das três condições abaixo for satisfeita:

i. pelo menos um dos nós *overlay* que formam o *link* for faltoso;
ii. pelo menos um dos *links* da rede física usados pelo *link* virtual for faltoso;
iii. pelo menos um dos nós da rede física por onde passa o *link* virtual for faltoso.

Para aumentar a probabilidade de um caminho livre de falhas ser encontrado, é desejável que haja caminhos disjuntos entre cada par de nós da rede *overlay*, ou seja, caminhos que não compartilhem nós ou *links* tanto do *overlay* como da rede física (com exceção dos nós de origem e destino). A inexistência de caminhos disjuntos entre um determinado par de nós *overlay A* e *B* significa que existe pelo menos um nó ou *link* que, se for faltoso, pode interromper completamente a comunicação entre *A* e *B*. Na prática,

uma boa maneira de obter caminhos disjuntos é ter cada nó do *overlay* localizado em redes *multihomed* (conectadas a mais de um provedor de *backbone*). O uso de *multihoming* é recomendado já há algum tempo para combater ataques de negação de serviço (DoS—*Denial of Service*) [12]. Embora os protocolos de roteamento e encaminhamento partam da premissa de que a rede física permanece conectada, caso a rede sofra uma partição a comunicação dentro dos seus componentes conectados não é afetada negativamente, embora a capacidade de adaptação da rede (seção 4.5) possa ser comprometida.

4.1.1. Mecanismos Criptográficos

Uma infra-estrutura de chaves públicas (PKI—*Public Key Infrastructure*) é utilizada para dar suporte aos mecanismos criptográficos usados nos protocolos de roteamento e encaminhamento de pacotes. A PKI considerada segue um modelo de confiança descentralizado, baseado no SPKI (*Simple Public Key Infrastructure*) [7]. Este tipo de modelo de confiança elimina o ponto único de falhas representado pela autoridade certificadora (AC) presente nas PKIs tradicionais, com modelo de confiança centralizado.

Para poder ser admitido na rede *overlay*, um nó deve apresentar um certificado de autorização SPKI que lhe confira o direito de acesso à rede e que seja assinado por mais de um emissor através de um esquema de *threshold subjects* [8].[1] A idéia é que o comprometimento de um único emissor não possibilite a inclusão de nós forjados na rede nem a exclusão (pela autenticação incorreta) de nós válidos.

Sempre que possível, os mecanismos criptográficos adotados baseiam-se em chaves secretas compartilhadas entre cada par de nós da rede. Essa preferência deve-se a razões de desempenho. As chaves secretas são estabelecidas sob demanda através de um protocolo Diffie-Hellman [6], que utiliza as chaves públicas disponíveis através da PKI para estabelecer as chaves secretas de forma segura.

4.2. Encaminhamento de Pacotes

O encaminhamento de pacotes determina como os roteadores processam pacotes de forma a transmiti-los da origem para o destino. O caso mais simples de encaminhamento é quando existe um *link* conectando os nós de origem e destino. A dificuldade surge quando não existe uma conexão direta entre os nós comunicantes. A solução mais primitiva para o problema é usar *flooding*: o nó de origem encaminha o pacote para todos os seus vizinhos, e cada roteador encaminha o pacote para todos os seus vizinhos com exceção daquele de onde veio o pacote. Isso garante que o pacote chegue a todos os nós da rede—o que trivialmente inclui o destinatário—mas é bastante custoso, pois o número de duplicatas de um pacote varia exponencialmente com o tamanho da rede.

A solução geralmente adotada, porém, é encaminhar o pacote por um conjunto de nós intermediários para chegar ao nó de destino, o que é bem menos custoso do que disseminar pacotes por toda a rede. Este caminho através de nós intermediários é chamado de rota, e é determinado por protocolos de roteamento, que são discutidos na seção 4.3. Por enquanto, considera-se que um nó *A* que deseja se comunicar com um nó *B* pode determinar uma rota apropriada na rede *overlay* para transmitir os seus pacotes.

[1] *Threshold subjects* determinam que *k* dentre *n* principais ("sujeitos") assinem um certificado de autorização para que ele seja válido.

Na ROTI, o encaminhamento de pacotes envolve autenticação, confidencialidade e confiabilidade. A autenticação garante a origem dos dados que estão sendo transmitidos, enquanto a confidencialidade garante que nenhum nó intermediário possa inspecionar o conteúdo dos pacotes transmitidos. A confiabilidade, por sua vez, garante que os pacotes sejam entregues mesmo na presença de faltas e intrusões na rede, desde que exista um caminho livre de faltas entre os nós comunicantes. O mecanismo de encaminhamento de pacotes utilizado considera tanto faltas acidentais quanto maliciosas.

A autenticação de origem (que garante integridade) é implementada através de assinaturas digitais no nó de origem, que são verificadas em cada roteador no caminho. Um nó correto encaminha um pacote apenas se a verificação da sua assinatura for bem sucedida; caso essa verificação falhe, o nó descarta o pacote, e registra uma falha para o *link* através do qual este pacote chegou. A confidencialidade, por sua vez, é obtida através da cifragem do conteúdo dos pacotes com um algoritmo criptográfico simétrico.

A confiabilidade na comunicação é obtida através da detecção e recuperação de falhas. A detecção de falhas é baseada em reconhecimentos positivos (ACKs). A técnica de ACKs funciona bem quando a rota usada apresenta apenas perda esporádica de pacotes. Porém, quando uma rota é interrompida ou passa a apresentar uma incidência muito grande de perdas esta não pode ser mais usada. Nessa situação, diferentes estratégias de recuperação de falhas podem ser adotadas:

(a) Mudança de rota: escolhe-se uma rota alternativa para o destino. Como o protocolo de roteamento oferece um mecanismo de localização de faltas, a nova rota deve evitar os *links* reportados como faltosos.
(b) *Flooding*: caso a rota não funcione, o pacote é transmitido através de *flooding*.

A ROTI utiliza a primeira estratégia, que é menos confiável porém bem menos custosa do que a segunda.

O encaminhamento de pacotes na ROTI ocorre da seguinte maneira. Se existe um *link* virtual ativo entre os nós de origem e destino, os pacotes são transmitidos através deste *link*, e o seu recebimento é confirmado através de ACKs. Quando não existe *link* virtual entre os nós, ou quando o *link* é declarado faltoso, tenta-se obter uma rota no *overlay* entre os nós de origem e destino através do protocolo de roteamento. Se existe uma rota apropriada, os pacotes passam a ser encaminhados por esta nova rota. Neste caso, utiliza-se *source routing*, com o nó de origem especificando no cabeçalho do pacote a rota completa a ser usada. Os nós intermediários precisam então verificar a assinatura do pacote (descartando pacotes com assinatura inválida) e enviá-lo para o próximo nó da rota. Caso a rota em uso contenha um *link* que venha a ser declarado faltoso, solicita-se ao protocolo de roteamento uma nova rota. Se o protocolo de roteamento não encontrar uma rota apropriada, o nó de origem considera o nó de destino inalcançável, e sinaliza essa condição para a camada superior.

4.3. Roteamento

Sempre que um pacote não pode ser transmitido diretamente através de um *link* virtual, ele é encaminhado ao destino através de nós intermediários. Para que isso seja possível, é necessário determinar uma rota que parta do nó de origem, passe por um ou mais nós intermediários e chegue ao nó de destino. Um protocolo de roteamento é o

responsável por armazenar e disseminar informações sobre as rotas entre os nós da rede de forma a permitir a escolha da melhor rota entre uma origem e um destino específicos.

Protocolos de roteamento podem ser proativos e reativos (sob demanda). Protocolos proativos são aqueles onde os roteadores trocam informações de roteamento periodicamente, independente da utilização das rotas. Neste caso, quando surge a necessidade de encaminhar um pacote o roteador já conhece a rota para o nó de destino, e pode transmitir o pacote imediatamente. Em contrapartida, este tipo de protocolo gera um *overhead* constante em função da troca de informações de roteamento, que independe das rotas estarem sendo usadas ou não. Os protocolos reativos, por outro lado, iniciam um processo de descoberta de rota apenas quando dados necessitam ser enviados. A rota descoberta fica então armazenada em *cache* até ser descartada por falta de uso (após um determinado período) ou por mudanças na topologia da rede. Este tipo de protocolo introduz uma latência adicional na transmissão quando é necessário descobrir uma nova rota.

A rede *overlay* tolerante a intrusões utiliza um protocolo de roteamento sob demanda. Este tipo de protocolo foi escolhido visando a um melhor desempenho da rede quando não existem falhas. Como pacotes só são encaminhados através de nós intermediários quando o *link* virtual entre os nós de origem e destino não está disponível, a latência adicional imposta pelo processo de descoberta de rotas é mais interessante do que o *overhead* constante de um protocolo proativo. Além disso, os protocolos de roteamento proativos possuem algumas características que os tornam menos indicados para o modelo de faltas considerado na ROTI [15].

A estratégia de roteamento adotada na ROTI é baseada em uma proposta para roteamento seguro em redes *ad hoc* sem fio [4], adaptada para uso em redes *overlay*. Essa estratégia implementa descoberta de rotas, detecção de falhas e gerenciamento de métricas de roteamento.

O processo de descoberta de rotas funciona da seguinte maneira: um nó *A* que deseja obter uma rota para um nó *B* monta uma requisição de rota assinada, que é disseminada na rede *overlay* através de *flooding*. A requisição é propagada até *B*, onde a sua autenticidade é confirmada e uma resposta é enviada para *A*, também através de *flooding*. À medida em que a resposta vai se propagando através da rede, os nós intermediários vão calculando a rota entre *A* e *B*. Para determinar a melhor rota, cada nó intermediário soma o custo da rota até o nó anterior (presente no pacote) com o custo do *link* entre o nó anterior e si próprio; o novo custo é armazenado no próprio pacote, que é assinado pelo nó antes de ser enviado para o próximo roteador. Todos os nós intermediários verificam a validade das assinaturas tanto de requisições como de respostas com o intuito de evitar que pacotes inválidos sejam disseminados através da rede. O *flooding* é usado neste caso para garantir a máxima robustez, evitando que requisições ou respostas se percam ao passarem por um *link* faltoso.

Diferente da proposta original [4], na ROTI o nó de origem armazena em um *cache* não apenas a melhor rota para o destino mas todas as rotas disjuntas (isto é, que não compartilham nenhum nó intermediário ou *link* virtual) obtidas. Isso permite que, ao ser requisitada uma nova rota em função de faltas na rota atual, o subsistema de roteamento possa fornecer imediatamente uma rota alternativa disjunta enquanto o protocolo de descoberta de rotas é executado novamente, eliminando a latência inicial normalmente

imposta pelo processo de descoberta.

Outro ponto importante é que o nó de origem descarta as rotas que incluam *links* faltosos, conforme será detalhado na seção 4.5. Essa estratégia de evitar *links* faltosos (em oposição a técnicas de detecção e exclusão de nós faltosos) permite que a rede tolere faltas bizantinas mesmo que exista uma maioria de nós faltosos e sem que seja necessário recorrer a protocolos de acordo para decidir de forma distribuída quais nós são faltosos e devem ser excluídos da rede.

4.4. Detecção de Falhas

O principal mecanismo de detecção de falhas na ROTI são os ACKs, que detectam falhas na transmissão de pacotes. Quando um ACK não é recebido antes do seu *timeout* expirar, o nó de origem registra uma perda para a rota em uso e retransmite o pacote. Quando o número de pacotes perdidos ultrapassa um dado limiar, considera-se que há uma falha no caminho, e inicia-se um processo para localizar o *link* faltoso. Esse processo consiste em retransmitir o pacote pela mesma rota, especificando no cabeçalho que todos os nós intermediários devem enviar um ACK para o nó de origem confirmando o recebimento do pacote. O *link* situado entre o último ACK recebido e o primeiro ACK perdido é declarado faltoso. Por exemplo, na figura 3, se o nó A envia um pacote para o nó E e recebe ACKs de B e C (mas não de D ou E), ele considera o *link* (C,D) faltoso.

É importante notar que o uso de ACKs oferece uma localização imprecisa de faltas. Seja o exemplo da figura 3, onde um nó A envia um pacote para um nó E e não recebe o ACK de D antes de expirar o respectivo *timer*. Isso pode acontecer por diversas razões: o pacote é perdido (caso ilustrado na figura), o ACK é perdido, D não envia o ACK, etc. Estas razões podem ser resumidas em uma ou mais de três causas: C é faltoso, D é faltoso e/ou o *link* (C,D) é faltoso. A detecção de falhas, nesse caso, pode reportar como suspeitos tanto o *link* (C,D) como os nós C e D. É importante observar que ambos os nós têm que ser considerados suspeitos, uma vez que é impossível (usando apenas ACKs) determinar qual deles é efetivamente o faltoso caso apenas um deles o seja.

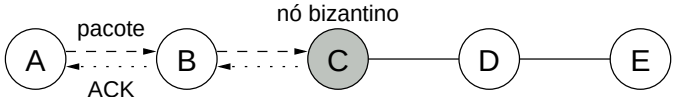

Figura 3: Detecção de falhas com ACKs

Em uma rede em que os nós e *links* podem apresentar comportamento bizantino, os ACKs podem ser manipulados pelos elementos faltosos. No exemplo da figura 3, o nó bizantino C pode deixar de transmitir ACKs do nó E em uma tentativa de incriminar o *link* correto (D,E). Para evitar que isso aconteça, a lista de nós que devem enviar ACKs para o origem e os próprios ACKs são cifrados com a chave secreta compartilhada entre cada nó do caminho e o nó de origem. Desta forma, um nó bizantino não pode verificar se um pacote que está sendo encaminhado contém um ACK ou não, o que o impede de interferir seletivamente na transmissão (o nó pode filtrar pacotes mesmo assim, mas isso acabaria por incriminar um de seus próprios *links*).

Os mecanismos criptográficos usados na ROTI também fornecem dados para a detecção de falhas. Voltando ao exemplo da figura 3, se D recebe um pacote com assinatura inválida de C ele registra uma falta para o *link* incidente (C,D), uma vez que pacotes

com assinatura inválida não devem ser propagados na rede. Certificados SPKI com assinaturas inválidas também representam uma falta do *link* por onde foram transmitidos. Faltas vinculadas aos mecanismos criptográficos não precisam ser localizadas, uma vez que só podem ser atribuídas ao *link* incidente ou ao nó vizinho.

4.5. Reconfiguração da Rede

Uma característica importante da ROTI é a sua capacidade de adaptação que consiste na reconfiguração dinâmica da topologia da rede. Existem dois aspectos envolvidos nessa reconfiguração: exclusão de *links* virtuais e ativação de nós de reserva.

Basicamente, um *link* virtual é excluído após apresentar uma determinada taxa de faltas λ_{max}. A cada falta registrada pelos mecanismos de detecção e localização de faltas, a taxa de faltas $\lambda_{I,J}$ do *link* (I,J) apontado como faltoso é analisada. Se $\lambda_{I,J} \leq \lambda_{max}$, o *link* (I,J) é posto em quarentena. Durante a quarentena, o *link* é ignorado, e qualquer rota que o contenha é descartada. Após um período t_q, ou após algum pacote válido chegar ao nó através de uma rota que passe pelo *link* em quarentena, este é reabilitado. A quarentena é usada para evitar que um *link* que esteja apenas temporariamente faltoso (devido a congestionamento na rede, por exemplo) seja excluído indevidamente por apresentar um grande número de faltas em um curto espaço de tempo.

Quando $\lambda_{I,J} > \lambda_{max}$, o *link* (I,J) é considerado permanentemente faltoso e deve ser excluído. Se (I,J) é um *link* local (ou seja, faz parte do conjunto de *links* virtuais do nó que deseja excluí-lo), ele deixa de ser usado para as comunicações deste nó: nenhum pacote é enviado através de (I,J), pacotes eventualmente recebidos através de (I,J) são descartados, e o *link* é ignorado no protocolo de roteamento. Se, por outro lado, (I,J) não for local, ele simplesmente passa a ser ignorado no protocolo de roteamento (rotas que passam por (I,J) são descartadas). Cabe ressaltar que a exclusão de um *link* virtual é uma decisão local: cada nó monitora os *links* da rede, e decide quando e quais excluir, de forma independente. Isso significa, por exemplo, que um nó correto *A* deve encaminhar pacotes originados por outros nós que passam por um *link* não local (I,J), mesmo que este *link* tenha sido excluído por *A*.

Quando um nó *A* detecta que um nó *B* ficou inacessível (isto é, todos os *links* virtuais para *B* foram removidos), ele tenta ativar um nó reserva *R* que seja topologicamente próximo a *B* (a intenção aqui é que os *hosts* ligados a *B*, ao perceber que este se tornou não funcional, migrem para um nó próximo). Para ativar o nó de reserva *R*, o nó *A* envia uma mensagem de ativação assinada `activate(R,B)` para *R*, e difunde (através de *flooding*) uma mensagem `newnode(R,B)` no *overlay*. Os nós do *overlay* que receberem `newnode(R,B)` e concordarem com a ativação de *R* também enviam `activate(R,B)` para o novo nó. Se *R* receber um mínimo de ϕ mensagens de ativação, ele tenta se tornar ativo no *overlay*. Para isso, ele tenta estabelecer *links* virtuais com ρ nós *overlay* ($\rho \leq \phi$) dentre os que lhe enviaram `activate(R,B)`. A ativação é validada com o estabelecimento do *link*, que só ocorre se *R* provar que detém pelo menos ϕ mensagens `activate(R,B)`; isso é provado incluindo as mensagens de ativação na requisição de estabelecimento de *link* virtual.

A lista de nós de reserva é instalada manualmente nos primeiros nós da rede. Essa lista é assinada pelas chaves emissoras de certificados SPKI. Quando um nó reserva é ativado, os nós que estabelecem *links* virtuais com ele enviam-lhe as suas cópias da

lista. Listas com assinatura inválida são descartadas, e são consideradas um indício de comportamento malicioso por parte de quem as envia.

4.6. Aspectos de Implementação

Para demonstrar a aplicabilidade da proposta e possibilitar a compreensão das suas implicações práticas, encontra-se em desenvolvimento um protótipo da ROTI. O *software* em cada nó do *overlay* tem a estrutura mostrada na figura 4.

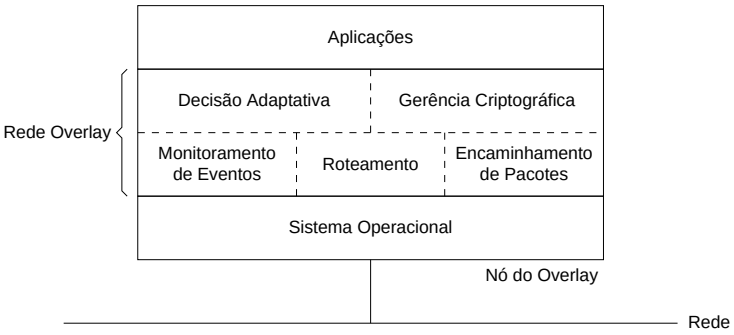

Figura 4: Arquitetura do protótipo

Os módulos de encaminhamento de pacotes e de roteamento implementam as funções apresentadas nas seções 4.2 e 4.3. Os pacotes da rede *overlay* são transmitidos na rede física como pacotes UDP, que oferecem uma comunicação não confiável com um *overhead* mínimo. Os pacotes recebidos da camada de superior (transporte) são cifrados através de um algoritmo simétrico, utilizando uma chave secreta compartilhada entre o nó de origem e o de destino.

O módulo de monitoramento de eventos é responsável pela detecção e localização de faltas, conforme descrito na seção 4.4. Este módulo baseia-se em informações de falhas fornecidas pelos demais módulos, como ACKs perdidos e assinaturas digitais inválidas.

O módulo de gerência criptográfica encarrega-se de recuperar, verificar e armazenar os certificados SPKI usados na rede *overlay*. Além disso, ele também é responsável por estabelecer e armazenar as chaves secretas compartilhadas com os outros nós da rede.

O módulo de decisão adaptativa recebe as informações do módulo de monitoramento de eventos, verifica se alguma ação de reconfiguração precisa ser tomada e determina qual seria essa ação. Em um nó de reserva, este módulo também é responsável por decidir quando o nó está apto a ser ativado e por dirigir o processo de ativação (seção 4.5).

5. Trabalhos Relacionados

Embora não haja registro na literatura de outras propostas de rede *overlay* tolerante a intrusões, diversas experiências possuem intersecção com este trabalho e podem ser usadas como referência para diferentes aspectos da arquitetura proposta.

Existem diversas experiências que utilizam redes *overlay* para oferecer serviços seguros ou tolerantes a faltas. Na RON [3] a rede *overlay* é completamente conectada, e cada nó monitora o estado dos seus *links* virtuais, redirecionando pacotes através de nós intermediários em caso de falha do *link*. A diferença crucial entre a RON e a ROTI

é que a primeira considera apenas faltas de *crash*, além de não oferecer mecanismos de segurança. O Spines [2] propõe uma rede *overlay* com características de segurança e confiabilidade. No entanto, os resultados do Spines até o momento enfatizam a comunicação confiável com alto desempenho usando ACKs *hop*-a-*hop* (em vez de fim-a-fim) [1]; os mecanismos de segurança e reconfiguração, embora mencionados em [2], não são definidos. Essa abordagem de comunicação confiável do Spines é considerada complementar à funcionalidade da ROTI, e pode ser utilizada em uma futura revisão da arquitetura.

O SecComm [11] é um serviço de comunicação ponto a ponto *survivable*, que utiliza diversidade de métodos para evitar que a quebra de um algoritmo ou chave comprometa as propriedades de mecanismos criptográficos. Isso é conseguido, por exemplo, cifrando dados múltiplas vezes (com algoritmos e/ou chaves de sessão diferentes). Assim como no caso do Spines, a abordagem do SecComm pode complementar os mecanismos criptográficos da ROTI.

A segurança no roteamento não é uma preocupação nova; Papadimitratos e Haas [15] trazem um *survey* envolvendo protocolos usados na Internet. O trabalho pioneiro sobre redes tolerantes a faltas bizantinas é a tese de Perlman [16], que combina *flooding* e roteamento *link state*; a influência desse trabalho pode ser constatada em muitas propostas atuais. O roteamento considerando faltas bizantinas adquire maior importância em redes *ad hoc* sem fio, onde a possibilidade dos nós estarem expostos a um ambiente hostil é maior do que em redes convencionais; propostas nesse contexto incluem OSDBR [4] e Ariadne [13].

A FVPN [10] é uma VPN (*Virtual Private Network*) tolerante a faltas com objetivos similares aos da ROTI. A primeira diferença entre a FVPN e a ROTI é que a primeira é projetada para uso dentro de um domínio de roteamento e utiliza informações de roteamento já disponíveis dentro desse domínio, enquanto que a ROTI não impõe qualquer restrição à topologia. Outro ponto é que a FVPN preconfigura rotas alternativas antes de utilizá-las, minimizando a latência na recuperação de falhas em troca de um possível desperdício de recursos de rede. Além disso, a FVPN usa apenas redundância na rede física, não utilizando indireção através da rede *overlay*; isso faz com que em determinadas situações a ROTI consiga entregar pacotes e a FVPN não. Entretanto, a principal diferença está na semântica de falhas: a FVPN age apenas em reação a falhas sinalizadas pelo protocolo de roteamento *link state*, o que não inclui comportamento bizantino dos elementos de rede.

6. Conclusões

Este artigo apresentou a ROTI, a primeira proposta de rede *overlay* tolerante a intrusões. Todos os mecanismos de tolerância a faltas e segurança usados na ROTI têm o objetivo de tornar a rede resiliente a faltas e intrusões. Neste contexto, destacam-se a natureza adaptativa da rede e a grande autonomia que cada nó tem para tomar suas próprias decisões.

Embora as premissas adotadas na ROTI garantam apenas de forma probabilista a resiliência da rede, é possível, através de premissas adicionais, fornecer garantias a respeito do funcionamento da rede. Por exemplo, se existirem $f+1$ caminhos disjuntos entre cada par de nós do *overlay* a rede é capaz de entregar pacotes entre dois nós corretos

mesmo que existam f faltas (na rede física ou no *overlay*). A opção por garantias probabilistas tem o objetivo justamente de evitar premissas excessivamente fortes que limitem a aplicabilidade da ROTI.

O próximo passo é, logicamente, a implementação de um protótipo da ROTI, trabalho que já se encontra em andamento. Esse protótipo permitirá a coleta de dados sobre o desempenho da arquitetura proposta e uma melhor compreensão do seu funcionamento, possibilitando a identificação de aspectos que precisam ser refinados. Além disso, os experimentos realizados com o protótipo servirão para ajustar os parâmetros ϕ, ρ, λ_{max} e t_q usados na reconfiguração da rede.

Referências

[1] Y. Amir and C. Danilov. Reliable Communication in Overlay Networks. In *Proc. Int'l Conf. on Dependable Systems and Networks*, pages 511–520, San Francisco, CA, June 2003.

[2] Y. Amir, C. Danilov, and C. Nita-Rotaru. High Performance, Robust, Secure and Transparent Overlay Network Service. In *Proc. Int'l Workshop on Future Directions in Distributed Computing*, Bertinoro (Italy), June 2002.

[3] D. G. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. 18th ACM Symp. on Operating Systems Principles*, pages 131–145, Banff, AB (Canada), Oct. 2001.

[4] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *Proc. ACM Workshop on Wireless Security*, pages 21–30, Atlanta, GA, Sept. 2002.

[5] Y. Deswarte, L. Blain, and J.-C. Fabre. Intrusion Tolerance in Distributed Computing Systems. In *Proc. IEEE Symp. on Security and Privacy*, pages 110–121, Oakland, CA, 1991.

[6] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, Nov. 1976.

[7] C. M. Ellison. SPKI Requirements. RFC 2692, Internet Engineering Task Force, Sept. 1999.

[8] C. M. Ellison, B. Frantz, B. W. Lampson, R. L. Rivest, B. Thomas, and T. Ylönen. SPKI Certificate Theory. RFC 2693, Internet Engineering Task Force, Sept. 1999.

[9] J. S. Fraga and D. Powell. A Fault and Intrusion-Tolerant File System. In *Proc. 3rd International Congress on Computer Security*, pages 203–218, Dublin (Ireland), Aug. 1985.

[10] J. Han, G. R. Malan, and F. Jahanian. Fault-Tolerant Virtual Private Networks within An Autonomous System. In *Proc. 21st Symp. on Reliable Distributed Systems*, pages 41–50, Suita (Japan), Oct. 2002.

[11] M. A. Hiltunen, R. D. Schlichting, and C. A. Ugarte. Building Survivable Services Using Redundancy and Adaptation. *IEEE Transactions on Computers*, 52(2):181–194, Feb. 2003.

[12] A. Householder, A. Manion, L. Pesante, G. M. Weaver, and R. Thomas. Managing the Threat of Denial-of-Service Attacks. CERT Coordination Center, Oct. 2001.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proc. 8th Annual Int'l Conf. on Mobile Computing and Networking*, pages 12–23, Atlanta, GA, Sept. 2002.

[14] A. D. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. In *Proc. ACM SIGCOMM Conf.*, pages 61–72, Pittsburgh, PA, Aug. 2002.

[15] P. Papadimitratos and Z. J. Haas. Securing the Internet Routing Infrastructure. *IEEE Communications Magazine*, 40(10):60–68, Oct. 2002.

[16] R. J. Perlman. *Network Layer Protocols with Byzantine Robustness*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, Jan. 1988.

[17] P. Veríssimo, N. F. Neves, and M. Correia. Intrusion-Tolerant Architectures: Concepts and Design. DI/FCUL TR 03-05, Department of Informatics, University of Lisbon, Apr. 2003.