Capítulo

3

Arquiteturas de Rede para a Próxima Geração da Internet

Carlos Kamienski, Dênio Mariz, Djamel Sadok, Stênio Fernandes

Abstract

As the Internet became a worldwide phenomenon, a number of solutions have been proposed to expand its functionalities beyond its original design. All research proposals intend to accomplish new features and try to solve some new issues not expected previously. For this reason, the Internet technology evolution can be seen as a "bunch of patches". In most cases, the adopted solutions violate fundamental principles that were established formerly, increase complexity, and are not fully interoperable. In view of the fact that all proposals try to address legitimate problems, it is imperative to rethink the current Internet architecture. The main goal of this document is to present a conceptual view of the original Internet design and discuss the most prominent proposals for changing its architecture from the Internet research community.

Resumo

Desde que a Internet alcançou dimensão global, várias soluções específicas foram adicionadas ao seu projeto original, visando atender novas demandas e resolver problemas não previstos originalmente, de maneira que a Internet tem evoluído como uma "colcha de retalhos". Em muitos casos, as soluções adotadas violam princípios estabelecidos pela própria concepção da Internet, aumentam sua complexidade e exibem problemas de interoperabilidade. Isto tem acontecido porque "não é possível fazer da Internet algo para o qual ela não foi projetada". Uma vez que todos esses ajustes buscam resolver problemas legítimos, é preciso repensar a arquitetura da Internet. O principal objetivo deste trabalho é apresentar os aspectos conceituais de projeto da Internet e discutir as principais propostas da comunidade científica para mudanças na sua arquitetura.

3.1. Introdução

Atualmente, a Internet é utilizada por uma grande quantidade de usuários ao redor do mundo e desempenha um papel essencial na vida das pessoas e organizações. De fato, para cerca de 800 milhões de pessoas, seria impensável hoje se privar de aplicações como correio eletrônico, navegação da Web, sistemas de mensagens instantâneas, aplicações de compartilhamento de arquivos, acesso a bancos, compras on-line, jogos em rede e até mesmo telefonia (VoIP). No entanto, essa grande dependência da rede nos faz hoje reféns da sua própria tecnologia, ou seja, grandes transformações não são possíveis em nome da preservação de investimentos e continuidade dos serviços.

Antes de atingir esta amplitude e influência na vida das pessoas, a Internet nasceu pequena, projetada por acadêmicos para ser uma rede em que a mudança constante era a sua principal característica. A rede era uma fonte de experiências para transcender os limites do conhecimento sobre a capacidade dos seres humanos de utilizar os computadores para melhorar a comunicação entre si. Por este motivo, ela precisava ser flexível e se reinventar constantemente. Desde que a Internet foi criada, em 1969, várias transformações ocorreram. A mais drástica foi a introdução dos protocolos TCP/IP, que só se concretizou após cerca de 15 anos de sua existência. É interessante ressaltar também que a Web e o protocolo HTTP, praticamente sinônimos de Internet, somente foram introduzidos na década de 1990, portanto quando a Internet já estava na sua maioridade. Além disso, a Internet atual está funcionando há mais de 30 anos e está continuamente recebendo novas missões e ajustes necessários para uma nova categoria de serviços avançados e novas tecnologias de rede.

A importância da Internet, o ritmo acelerado do desenvolvimento de novas tecnologias e a preservação dos investimentos e manutenção da estabilidade da rede, geram hoje uma situação de dualidade quanto à sua evolução. Por um lado, a rede está em constante evolução para atender novas demandas. Essas modificações, entretanto, não são profundas e fazem da atual arquitetura da Internet uma colcha de retalhos. Por outro lado, grandes transformações são limitadas.

Este documento aborda conceitos, modelos, características, taxonomias, problemas e desafios encontrados no projeto de novas arquiteturas para a Internet e apresenta uma visão geral das propostas mais discutidas pela comunidade científica. O objetivo é incentivar o leitor a uma mudança de paradigma com relação à sua compreensão da Internet, fornecendo subsídios para expandir o discernimento sobre seu funcionamento e princípios básicos. De um modo mais específico, este documento procura auxiliar o leitor a desenvolver competências para:

- Compreender a necessidade de mudanças na arquitetura da Internet para suportar novos serviços e a complexidade inerente para sua implementação;
- Identificar os princípios básicos utilizados para o projeto e implantação da Internet e que até hoje em dia são extensivamente usados pelos membros da comunidade da Internet no projeto de novas tecnologias;
- Identificar as principais questões de projeto da Internet, que devem ser modificadas a fim de que ela possa evoluir para cumprir novos objetivos;

- Conhecer as principais propostas da comunidade científica para mudanças arquiteturais na Internet, bem como entender como elas se relacionam entre si e seu impacto sobre a Internet atual;
- Perceber as dificuldades de se promover grandes modificações na Internet atual, bem como ter ciência de que freqüentemente novas propostas novas são apenas propostas antigas revestidas de uma roupagem diferente.

Um aspecto importante deste documento é apresentar questões conceituais de projeto da Internet, quando muitos alunos, profissionais ou professores apenas a vêem como algo finalizado, intangível e imutável. Isto é interessante porque quebra com alguns paradigmas no ensino e pesquisa em redes de computadores.

Na seqüência, a seção 3.1 prossegue com a motivação para as mudanças na arquitetura da Internet. A seção 3.2 apresenta os princípios fundamentais da Internet, que devem se considerados ao tentar modificá-la. Na seção 3.3, seis importantes questões de projeto na Internet são identificadas e os seus problemas são relatados. Algumas propostas de alteração da Internet são apresentadas na seção 3.4. As principais contribuições dessas propostas são sumarizadas na seção 3.5. Finalmente, a seção 3.6 conclui o texto com as descobertas obtidas no desenvolvimento do trabalho.

3.1.1. Por que o Modelo da Internet não Evoluiu?

A Internet é baseada no conceito de comutação de pacotes, inventado por Leonard Kleinrock em 1961, e posto em prática na Arpanet, que entrou em operação em 1969 [55]. A Internet desde o princípio foi concebida para interconectar múltiplas redes distintas, em vez de seguir uma topologia e projeto específico. O modelo de serviço de "melhor esforço" também sempre esteve presente, rezando para que os roteadores encaminhem os pacotes ao seu destino da melhor maneira possível. Se isso não for possível, os pacotes são descartados e possivelmente retransmitidos pelo sistema final transmissor. Os roteadores são considerados "caixas pretas", não retendo nenhuma informação sobre os pacotes que encaminham.

Embora muitas pessoas acreditem que o conjunto de protocolos TCP/IP esteve sempre ligado à idéia da Internet, eles somente foram amplamente adotados em 1983, substituindo os antigos protocolos NCP. Isso mostra que naquela época ela suportava mudanças muito mais drásticas do que as que hoje se propõe, como por exemplo, a transição de IPv4 para IPv6. Por sinal, uma questão curiosa sobre TCP/IP é que primeiramente foi proposto o protocolo TCP, que incluía as funções de rede e de transporte. Somente mais tarde ele foi dividido em TCP e IP.

Nos últimos anos, desde que a Internet foi aberta ao uso comercial em 1993, a sua arquitetura transformou-se de algo flexível, projetado para sofrer mudanças, em uma estrutura engessada, que não pode ser facilmente alterada. As razões para isso são variadas, desde a preservação de investimentos realizados, passando pelo risco de colapsos operacionais, até o incômodo de substituição do software em milhões de estações em posse dos usuários. No entanto, isso não significa que ela está parada. Desde que a Internet alcançou dimensão global, várias soluções específicas, para problemas não previstos e verificados sob demanda foram adicionadas ao seu projeto original, de maneira que a Internet tem evoluído como uma "colcha de retalhos". Em outras palavras, como alterações "limpas" não são possíveis, várias alterações

pragmáticas estão ocorrendo, mas que violam alguns preceitos básicos. Um grande exemplo é a disseminação das redes com endereços internos, não roteáveis na Internet.

Esses fatos refletem os problemas de evolução nas camadas de rede e de transporte da Internet (protocolos IP e TCP, respectivamente). Acima e abaixo, a evolução tem sido rápida e muitas vezes surpreendente. A camada de aplicação e principalmente os aplicativos têm evoluído extraordinariamente desde a criação da Internet. O correio eletrônico foi criado em 1972, a Web em 1991 e as aplicações *peer-to-peer* (p2p) [50] em 1999. Por outro lado, as tecnologias sub-IP (camadas física e de enlace) têm alcançado altos índices de atualização. A Arpanet original era interligada por enlaces de 56 Kbps. Atualmente, uma única fibra ótica pode comportar vários comprimentos de onda (usando WDM, multiplexação por divisão de comprimento de onda), cada um com capacidade de transmitir até 10 Gbps.

3.1.2. Novos Desafios que Exigem Mudanças

A Internet atual está funcionando há mais de 30 anos e está continuamente recebendo novas missões e ajustes necessários para uma nova categoria de serviços avançados e novas tecnologias de rede. Portanto, novos desafios exigem mudanças e grandes desafios freqüentemente exigem grandes mudanças. Os usuários têm gerado demandas que impulsionam a Internet a se tornar cada vez mais ubíqua, móvel e sem fio. Os sistemas celulares de terceira e quarta geração devem obrigatoriamente ser incorporados à Internet. Certamente na sua concepção original, alta mobilidade não era um requisito relevante, mas que hoje deve ser incorporada de maneira natural a sua arquitetura.

Por outro lado, a Internet começou como uma rede homogênea, mas atualmente isso está cada vez mais distante de ser a norma, mas a exceção. Existem redes com características complemente diferentes da Internet tradicional baseada em TCP/IP, como por exemplo, redes de sensores sem fio e redes com roteamento ad-hoc. A integração dessas redes heterogêneas é um novo desafio, que deve ser considerado por qualquer projeto de alteração na arquitetura da Internet que tenha a pretensão de sobreviver aos novos usos e novas demandas que o futuro possa trazer.

3.2. Princípios da Arquitetura da Internet

Esta seção contempla os princípios básicos que sempre nortearam as decisões de projeto na Internet e que ainda hoje são importantes para a sua compreensão e evolução. Qualquer nova proposta de arquitetura para a Internet deve levar em consideração os sues princípios de modo a não violar principalmente aqueles que são fundamentais. Por outro lado, quando uma proposta revolucionária for apresentada, ela deve saber claramente quais os princípios que está violando.

3.2.1. Existe uma Arquitetura da Internet?

Muitos membros da comunidade da Internet acreditam que não há uma “Arquitetura da Internet” (propriamente dita, de acordo com os conceitos OSI/ISO), mas apenas uma tradição, que não tinha sido redigida até a RFC 1958 [6]. No entanto, normalmente autores assumem que existe uma arquitetura, para poder compará-la com o modelo OSI. Inclusive, é relativamente bem difundido o conceito de que a “a arquitetura da Internet tem quatro camadas”. Se nem os mentores da Internet têm certeza de que existe uma arquitetura, como ela pode ter quatro camadas?

Em termos gerais, no entanto, a comunidade acredita que o principal objetivo é a *conectividade*, a ferramenta é o *protocolo IP* e a inteligência não está dentro da rede, mas nos sistemas finais. O grande crescimento observado nos últimos anos mostra que na verdade a conectividade é o maior resultado da Internet, que tem mais valor do que qualquer aplicação individual, como e-mail ou Web. A chave para a conectividade global é a camada de rede (com o IP). A chave para explorar essa camada sobre diversas tecnologias para prover a conectividade global é o argumento fim a fim (seção 3.2.2). Um sentimento da comunidade é que deveria haver um único protocolo na camada de rede, dada a sua importância. Na Internet, tudo se executa sobre o protocolo IP, que por sua vez, pode executar sobre qualquer tecnologia. Existe, porém, algum interesse para haver mais do que um protocolo, como para permitir a transição do IPv4 para o IPv6.

A evolução da Internet depende de um consenso aproximado sobre propostas técnicas e código executando (e estável). A definição de "consenso aproximado" é dada na RFC 2418 [3], que define o modo de se chegar a um consenso a ser usado pelos grupos de trabalhos da IETF (www.ietf.org). Por outro lado, a Internet sempre funcionou baseada no conceito de que experiência e resultados com implementações reais são mais importantes de que qualquer princípio de arquitetura. O lema da IETF, atribuído a David Clark[1] é: "*We reject presidents, kings and voting; we believe in rough consensus and running code"* ("Nós rejeitamos presidentes, reis e votação; nós acreditamos em consenso aproximado e código executando").

3.2.2. O Argumento Fim a Fim

A tecnologia da Internet é baseada em um princípio fundamental, chamado de argumento fim a fim [27], que determina que toda a inteligência deve ser depositada nos sistemas finais e a rede deve executar tarefas muito simples. Esse argumento foi estabelecido por Salzer, Reed e Clark no início dos anos 1980 e até hoje ele é utilizado como base para todos os tipos de alterações que se deseja fazer na Internet. De fato, em muitos aspectos o argumento fim a fim vem sendo gradualmente violado, devido a mudanças pragmáticas realizadas pelos provedores e fabricantes de equipamentos (seção 3.3.5).

O argumento fim a fim sugere que as funções localizadas nos níveis inferiores de um sistema podem ser redundantes ou de pouco valor, quando comparadas com o custo de implementá-las nesse nível. Em geral, para serem completa e corretamente implementadas, as funções precisam do conhecimento e ajuda dos níveis superiores, que estão localizadas nos pontos finais de um sistema de comunicação. Algumas vezes, versões incompletas da função podem ser implementadas pelo sistema de comunicação para melhorar o desempenho. Esse caso pode ser visto no controle de erros realizado por algumas implementações de camada de enlace de dados. Com alguns meios de transmissão, como comunicação sem fio, é útil controlar os erros, mas a função está incompleta, pois não tem a dimensão fim a fim.

Este princípio tem conseqüências importantes na obtenção de um dos principais objetivos originais na Internet (seção 3.2.4), a sobrevivência a falhas parciais da rede [6]. Um projeto de protocolo fim a fim não deveria contar com a manutenção de estado (ex: informação sobre o estado da comunicação fim a fim, como para o protocolo TCP)

[1] David Clark foi diretor da IAB (*Internet Architecture Board*, comitê especial da IETF, www.iab.org) por vários anos e autor de vários artigos seminais sobre a arquitetura da Internet, como [9][10].

dentro da rede. O estado da comunicação deveria ser mantido apenas pelos sistemas finais, de tal modo a somente poder ser destruído caso o próprio sistema final seja desativado. Este princípio é conhecido como "compartilhamento de destino" (*fate sharing*), porque as várias comunicações fim a fim compartilham do mesmo destino do sistema final, ou seja, se a estação sai do ar abruptamente, todas as conexões são conseqüentemente interrompidas.

O trabalho do núcleo da rede é transmitir pacotes da maneira mais eficiente e flexível possível. Tudo o mais deveria ser feito nas bordas (sistemas finais). Em geral, o único tipo de informação mantido pela rede são as tabelas de roteamento. No entanto, os roteadores aprendem as rotas dinamicamente e a saída de um roteador não deveria causar a falha nas comunicações, a menos que ele fosse um ponto único de passagem de dados.

3.2.3. O Princípio da Mudança Constante

Uma das realizações mais notáveis da Internet não é necessariamente o que ela é capaz de fazer hoje, mas o fato de ter assumido as dimensões atuais, comparada aos seus propósitos iniciais. Ela iniciou com objetivos bem modestos, não foi projetada para ser utilizada por milhões de pessoas no mundo inteiro. Com toda certeza, o conjunto de princípios que balizou o seu aparecimento e que hoje suporta a sua evolução é o grande responsável por isso. Na verdade, esses princípios também não são imutáveis. "O princípio da mudança constante talvez seja o único princípio da Internet que deveria sobreviver indefinidamente" [6]. Essa característica permite que grandes transformações se acomodem naturalmente na estrutura da Internet

Os projetistas da Internet compreenderam desde o início que a rede tinha que ser projetada visando generalidade, a Internet deveria evoluir tão rapidamente quanto a indústria de computadores [42]. A generalidade projetada para a Internet permitiu tanto que ela suportasse aplicações que não foram contempladas no seu projeto original, como também que admitisse novas tecnologias com diferenças drásticas em desempenho e comportamento. A seu projeto permitiu acréscimos em capacidade nos enlaces em muitas ordens de magnitude (ex: de 56Kbps para 10 Gbps), assim como a inclusão de tecnologias que causam certa desordem, como os novos avanços em mobilidade. Este sucesso implica necessariamente que para continuar a evoluir a Internet, os novos projetos devem levar as mudanças em consideração.

Soluções pontuais direcionadas a fazer otimizações puramente transitórias, que trocam melhorias em desempenho pela perda da generalidade deveriam ser evitadas. Esse raciocínio leva a um meta-princípio que tem um grande impacto em muitas decisões técnicas na arquitetura da Internet [12]: "evitar otimizações não inteligentes". Uma forma extensa é: "qualquer projeto envolvendo a Internet deveria favorecer a generalidade, adaptação e evolução, acima da eficiência e até mesmo da funcionalidade".

Um problema mais difícil de enxergar, mas nem por isso menos importante, ocorre quando uma determinada tecnologia se consolida no mercado. Quando isto ocorre, é natural que existam pressões de vários setores da comunidade Internet para barrar as mudanças, por vários motivos, como preservação de investimentos, treinamento de pessoal e manutenção da estabilidade da rede. Nesse momento, deve haver uma ação explícita, incluída na arquitetura, que preserve a capacidade de mudar, evoluir e avançar a tecnologia [42]. Uma conseqüência é que essa ação envolve

sacrifícios em outras dimensões, como desempenho e eficiência. Infelizmente, esse sacrifício freqüentemente resulta em custos maiores a curto prazo, para preservar benefícios menos concretos a longo prazo. O desafio é projetar uma arquitetura que compatibilize esses dois objetivos conflitantes.

Existem, no entanto, fatores que limitam a capacidade de mudanças em uma determinada arquitetura de rede, chamados de invariantes [43]. No caso da Internet, o exemplo mais expressivo de invariante é o endereço IP. Uma mudança simples no formato do endereço não consegue ser facilmente acomodada, mesmo que os outros princípios permaneçam inalterados. A difícil transição para o IPv6 é um exemplo significativo do poder limitante desse invariante.

3.2.4. Objetivos de Projeto da Internet

Compreender o funcionamento da Internet e as decisões que foram tomadas, passa pelo conhecimento dos objetivos levados em consideração para a sua criação. A primeira formalização dos princípios foi feita por David Clark em 1988 [10]. No projeto NewArch [12], Clark *et al.* redefinem estes objetivos, e acrescentam alguns novos objetivos (ou requisitos). Esta seção se baseia na análise realizada em NewArch. Os objetivos primários originais da Internet são:

1. *Multiplexação*: A Internet é baseada em comutação de pacotes e esse objetivo fundamental tem implicações no resto da arquitetura. Por exemplo, o pacote é a unidade utilizada para transmissão de dados pela rede até os sistemas finais e todas as tecnologias devem suportar pacotes. A utilização multiplexada da rede por sistema finais que enviam e recebem apenas pacotes é algo que deveria ser imutável na Internet, mesmo que outras unidades de transmissão pudessem apresentar otimizações. A grande flexibilidade dessa abordagem justifica a sua permanência.
2. *Sobrevivência*: A comunicação na Internet deve continuar, independente da falha de roteadores ou redes inteiras, contando que existe algum caminho possível. Esse requisito foi chamado originalmente de sobrevivência, devido ao contexto militar da época, mas atualmente é mais conhecido como robustez. Este requisito implica que a rede deve se adaptar dinamicamente a falhas de hardware ou software e favorece a utilização de protocolos que são, de alguma forma, “auto curáveis”.
3. *Generalidade de serviços*: A Internet deve permitir que uma grande variedade de tipos de aplicações seja implantada, viabilizada através do suporte de tipos diferentes de serviços na camada de transporte. Dois tipos estão em utilização na Internet atual: um serviço bidirecional de transporte confiável de bytes e um serviço de datagrama que entrega pacotes individuais sem nenhum tipo de garantia. Os dois principais protocolos que implementam esses serviços são TCP e UDP, respectivamente.
4. *Diversidade de tecnologia de sub-rede*: A Internet deve acomodar e interconectar uma grande diversidade de redes que usam características muito diversas. Para isso, a única suposição que o protocolo IP faz sobre a rede subjacente é que ela seja capaz de transmitir os pacotes de um lado para outro. Ser capaz de lidar com tecnologias heterogêneas foi e continua sendo

fundamental para a universalização da Internet, além de permitir atualizações tecnológicas parciais.

Outros objetivos foram identificados originalmente, mas considerados secundários. Outros foram adicionados mais tarde:

5. *Escalabilidade:* A rede deve permitir um crescimento geral consistente, mesmo com a constatação atual de que o número de sistemas finais cresce exponencialmente.
6. *Gerenciamento distribuído*: Basicamente, os recursos e serviços da Internet são gerenciados de maneira distribuída. No início, esse era apenas um objetivo teórico, porque um único provedor tinha a concessão do *backbone* da Arpanet. No entanto, desde que as redes começaram a se multiplicar, o gerenciamento foi ficando cada vez mais distribuído, incluindo serviços que vão desde o roteamento até a resolução de nomes (DNS).
7. *Segurança:* O software e hardware da rede devem suportar funções comuns de segurança, como privacidade, integridade e autenticação. Tanto a infra-estrutura de rede (os roteadores) quanto os sistemas finais devem se preocupar com segurança.
8. *Mobilidade:* A Internet deve suportar que uma estação mude o ponto onde está conectada com a rede dinamicamente. Um problema que vem da mobilidade é que a estação não pode permanecer com o mesmo endereço, para que o roteamento seja escalável (porque as rotas são anunciadas para redes e não para estações individuais).
9. *Alocação de capacidade*: Em geral, existe um consenso de que a capacidade da rede deveria ser alocada de maneira "justa" (o protocolo TCP, por exemplo, tenta fazer isso). No entanto, devido ao contexto militar havia originalmente o requisito de suportar deliberadamente algum nível de injustiça, através dos bits de precedência do cabeçalho do protocolo IP (atualmente chamados de *DSField*). Atualmente, vários provedores gostariam de poder alocar a capacidade de suas redes de maneira injusta, por uma questão de política ou preço, fornecendo assim, algum tipo de Qualidade de Serviço (QoS).

Outros requisitos, embora tenham sido identificados logo no início da Internet, nunca foram colocados na prática explicitamente como objetivos. Por exemplo, a necessidade de a rede ter uma boa relação custo/benefício nunca foi considerada, devido ao contexto militar. Outra questão desconsiderada foi a contabilidade dos recursos utilizados. Esses dois objetivos são atualmente bastante importantes na era da Internet comercial. Uma rede projetada principalmente para uso comercial, certamente trocaria de posição esses objetivos. Com toda certeza, as características da Internet atual se devem principalmente a uma forte influência dos quatro primeiros objetivos.

A conclusão dessa discussão é que qualquer nova proposta para modificar a Internet deveria considerar os objetivos, velhos e novos, para não ter o perigo de perder as características que fizeram o seu sucesso. Um perigo associado é tornar a rede mais complexa do que deveria. A simplicidade da Internet é o tema da seção 3.2.5.

3.2.5. O Princípio da Simplicidade e o Modelo da Ampulheta

O princípio da simplicidade tem sido usado intensivamente no projeto da Internet e foi claramente especificado na RFC 3439 [4]. Ele especifica que a complexidade é o principal mecanismo que impede a escalabilidade e causa o aumento dos custos. Além disso, a simplicidade proporciona robustez, porque soluções complexas em geral são mais suscetíveis a erros. A "espiral da robustez/complexidade/fragilidade" demonstra os efeitos do uso descontrolado da complexidade [44]. Para se obter robustez, muitas vezes a solução mais fácil é adicionar complexidade, que por sua vez introduz novas fragilidades, cuja solução demanda mais complexidade.

Existem alguns conceitos relacionados com o princípio da simplicidade, como o Occam's Razor e o princípio KISS (*Keep It Simple, Stupid*). O primeiro diz que a pluralidade não deveria ser usada sem necessidade, ou seja, a complexidade deve ser controlada. O segundo diz que ao decidir sobre a adoção de uma entre várias soluções que resolvem o mesmo problema, deve-se escolher sempre a mais simples.

Em geral, pode-se observar que os princípios da arquitetura da Internet são inter-relacionados e interdependentes. Por exemplo, o terceiro e quarto objetivos primários da Internet (seção 3.2.4) são a base de uma máxima que ficou famosa na Internet, que diz "IP sobre tudo e tudo sobre IP" [44]. Essa formulação pode também ser considerada uma conseqüência da aplicação do argumento fim a fim e do princípio da simplicidade, principalmente na camada de rede. De fato, a adoção de uma camada de rede minimalista, leva a uma analogia com uma ampulheta, onde o protocolo IP é a sua cintura fina, como mostra a Figura 3-1. Seguindo o argumento fim a fim, pode-se ver que a camada de rede é extremamente simples. A complexidade reside nos sistemas finais, cuja funcionalidade é implementada nas camadas acima do IP.

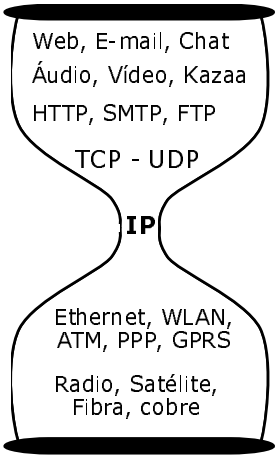


Figura 3-1 – O modelo de ampulheta da Internet

A utilização de um modelo em camadas também favoreceu o surgimento da ampulheta, com um mecanismo genérico de entrega de pacotes do protocolo IP na sua cintura. Ele proporciona uma separação crítica entre uma infra-estrutura física cada vez mais versátil e que avança a passos amplos abaixo da cintura e uma demanda crescente dos usuários por serviços e aplicações de alto nível acima da cintura. O lema "IP sobre tudo e tudo sobre IP" proporciona grande robustez a mudanças nas camadas abaixo e

acima da cintura, mas dificulta a modificação justamente no protocolo IP e também nos protocolos TCP e UDP que estão logo acima dele.

3.3. Principais Questões de Projeto

Essa seção enfoca as principais questões de projeto da Internet que devem ser consideradas no projeto de arquitetura futuras para a Internet. Algumas dessas questões foram violadas no decorrer nos anos ou simplesmente não foram consideradas no projeto original da Internet. As questões tratadas são endereçamento/nomeação, roteamento, segurança, mobilidade, transparência fim a fim e modelo em camadas. Esta categorização foi adotada pelos autores após análise criteriosa de várias propostas de mudanças na Internet. Na bibliografia pesquisada, essas seis questões são citadas com maior freqüência, mas outras poderiam ser incluídas, como Qualidade de Serviço (QoS) e difusão seletiva (*multicast*).

Essas questões, no entanto, não encerram modificações estanques. Pelo contrário, freqüentemente as questões são inter-relacionadas, ou seja, o que ocorre em uma delas afeta outras. Um exemplo interessante é a interação entre transparência e segurança, com relação à utilização de NAT e IPsec. A transparência é quebrada na maioria das redes com a justificativa da segurança. Por outro lado, a quebra da transparência impede que, por sua vez, outras soluções de segurança sejam adotadas.

3.3.1. Endereçamento e Nomeação

O projeto original da Internet excluiu muitos aspectos necessários às demandas de serviços atuais, tais como soluções para mobilidade, multi-endereçamento e segurança. Conforme mencionado anteriormente, as soluções propostas abordam os problemas de maneira isolada e paliativa. Especificamente, a camada IP tem servido como base para essas soluções visto que a maioria propõe sua extensão para suportar novas funcionalidades. Por exemplo, isto torna-se bastante evidente com as propostas do IP Móvel [25] e IPsec [21]. Além disso, sua arquitetura original foi projetada para basicamente prover comunicações *unicast* entre nós fixos.

Uma característica da camada IP é que seu protocolo acopla fortemente endereçamento (localização na rede) e identificação (identidade) em um único atributo, que é o seu endereço IP. Até então esta é uma situação aceitável, visto que na sua maioria os sistemas finais são estacionários, mas torna-se um problema em redes com alta mobilidade. Tais problemas surgem devido ao simples fato de que o endereço IP no papel de identidade do nó, não deveria ser mudado. Por outro lado, uma vez que ele também descreve a localização na rede, o endereço IP necessariamente deveria mudar à medida que o nó muda seu ponto de conexão. Em [49], Moskowitz *et al*. argumentam sobre a impossibilidade de se conseguir tal mudança dinâmica de maneira estável.

Segundo Eriksson *et al.* [14], soluções como DHCP, NAT [29] e IP Móvel sempre acarretam novos problemas e esses paliativos para mobilidade e escalabilidade no protocolo IP tem seus limites. Desta forma, os autores argumentam a impossibilidade de se obter, sob a estrutura atual, uma camada de rede ágil, *plug and play* e escalável. Em [30], Stoica *et al.* argumentam que apesar da maioria destes paliativos atingirem os objetivos desejados, outras funcionalidades não são plenamente suportadas. Por exemplo, as propostas para *multicast* na camada de aplicação não atende aos requisitos de mobilidade e vice-versa. Em [16], Francis e Gummadi expõem alguns motivos da

popularidade do NAT, além de suas vantagens e desvantagens. Um dos principais benefícios do NAT, juntamente com a expansão do espaço de endereçamento do IPv4, é a separação e isolamento dos espaços de endereçamento global e local. Por outro lado, um aspecto negativo citado é que sua implantação em larga escala acarreta numa maior complexidade na construção de novas aplicações e protocolos, visto que é necessário levar em consideração as operações de tradução de endereços ou portas.

Um outro aspecto importante refere-se à estratégia de nomeação na Internet. Atualmente a Internet dispõe de um mecanismo único para resolução de nomes, o *Domain Name System* (DNS), que converte nomes de domínio no nível do usuário em endereços IP. Conforme descrito em [1], a Web requer um serviço de resolução de referências (*Reference Resolution Service* – RRS) para fazer o mapeamento das referências (ex: *Uniform Resource Location* - URL) com as suas respectivas localizações na rede. As referências na Web seguem uma estrutura *hostname/pathname* e o serviço de DNS funciona como um RRS, mapeando o nome do sistema final para um endereço IP, onde um determinado objeto está armazenado.

É importante observar que devido ao enorme crescimento da Web observado na última década, novas funcionalidades tornaram-se necessárias, tais como migração e replicação de conteúdo. Para este fim, provedores de conteúdo têm utilizado recursos adicionais, como por exemplo as Redes de Distribuição de Conteúdo (*Content Distribution Network* - CDN). Em linhas gerais, o objetivo de uma CDN é distribuir geograficamente um conjunto de servidores de conteúdo que mantém cópias de objetos de algum servidor principal. Desta maneira, ao solicitar um objeto referenciado na Web (no servidor original), um cliente pode recebê-lo de um dos servidores da CDN. Além disso, a localização geográfica do servidor que vai fornecer o objeto pode estar mais próxima ao cliente, de tal forma que se evite o tráfego por um longo percurso pelo núcleo da rede. Desta forma, é então possível afirmar que o problema original de migração e replicação de conteúdo na Web não é uma questão resolvida e apenas foi mascarada através de recursos externos como as CDN. A característica intrínseca de uma URL no qual amarra referências a objetos a sistemas finais específicos dificulta a movimentação e replicação de conteúdo de maneira “natural”. Conseqüentemente, como a Web depende fortemente do DNS, restringindo sua flexibilidade, Walfish *et al*. [52] argumentam que ambos sistemas se beneficiariam com uma forte mudança no RRS, de tal forma que houvesse um desacoplamento da Web do DNS.

3.3.2. Roteamento

O serviço de roteamento na Internet foi claramente um dos mais importantes na sua especificação original, pois permitiu que um conjunto de redes heterogêneas pudesse se interconectar. Com sua expansão, seu roteamento evoluiu de uma estrutura plana para uma organização hierárquica, onde alguns roteadores são usados para troca de informações de entre grupos de redes sob mesmo domínio administrativo (Sistema Autônomo - SA), enquanto outros são usados para troca de informações dentro de um mesmo SA. De maneira simplista, a classificação de sua organização pode ser feita da seguinte forma:

a) Roteadores Internos (*Interior Routers*): trocam informações dentro de um mesmo SA e utilizam protocolos de roteamento interno (*Interior Routing*

Protocols). Exemplos destes protocolos incluem o *Routing Information Protocol* (RIP) e o *Open Shortest Path First* (OSPF);

b) Roteadores Externos (*Exterior Routers*): trocam informações entre SA e utilizam protocolos de roteamento externo (*Exterior Routing Protocols*), como por exemplo o *Border Gateway Protocol* (BGP).

Essa estrutura organizacional de roteamento aparentemente é estável e poderia plenamente suportar o crescimento da rede, como aquele observado na última década. Porém, a expansão da rede vem acompanhada pela demanda por novos serviços que implicam em modificações na arquitetura de roteamento.

Neste ponto, vale a pena enfatizar que o termo arquitetura de roteamento não refere-se única e diretamente a protocolos de roteamento (ex: RIP, OSPF ou BGP). Uma arquitetura de roteamento leva em consideração a maneira como a rede trata o tráfego do usuário. Desta forma, uma arquitetura de roteamento consiste basicamente:

a) das técnicas de troca de informação entre seus elementos;

b) dos algoritmos de cômputo e seleção dos caminhos entre as entidades participantes, e;

c) na forma de encaminhamento do tráfego.

Em linhas gerais, a atual arquitetura de roteamento na Internet baseia-se em algoritmos Vetor-Distância[2] (*Distance Vector* – DV) para a troca de informação das tabelas de roteamento e no encaminhamento do tipo salto a salto (*hop-by-hop*). Este modelo tem sido criticado por impor restrições em relação a escalabilidade da rede para acomodar novas funcionalidades [32][8][26][33]. Em [32], Castineyra *et al.* argumentam que uma série de fatores, incluindo avanços tecnológicos e demandas econômicas, determinam a evolução da rede. Desta forma, os autores listam uma série de restrições, características e requisitos que têm implicações diretas nas arquiteturas de roteamento. Dentre as mais importantes, pode-se citar:

a) diferentes tipos de enlaces de comunicação compõem a rede, tais como enlaces fixos, ópticos, sem fio, satélite com diferentes características em termos de atraso, taxas de erro, vazão etc.;

b) seus elementos, tais como redes, roteadores, sistemas finais e processos, podem ser móveis;

c) Usuários podem especificar os requisitos de tráfego que podem variar entre sessões;

d) Provedores de Serviços (ISP) e usuários mantém uma relação intrínseca, de tal forma que à medida que os usuários desenvolvem aplicações mais complexas com requisitos especiais, os provedores de serviços tentam atender estas novas demandas. Reciprocamente, à medida que os provedores implantam novos serviços, os usuários desenvolvem aplicações que os utilizam plenamente. Vale salientar que ISPs são mais resistentes à implantação de protocolos complexos em suas redes, devido suas inerentes dificuldades de gerenciamento.

[2] Apesar dos protocolos OSPF e IS-IS usarem algoritmos de Estado de Enlace, sua implantação é limitada em algumas partes da rede.

Em linhas gerais, estas características apontam que uma arquitetura de roteamento adequada deveria suportar recursos especiais (ex: roteamento específico por serviço ou suporte à mobilidade) usando procedimentos simples que consumam quantidades mínimas de recursos. Especificamente, a arquitetura de roteamento atual da Internet exibe alguns problemas, listados a seguir.

a) No nível de Sistemas Autônomos (domínio), usuários podem escolher seu provedor de serviço (ISP), mas uma vez que seu tráfego entra na rede, eles não têm controle sobre as rotas que os pacotes devem fluir. As rotas selecionadas pelos ISPs são baseadas em decisões econômicas (ex: acordo entre pares) e tecnicamente implementadas sob roteamento BGP. Em [32], Yang considera que tal modelo, onde não se fornece controle do roteamento para o usuário, não promove um nível adequado de competição de mercado. Seu principal argumento baseia-se no fato de que a escolha do roteamento pelo usuário no nível de domínios irá impor uma disciplina econômica no mercado e promover inovação e introdução de novos serviços na rede. Guardadas as devidas diferenças, o processo poderia ser similar ao atual provimento de serviços telefônicos, no qual, o usuário pode escolher o seu provedor de serviços de longa distância, independentemente do provedor de serviços locais.
b) Roteamento e filtragem de pacotes (ex: por *firewall*) são tratados separadamente, apesar da explícita relação entre eles. Não existe um modelo unificado de segurança e roteamento que trate coerentemente ***onde*** existe permissão de fluxo de pacotes (segurança) e ***por onde*** eles devem ser enviados (roteamento) [26].
c) Apesar de não ser um problema de arquitetura e sim específico de protocolo, o BGP tem diversos problemas apontados recentemente. Primeiramente, o crescimento da Internet tem acarretado problemas de escalabilidade [23], instabilidade [28] e convergência [22] nos roteadores com BGP. Além disso, problemas de segurança podem surgir em roteadores mal configurados [19].

Desta forma, algumas soluções propostas requerem mudanças na arquitetura de roteamento atual.

3.3.3. Segurança

A Internet não foi projetada pensando em segurança. As especificações dos protocolos TCP e IP, considerados os mais importantes protocolos da Internet, foram concluídas pelo IETF no início dos anos 80. Desde então, a Internet evoluiu de um projeto especializado conectando alguns milhares de usuários para uma ferramenta de propósito geral e de escopo global que conecta atualmente cerca de 800 milhões de usuários [34].

Embora seu crescimento tenha trazido soluções para várias áreas, ele também é responsável por alguns problemas de segurança. Na verdade muitas das características da Internet que hoje são entendidas como vulnerabilidades já existiam nos primórdios da Internet, mas as ameaças eram poucas e, portanto as vulnerabilidades não eram exploradas na escala atualmente observada. Como a Internet era pequena, os usuários podiam confiar uns nos outros, até porque eles também dependiam uns dos outros. Assim, os problemas de segurança praticamente não existiam. Em geral, uma vulnerabilidade é uma característica (imperfeição ou falha) que pode ser explorada por

um atacante para conduzir um ataque [35]. Um ataque é uma ação ou tentativa de violar uma política de segurança ou causar danos a um sistema.

Considerando a Internet em geral, as vulnerabilidades podem existir em *aplicações* e nos *protocolos de comunicação* e podem ser originadas por falhas de *projeto*, de *implementação* ou de *configuração* [36]. O conjunto de todas essas vulnerabilidades compõe as causas dos problemas de segurança existentes *na* Internet, mas não necessariamente todos são causados por falha dos protocolos usados para dar suporte à rede. Visando distinguir quais das vulnerabilidades podem ser atribuídas a deficiências arquiteturais da Internet, analisamos neste trabalho apenas as vulnerabilidades que são *causadas pela* Internet, deixando de lado aquelas que são exploradas *com o uso da* Internet, tais como: *Ataques de "buffer overflow"* (falha na implementação de aplicações), *Ataque "ping da morte"* e *Ataque de "fragmentação IP"* (falhas na implementação do protocolo IP), entre outros.

Problemas no Protocolo IP

Uma característica chave da Internet é que qualquer aplicação pode mandar qualquer coisa para qualquer um a qualquer momento, sem a necessidade de obter permissão [37]. Por um lado isso é bom por garantir que a Internet é "aberta", ou seja, facilita que novas aplicações sejam projetadas, implementadas e disseminadas rapidamente, sem ter que aguardar que novas características sejam adicionadas à rede. Mas, por outro lado, essa característica pode ser explorada para a condução de um tipo de ataques considerado como dos mais difíceis de combater na Internet: os ataques de Negação de Serviço (*Denial of Service - DoS*) ou Negação de Serviço Distribuída (*Distributed Denial of Service – DDoS*) [38].

Os ataques de DoS podem ser dirigidos aos sistemas finais [39] ou à infra-estrutura de rede, com o intuito de causar a) a parada ou a redução da capacidade de um serviço; b) a exaustão de recursos de CPU; ou c) a exaustão da vazão de enlaces da rede. Um ataque de DoS é distribuído (DDoS) quando um usuário primeiro compromete um estação (conhecida como "zumbi") e a utiliza remotamente para atacar outras estações. Em um DDoS, vários "zumbis" são envolvidos formar um exército e são coordenados para agirem de forma simultânea, enviando pacotes ilegais (ex: *checksum* incorreto, fragmentação inválida) ou mesmo legais (TCP SYN, ICMP *echo request* etc) para uma estação vítima, na tentativa de exaurir os seus recursos e forçá-lo a uma condição de negação de serviço aos seus clientes legítimos.

A Internet, portanto, é vulnerável a ataques de DoS e qualquer estação (ou conjunto de estações) com suficiente largura de banda pode interromper uma conexão ou parar um serviço simplesmente enviando uma inundação de pacotes não solicitados. Ataques de DoS são freqüentes, alcançam escala global e são muito difíceis de rastrear, porque usam a técnica de *spoofing* e dificilmente podem ser filtrados sem atrapalhar o tráfego legítimo [38].

Problemas no Protocolo TCP

O TCP é amplamente implementado e o protocolo fim-a-fim para transferência de dados confiável mais usado na Internet. Quando foi especificado há mais de 20 anos, a Internet, com a conhecemos hoje, era muito diferente e muitas das ameaças hoje conhecidas não eram comuns. Assim, o TCP não incorporou mecanismos de segurança suficientes para lidar com as ameaças a que está sujeito atualmente.

Recentemente algumas sérias ameaças foram descritas, as quais abrem espaço para ataques de DoS e "injeção de dados" [40]. Esses ataques podem ser conduzidos de forma "cega" (*blind attack*) pelo atacante, ou seja, o atacante não precisa necessariamente capturar o tráfego entre as vítimas: ele apenas precisa "adivinhar" algumas informações.

Um ataque ao TCP conhecido como "*blind reset*" [40] pode ser conduzido da seguinte forma. A RFC793 dita que o no recebimento de um RST em uma conexão estabelecida, se o número de seqüência está dentro do intervalo esperado (ou seja, é um segmento aceitável dentro da janela anunciada pelo receptor) então o receptor deve encerrar a conexão. Do contrário, o segmento recebido é descartado. Assim, basta que um atacante adivinhe um número de seqüência compreendido entre o último segmento reconhecido pelo receptor mais a metade do tamanho da janela do receptor, o que não é tão difícil, como discutido em [40].

Um outro ataque similar descrito em [40] é o *blind data injection* (injeção de dados cega), que consiste em submeter um segmento aceitável para o receptor, de forma semelhante ao um *blind reset*, mas desta vez contendo dados ao invés de um RST. Isso faz com que o receptor admita o segmento em sua fila de remontagem e, caso este segmento esteja fora de ordem, ele é repassado para a camada superior depois que o emissor complete o envio dos dados que preenchem o espaço vazio. A inserção de dados espúrios usando esta técnica leva à corrupção dos dados recebidos pela aplicação.

As implicações das vulnerabilidades no TCP são diversas e se estendem aos serviços que dele se utilizam. Por exemplo, o protocolo BGP, usado por roteadores para troca de informações de roteamento entre domínios autônomos da Internet, estabelece conexões TCP de longa duração para o envio confiável de dados. Se uma sessão BGP é interrompida, ela pode causar pequenos intervalos de parada para re-conexão, o que pode ter algum impacto dependendo do tamanho da tabela de rotas e a freqüência do ataque. Embora não tenha sido ainda reportado, cogita-se ser possível injetar rotas falsas através de um ataque "*blind data injection*" em conexões BGP, o que pode ser resultar em um DoS.

3.3.4. Mobilidade

Quando o TCP/IP foi projetado, computadores não eram portáteis como atualmente e, portanto, a mobilidade não foi levada em consideração. Como a necessidade crescente de dar suporte à mobilidade, o IETF criou o "Grupo de Trabalho IP Móvel" (*Mobile IP Working Group* – MIPWG) para estudar soluções para permitir que estações móveis pudessem usar, de forma transparente, o mesmo número IP da sua rede original enquanto se moviam por outras subredes. Os principais requisitos impostos ao MIPWG exigiam uma solução: a) que não modificasse o TCP/IP de estações fixas; b) onde as estações móveis interoperassem com as estações fixas; c) em que a mobilidade fosse transparente e pudessem preservar conexões em andamento; d) escalável; e e) segura.

Em 1996, Perkins *et al.* publicaram a primeira proposta consistente para o MIPv4 na RFC 2002, que foi depois revisada e atualizada pela RFC 3220 e finalmente pela RFC 3344 [45]. Considerando os requisitos, a solução proposta pelo IETF e adotada pela comunidade foi de fato elegante e aceitável. Basicamente, ela exige mudanças apenas na implementação dos protocolos IP e ICMP da "estação móvel" e a criação de dois agentes (tipicamente implementados em roteadores): o *"home agent"* (na rede original da

estação móvel) e o *"foreign agent"* (na rede estrangeira visitada). O IP móvel, portanto, estará disponível apenas nas redes que implementem essas funcionalidades.

Apesar de trazer mobilidade à Internet, alguns problemas ainda não foram completamente resolvidos pelo IP Móvel. O primeiro deles é que o IP móvel usa tunelamento para comunicação entre o *"home agent"* e o novo endereço da estação móvel na rede estrangeira. Para o caminho inverso, a estação móvel normalmente envia pacotes através do roteador da rede estrangeira assumindo o seu IP original como IP de origem. Isso pode causar alguns problemas com *firewalls* [45]. Assim, uma rede estrangeira que estiver filtrando o tráfego de saída (*egress filtering*) poderá não permitir a saída de tráfego que não se origina na sua rede interna, como é o caso do IP original da estação móvel na rede estrangeira. De forma semelhante, se a rede de origem da estação móvel estiver configurada para filtrar o tráfego de entrada (*ingress filtering*) ela não aceitará tráfego de entrada vindo de um endereço interno.

Para solucionar este problema, o IP móvel pode usar a técnica de *tunelamento reverso* visando estabelecer um túnel topologicamente correto entre o novo endereço da estação móvel na rede estrangeira e o *"home agent"*. Esta extensão ao IP móvel é proposta em [46], mas não se propõe a resolver outros os problemas ainda existentes com os *firewalls* ao longo de todo o caminho do túnel.

O tunelamento reverso, apesar de aceitável para o problema dos filtros de entrada e saída, causa outros novos problemas. Um deles é uma ineficiência de roteamento entre a estação móvel e a "estação correspondente", já que força a triangulação do tráfego entre os dois através do *"home agent"*. Ainda, o tunelamento usa cabeçalhos adicionais nos pacotes para os túneis de ida e de volta, o que representa alguma sobrecarga no volume de tráfego.

Outro aspecto relevante é que muitas estações móveis usam algum mecanismo para se manter conectado à VPN da sua redes corporativa. O MIPv4 e as VPNs baseadas em IPsec ainda enfrentam muitos problemas, pois o IPsec requer renegociação quando a "estação móvel" se movimenta entre redes estrangeiras para manter a "estação móvel" conectado a sua respectivas VPN de origem [48].

Durante o movimento da "estação móvel" entre as redes estrangeiras, pode haver períodos de desconexão que levam a uma parada no envio e recepção de dados. Mesmo quando as redes têm cobertura sobreposta, o *handoff* requer a descoberta, negociação e o registro da "estação móvel" na nova rede [45],[47]. A latência observada durante o *handoff* da "estação móvel" causa problemas às aplicações que usam TCP, principalmente porque o emissor reage como se estivesse havendo congestionamento e reduz a taxa de transmissão, muitas vezes recaindo na redução da vazão (TCP *slow start*). Assim, questões já conhecidas do TCP para redes sem fio agora também interferem no MIPv4.

3.3.5. Transparência Fim a Fim

Transparência fim a fim é uma característica da Internet que foi identificada no argumento fim a fim (seção 3.2.2) e nos últimos tempos tem sido gradualmente perdida. O conceito original pode ser definido como [7]:

1. A existência de um único esquema de endereçamento universal para toda a Internet, que permite conectividade fim a fim sem necessitar de dispositivos de interconexão para rede com tipos distintos de endereços.
2. O mecanismo que permite os pacotes trafegarem da sua origem ao seu destino na rede sem serem modificados na sua essência. Isso significa que a rede deve se comportar como uma caixa-preta: o que for colocado nela na sua origem é o que deve sair no destino. Algumas pequenas modificações são aceitas nos pacotes, como o decremento do campo TTL (*Time to Live*), a fragmentação de pacotes e a modificação dos bits de tipo de serviço (campo *DSField*).

A primeira característica foi amplamente explorada na Internet porque determina que os endereços sejam únicos e constantes (duráveis) na rede. Particularmente, os endereços IP foram incorporados nos identificadores de transporte. Por exemplo, uma conexão TCP é identificada por cinco campos: IP de origem/destino, porta de origem/destino e protocolo. Pode-se ver que esse é um caso de violação dos conceitos tradicionais de ocultação das camadas.

A segunda característica foi uma das responsáveis pelo grande sucesso da Internet, porque permite que novas aplicações sejam facilmente implantadas na rede. Isso ocorre porque, se a rede não modifica os pacotes, os sistemas finais são os únicos responsáveis pelo funcionamento da aplicação. Com essa característica, a implantação de uma nova aplicação na Internet requer apenas que ela esteja funcionando nos sistemas finais, enquanto que *nenhuma modificação é necessária no núcleo da rede* (ou seja, nos roteadores) [18].

Uma conseqüência do grande crescimento apresentado pela Internet nos anos 1990 foi que ela perdeu a transparência fim a fim. De um lado, os endereços IPv4 da Internet não mais podem ser considerados nem globalmente únicos nem constantes fim a fim. De outro lado, os pacotes têm freqüentemente seu conteúdo significativamente alterado ao longo do seu trajeto na rede.

As causas da perda da transparência são os dispositivos criados para a alocação criteriosa dos endereços IPv4 e a utilização crescente da tecnologia TCP/IP por empresas e usuários residenciais. Essencialmente, esses dois fatores levaram à introdução de mecanismos chamados de dispositivos intermediários, ou *middleboxes* [5], que estão no caminho dos pacotes IP, mas que realizam funções não convencionais, que não são aquelas funções padrão realizadas por roteadores. A RFC 3234 [5] apresenta vários exemplos de dispositivos intermediários, como: NAT (*Network Address Translation*) [29], *firewall* [17], *proxies* e *caches* [15]. Em geral, o objetivo é utilizar uma menor quantidade de endereços IP, melhorar o desempenho e aumentar a segurança. Especificamente, algumas causas da perda de transparência são:

- O modelo de Intranet: como redes corporativas têm em geral informações confidenciais, segurança é algo fundamental (seção 3.3.3).
- Alocação dinâmica de endereços: o uso crescendo de PPP e DHCP faz com que não se possa vincular um endereço IP a um determinado sistema final.
- *Firewalls*: representam o principal mecanismo para barrar o acesso externo a uma rede privada.
- Endereços privados e NATs: a utilização de endereços privados para aumentar a quantidade de endereços disponíveis, facilitar a troca de provedor e prover

segurança, fazem com que a maioria dos usuários da Internet hoje esteja em redes que usam NAT (ou seja, estão atrás do NAT).

- *Proxies, Caches* e *Gateways* de aplicação: são dispositivos que se interpõem na comunicação entre um cliente e um servidor. Por exemplo, um *proxy* Web abre duas conexões, uma com o cliente e outra com o servidor. Eles podem também analisar, modificar e bloquear algumas mensagens de protocolos aplicação, como gateways de FTP que impedem que certos comandos (ex: put) sejam executados de fora para dentro da rede.

Um grande problema com a perda da transparência é que qualquer aplicação que assuma que endereços são únicos e que não são modificados irá falhar. No caso de protocolos multimídia, como H.323 e SIP, eles usam vários fluxos diferentes simultaneamente, negociando endereços de rede e portas de transporte dinamicamente. Como os endereços não são constantes, é possível que uma máquina atrás de um NAT indique um endereço de rede interna (ex. 192.168.x.x) para o seu par exterior. Obviamente, ele não irá conseguir se conectar a esse endereço.

3.3.6. Modelo em Camadas

Em uma arquitetura tradicional de rede as funções de comunicação são organizadas em níveis aninhados de abstração, chamados de "camadas de protocolos". Este princípio, documentado na RFC 46 [24], diz que é conveniente e útil que a rede seja vista com uma hierarquia de protocolos e de níveis de implementação. A camada N oferece um serviço à camada N+1 e constrói o seu serviço usando os serviços da camada N-1. Além disso, os metadados que controlam a entrega dos pacotes são organizados como "cabeçalhos de protocolos", um para cada camada [20]. O modelo em camadas provê algumas características importantes, como [2]:

1. *Modularidade*: significa quebrar um sistema em partes, para permitir seu desenvolvimento independente, facilidade de substituição e reutilização de componentes.
2. *Encapsulamento*: é uma conseqüência da modularidade, que proporciona a ocultação da informação e independência entre módulos. O encapsulamento provê a abstração, através do agrupamento de mecanismos complexos em um módulo e uma interface simples para a ocultação da complexidade. Em um modelo em camadas, cada camada é encasulada na camada inferior.
3. *Estrutura dos metadados*: os metadados (cabeçalhos) são estruturados em forma de pilha em um modelo em camadas. O último cabeçalho a ser introduzido no transmissor é o primeiro a ser processado no receptor.
4. *Regras de processamento*: os cabeçalhos são processados em ordem rígida, na medida em que são encapsulados no transmissor e removidos no receptor. Isso faz com que o comportamento dos componentes de rede seja previsível, porque cada camada confia na operação correta das outras camadas.

Este modelo de rede em camadas apresentou um bom funcionamento como um princípio de organização, mas ele funcionou melhor quando o argumento fim a fim era seguido mais rigidamente na arquitetura da Internet original. Atualmente, alguns problemas e limitações podem ser observados:

- Existe uma pressão constante pela introdução de violações às camadas. Mesmo a arquitetura base da Internet (os protocolos IP e TCP) inclui uma violação implícita, como no controle de congestionamento, que é detectado pela camada de rede mas exercido pela camada de transporte. Outra violação é a inclusão dos endereços IP nos identificadores do TCP.
- Objeções em alterar implementações estáveis e interfaces antigas entre camadas freqüentemente levam projetistas a inserir novas funcionalidades entre as camadas existentes, gerando uma grande proliferação de subcamadas. Por exemplo, TLS (*Transport Layer Security*) é uma subcamada 4½, IPsec (*IP Security*) é uma subcamada 3½ e MPLS (*Multiprotocol Layer Switching*) é uma subcamada 2½.
- A proliferação dos dispositivos intermediários (*middleboxes*), é uma grande ameaça à arquitetura. Embora sejam introduzidos para resolver outros problemas, eles necessitam de informações de controle que não conseguem serem introduzidos facilmente na pilha de protocolos. O resultado é que são necessários protocolos especiais de sinalização desconectados dos dados (*out of band*), que não é uma característica típica da Internet.
- Nem todos os aspectos podem ser claramente delimitados em uma única camada. Por exemplo, questões de desempenho permeiam 'todas as camadas e deveriam ser tratadas desta forma, ou seja, um pouco em cada camada e de forma integrada. No entanto, o modelo em camadas dificulta a otimização porque a modularidade e a ordem rígida no processamento pode comprometer a eficiência na implementação [9]. Outro aspecto é a segurança, que deve ser vista globalmente, não podendo ser confinada em uma única camada.

A evolução dos protocolos de rede é muito difícil, especialmente na camada de rede, porque muitas funções precisam ser agregadas aos protocolos. Isto leva a uma nova crença, de que talvez camadas não sejam uma abstração suficientemente flexível para modularizar o software de rede [12] e portanto outros modelos seja necessários.

3.4. Propostas de Mudança da Arquitetura da Internet

Esta seção apresenta as propostas que incorporam modificações na arquitetura da Internet, com relação às questões de projeto analisadas na seção 3.3. A escolha das propostas foi feita por dois motivos: pela sua notoriedade acadêmica (citação por outras propostas) e por uma tentativa de abranger as questões de projeto.

3.4.1. Arquitetura de Nomes em Camadas

O sistema de resolução de nomes da Internet, o DNS, usa apenas um nível de indireção. Ou seja, o DNS converte diretamente os nomes de domínio em números IP. O resultado é que, quando um usuário (*browser*) deseja localizar uma informação ou um serviço (ex: http://hotelfazenda.com.br/reservas), o DNS aponta o IP da estação onde a informação reside, ou seja, a informação é endereçada em relação ao local onde reside e ambos são fortemente associados entre si. Se a informação é movida ou replicada para outro lugar, ela não pode ser automaticamente localizada.

A Arquitetura de Nomes em Camadas (*Layered Naming Architecture* – LNA) [1] propõe a separação semântica entre a identificação do serviço (ou informação) da

estação onde ela reside. Esse desacoplamento semântico é alcançado com a introdução de um Identificador de Serviço (*service identifier* – SID) e de um Identificador de Sistema final (*endpoint identifier* – EID). Os dois identificadores, portanto, separam a identificação do serviço das sistemas finais onde eles estão residentes. Nesse esquema, o usuário somente precisa obter o identificador do serviço que deseja acessar (SID) e sua localização (EID) é obtida automaticamente.

A arquitetura LNA, portanto, requer a introdução de níveis adicionais para a resolução de nomes. Agora, a resolução de nomes deve ser executada em três níveis, como mostrado na Figura 3-2. O primeiro envolve converter uma informação do nível do usuário para o identificador do serviço, ou seja, obter o SID. O segundo nível envolve converter o identificador do serviço (SID) para o identificador da sistema final (EID). O terceiro e último nível converte e o EID para o endereço IP.

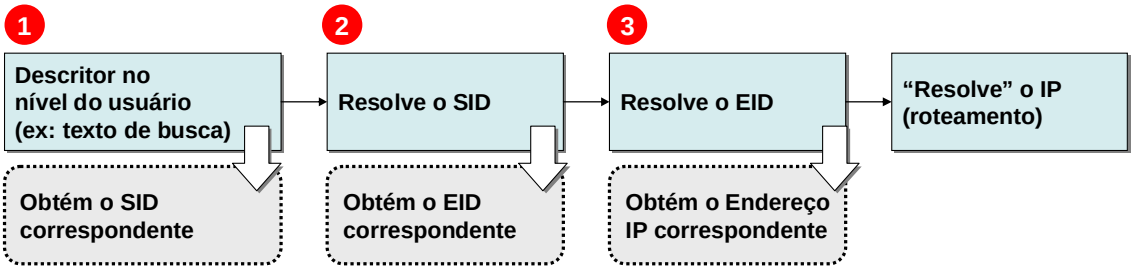


Figura 3-2 – As camadas de resolução da LNA

O primeiro nível, usado para converter uma informação do nível do usuário para um SID pode ser alcançado através de algum serviço de busca. Por exemplo, ao invés de diretórios de busca retornarem URLs (como atualmente) eles retornariam SIDs.

O segundo nível requer o envolvimento da aplicação, tal como atualmente que inclui o "resolvedor". Considere-se uma aplicação ***A*** executando em uma estação ***E*** desejando acessar o serviço ou uma informação representada por um SID ***S***. A aplicação passa ***S*** para a camada de resolução de SID, que contacta a infra-estrutura de resolução e retorna um ou mais triplas do tipo *(EID,transporte,porta)*, cada tripla representando uma instância do serviço desejado, onde *transporte* e *porta* indicam o protocolo de transporte e porta, respectivamente. Por exemplo, duas triplas poderiam ser $(EID_1,TCP,80)$ e $(EID_2,TCP,8080)$. Dependendo do tipo do serviço, informações adicionais podem ser passadas junto com cada tripla. Por exemplo, se ***S*** representa uma página da *web* (e não apenas um servidor *web*), um caminho representando o arquivo pode ser adicionado.

De posse das triplas mencionadas, ***A*** pode se comunicar com o EID especificado usando o protocolo de transporte e porta indicados. Os protocolos de transporte, agora vinculados ao EID (e não mais a endereços IP) poderiam usar o EID da estação de ***E*** como "origem" e o EID de uma das triplas como "destino". Dependendo do tipo de aplicação, ***A*** poderia usar múltiplas triplas para conexões simultâneas por para *backup* em caso de falha da conexão atual. Se todas as triplas falharem, ***A*** pode re-invocar nova resolução para o SID e obter novas triplas.

No terceiro nível, o protocolo de transporte prepara pacotes e os repassa para a camada EID. A camada EID resolve o EID e obtém um ou mais endereços IP (mais de

um quando a estação for *multi-homed* [3] ou quando a estação é uma representação lógica de um conjunto de máquinas). A camada EID adota o IP do emissor como origem e um dos endereços como IP de destino e repassa os pacotes para a camada IP. Se o IP destino não for alcançável, a camada EID pode usar um dos outros IP obtidos no processo de resolução. Se nenhum IP funcionar, a camada re-invoca a resolução do EID para obter novos endereços IP.

Em alguns casos, a entidade destinatária não quer manusear diretamente o tráfego recebido, preferindo direcionar a conexão (delegar) para outra estação de sua escolha. A arquitetura LNA prevê uma forma mais genérica para resolução que permite suportar esse tipo de delegação. Os autores argumentam que este tipo de delegação não altera essencialmente a confiabilidade do relacionamento entre a origem e o destino (se X confia em Y, então também confia nos delegados de Y). Esta delegação pode se dar tanto no nível de SID (serviço) quanto no nível de EID (estação), permitindo a interposição de intermediários no nível de serviços e aplicações (ex: gateways) ou no nível de rede (ex: *firewall*, NAT e VPN).

A arquitetura LNA apresenta vários benefícios. Primeiro, nomear dados e serviços com SIDs, soluciona o problema de usar a URL para tal finalidade e amarrar o serviço ou dado com a estação onde reside. Com o SID as aplicações são nomeadas permanentemente, independentemente da sua localização e estabelece a idéia de que serviços e dados são os objetos de "primeiro nível" na Internet. Segundo, nomear estações com EIDs fornece uma solução natural para mobilidade e *multi-homing*: se uma estação identificada por um EID ***e*** muda seu endereço IP, então a camada de resolução EID re-resolve ***e*** para obter o novo endereço IP. Esta religação automática permite a operação contínua na presença de mobilidade e provê redirecionamento IP em caso de falha em estações *multi-homed*. Por último, permite integrar *middleboxes* como NAT e *firewall* na arquitetura da Internet, sem violar o princípio fim-a-fim ou a semântica do IP.

LNA também traz alguns problemas. O principal deles é que o SID, ao contrário de uma URL, não apresenta uma estrutura hierárquica de domínios. O SID é "plano" (no sentido de que não é hierárquico) e é, na verdade uma seqüência de bits, ou um número. A explicação para isso diz respeito à escolha do mecanismo DHT [51] para conferir eficiência ao processo de resolução.

Como o SID é um identificador que representa o endereço de um serviço ou informação, o usuário pode naturalmente desejar retê-lo, tal como o faz atualmente com uma URL. O SID, entretanto, traz nenhuma informação que permita ao usuário uma associação mnemônica, o que o torna inadequado para manuseio de usuários.

A adoção da arquitetura LNA requer grandes modificações nas camadas de transporte e aplicação, devido à nova infra-estrutura de resolução de nomes, embora praticamente preserve todo o comportamento do IP. Sua implementação não é simples, mas pode se dar de forma incremental e as mudanças nas aplicações de sistemas operacionais podem manter compatibilidade com o mecanismo antigo para permitir transição suave.

---

[3] Uma estação é *multi-homed* quando ela participa simultaneamente de mais de uma sub-rede IP, como é o caso de roteadores e alguns servidores que buscam redundância de comunicação no nível IP

3.4.2. Arquitetura FARA

Conforme apresentado na seção 3.2, o modelo atual de endereçamento na Internet usa o endereço IP como localizador da rede e identificador do sistema final, o que acarreta numa série de problemas previamente identificados. Clark *et al.* apresentaram a Arquitetura FARA (*Forwarding directive, Association, and Rendezvous Architecture*) [11] com o objetivo de tentar aliviar essa sobrecarga do endereço IP. É importante ressaltar que esta arquitetura define um conjunto abstrato de componentes e suas relações, fornecendo um arcabouço consistente e de alto-nível sem especificar detalhes de mecanismos e formatos de seus constituintes. A partir deste arcabouço diversas arquiteturas específicas podem ser derivadas.

Na arquitetura FARA, a comunicação entre os sistemas finais é substituído pela troca de pacotes entre **entidades,** sobre um **substrato de comunicação**. Uma entidade é um conceito abstrato, que pode ser um processo, um *thread*, um computador, um agrupamento de computadores etc. A comunicação entre entidades é feita através de conexões lógicas, puramente fim-a-fim, chamadas de **associações**. Essas associações requerem que as entidades comunicantes mantenham estados persistentes de comunicação. Desta forma, cada pacote pertence a uma única associação e uma entidade pode ter múltiplas associações concorrentes. Em cada pacote existe um identificador que possibilita à entidade receptora a adequada demultiplexação para uma associação particular. Esse identificador é chamado de **identificador de associação** (**AId**) e são estritamente locais a cada entidade. Para o substrato de comunicação**,** a arquitetura assume que este componente possui mecanismos não orientados à conexão para entrega dos pacotes das associações. Isto implica que as entidades são responsáveis pelo tratamento da confiabilidade na comunicação. Ainda, vale a pena ressaltar que a arquitetura não considera a existência de um espaço de nomes global nem para associações nem para entidades.

Quando uma entidade precisa enviar um pacote para uma das suas associações, ele entrega-o para seu substrato de comunicação com um campo de cabeçalho chamado de **diretiva de encaminhamento (*Forwarding Directive* – FD)** de destino, que contém as informações necessárias para o roteamento e entrega do pacote à entidade receptora correspondente. Neste contexto, o componente FD substitui o endereço IP no roteamento de pacotes. A arquitetura também supõe a necessidade de o pacote conter uma FD da fonte, para o caso de haver necessidade de retorno de informações à fonte. Os autores neste ponto relembram que a entrega de pacotes não é para um nó, e sim para a entidade que contém a associação correspondente. É importante então enfatizar que a partir destes conceitos, surgem diferenças substanciais em relação à arquitetura atual da Internet. Especificamente, o endereço IP da arquitetura atual faz o papel da FD e do AId**.** Na arquitetura FARA, essas funções foram deliberadamente separadas e, além disso, ela não requer um espaço de endereços global único. Observe que os elementos destacados em negrito, a saber as **entidades**, as **associações** e seus identificadores **AId**, o **substrato de comunicação** e a **FD**, formam os componentes básicos da arquitetura.

É importante observar as conseqüências da modularidade inerente à FARA, uma vez que o modelo separa os mecanismos de encaminhamento, que acontecem no substrato de comunicação, das funções de comunicação fim-a-fim executadas pelas entidades. Este modelo permite mudanças nos mecanismos de encaminhamento de tal forma que os detalhes da estrutura do substrato de comunicação ficam invisíveis às

entidades, possibilitando que problemas de mobilidade em geral, por exemplo, possam ser plenamente solucionadas.

Na concepção da arquitetura FARA, algumas suposições importantes foram feitas sobre os seus componentes:

- Toda entidade na FARA é móvel e carrega consigo os estados da aplicação e de comunicação. O modelo também permite a mobilidade da entidade quando ela faz parte de um sistema final ou rede que também é móvel;
- Não existe um espaço de nomes global para as associações. O AId é único numa entidade e também local. Assim, esse identificador não muda mesmo se a entidade é móvel;
- Não existe definição de um conjunto global de nomes para as entidades;
- Não é imperativo que os sistemas finais tenham seus endereços escolhidos a partir de um único espaço global de endereços.

Levando em consideração estas suposições, para se estabelecer uma associação entre duas entidades A e B, a entidade A envia uma mensagem para a entidade B, supondo que A possui uma FD para alcançar B. Um problema de inicialização surge neste processo. Como o primeiro pacote irá carregar um AId, se os AId são locais às entidades? Observe que todos os outros pacotes só poderão fluir de A para B uma vez tendo disponível a FD e o AId destino. Para solucionar isto, a FARA adiciona dois novos componentes, a saber o mecanismo *Rendezveous* e o Sistema de Diretório FARA (*FARA Directory System - fDS*).

Mecanismo *Rendezveous*: para criação de uma associação, o primeiro pacote deve ser especial, pois ao invés de carregar o AId destino, ele carrega um *Rendezvous Information String* (**RI**), que a entidade receptora usará para estabelecer uma associação e definir um AId. O mecanismo *Rendezvous* consiste da fase de “descoberta” e da fase de "iniciação". A fase de "descoberta" retorna um par (FD, RI), onde o FD é necessário para a entrega do primeiro pacote à entidade correta, enquanto o RI é usado no destino para criar a associação (na fase “iniciação”).

fDS: a fase da “descoberta” pode ser realizada através de diversos serviços de alto nível, tal como um serviço similar ao DNS. A arquitetura FARA não define especificamente a maneira como o processo de descoberta pode ser realizado, encapsulando essa tarefa num serviço de diretório genérico e deixando os detalhes para a implementação.

3.4.3. Arquitetura NIRA

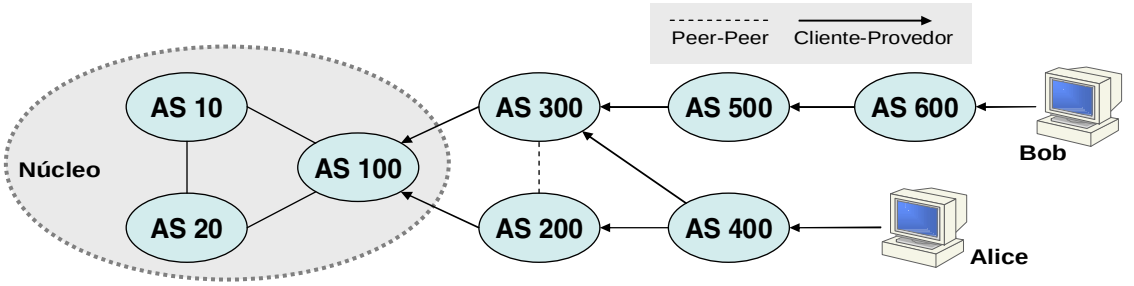
A arquitetura de roteamento NIRA (*New Internet Routing Architecture*) [32] foi projetada para possibilitar ao usuário a escolha de rotas no nível de domínio. Vale a pena enfatizar que o roteamento no nível de domínio refere-se à seqüência de domínio que um pacote atravessa, que é diferente do roteamento no nível de roteador, que trata da seqüência de roteadores.

Ao se propor uma arquitetura de roteamento com estas características, diversos problemas devem ser abordados. O primeiro está relacionado ao descobrimento das rotas pelo usuário. Segundo, entra o aspecto de como estas rotas devem ser representadas. E finalmente, entram os aspectos de ordem econômica com a implantação desta estratégia, de tal forma que fique extremamente claro a maneira que um provedor de serviços poderia ser compensado quando um usuário escolhe usar seus serviços. Conforme

enfatizado por Yang [32], os provedores podem não ter motivação suficiente para encaminhar pacotes de acordo com a especificação do usuário, se não houver uma compensação justa.

O mecanismo de descoberta de rotas em NIRA é dividido entre as duas partes, fonte e destino, de tal forma que cada elemento somente necessita ter conhecimento de sua parte da rede. Cada usuário sabe previamente (ou descobre por algum procedimento) as informações da topologia da rede nos domínios que provêem serviços para ele, de acordo com relações contratuais. A fonte também busca, sob demanda, as informações da topologia da rede do provedor de serviço de destino. Por fim, há a combinação destas duas informações para especificar uma rota que alcance o destino desejado.

Com o objetivo de obter uma solução elegante, a arquitetura NIRA identifica algumas questões essenciais de projeto. Os requisitos considerados foram escalabilidade, robustez, eficiência, heterogeneidade das escolhas do usuário e compensação para os provedores de serviços. Assim, para atender estes requisitos de projeto e solucionar alguns dos problemas previamente apresentados na seção 3.2, a arquitetura NIRA apresenta algumas soluções para o modelo de rede, endereçamento, a representação de rotas, a descoberta de rotas e compensação para o provedor, que são discutidas a seguir.



	AS 100	AS 200	AS 300	AS 400	AS 500	AS 600	Alice	Bob
AllocPf	ae80::/16	ae80:1::/32	ae80:2::/32	ae80:1:1::/48 ae80:2:1::/48	ae80:2:2::/48	Ae80:2:2:2:/64		
InterAddr (Addr)	ae80::/96		ae80:2::/96	ae80:1:1::/96 ae80:2:1::/96	ae80:2:2::/96	Ae80:2:2:2::/96	Ae80:1:1::ec Ae80:2:1:ec	Ae80:2:2:2::6c1a

Figura 3-3 - Modelo de Rede da Arquitetura NIRA

Modelo de Rede: o método de representação de rotas e endereçamento da arquitetura NIRA baseia-se na estrutura da Internet no nível de políticas. Em outras palavras, um domínio decidirá se irá ou não prover serviços de trânsito para os domínios adjacentes, baseado nas suas relações de negócio com eles. Outras definições são necessárias para uma melhor compreensão desta arquitetura. Primeiro, uma rota típica no nível de domínio é rotulada como *valley-free*, isto é, quando um pacote enviado por um usuário é primeiramente "empurrado" em direção à estrutura de seu provedor, fluindo depois em direção à cadeia do provedor de destino. Adicionalmente, a região da rede onde pacotes não podem ser "empurrados" é chamada do núcleo da Internet. Ainda, enlaces em níveis inferiores (*low-level peering link)* podem conectar as cadeias dos provedores da fonte e destino, onde os pacotes poderiam utilizar este atalho. A Figura 3-3 reproduz o diagrama do modelo de rede da arquitetura NIRA, originalmente apresentada em [32]. Os provedores no núcleo estão indicados como AS 10, AS 20 e AS 100. As rotas 400-200-100-300-500-600 e 400-200-300-500-600 são rotas *valley-free.*

Endereçamento: NIRA usa um esquema de endereçamento hierárquico por provedor para reduzir a sobrecarga da construção e representação de rotas. Desta forma, para cada provedor num nível hierárquico superior seria alocado um prefixo de endereçamento globalmente único. Este por sua vez alocaria prefixos de endereços para seus clientes a partir do seu espaço de endereçamento. Recursivamente, esses clientes poderiam também fazer o mesmo com seus respectivos clientes. A proposta original do NIRA assume endereços de tamanho fixo de 128 bits. Um endereço então seria uma concatenação dos endereços inter e intra-domínio. A parte inter-domínio de um endereço de nós no mesmo domínio tem o mesmo tamanho. Desta forma, é possível um nó manter um único endereço intra-domínio. Apresentamos na Figura 3-3 um exemplo do esquema de endereçamento da arquitetura NIRA, utilizando a convenção para representação de endereços IPv6. O comprimento em bits do prefixo do endereço ou de um endereço inter-domínio é especificado após uma barra, "/". Neste exemplo, o provedor no nível mais superior (AS 100) tem como prefixo alocado *ae80::/16* (AllocPf) e tem seu endereço inter-domínio *ae80::/96* (InterAddr). O AS 100 aloca o prefixo *ae80:1::/32* para seu cliente AS 200 e o prefixo *ae80:2::/32* para o AS 300. O processo segue recursivamente para os AS 400, AS 500 e AS 600. Observe que o AS 400 tem dois segmentos de rota para o núcleo (400-200-100 e 400-300-100) e portando tem dois prefixos *ae80:1:1::/48* e *ae80:2:1::/48*. Alice é um sistema final no AS 400 com endereços *ae80:1:1::ec* e *ae80:2:1::ec*, enquanto Bob que está no AS 600 tem o endereço *ae80:2:2:2::6c1a*.

Representação de Rotas: Para o esquema de representação de rotas, NIRA baseia-se no prefixo do endereço que identifica um segmento de rota (rota parcial). Em alguns casos, um par de endereços (fonte e destino) pode representar rotas *valley-free*. Cada uma destas rotas consiste de dois segmentos. Um segmento de rota é a cadeia dos provedores que aloca o endereço fonte, enquanto o outro é a cadeia que aloca o endereço de destino. Observe que os dois segmentos ou alcançam um provedor em comum ou o núcleo da Internet. Por exemplo, novamente na Figura 3-3 o par de endereços *ae80:1:1::ec* e *ae80:2:2:2::6c1a* identificam a rota no nível de domínio entre os sistemas finais Alice e Bob, a saber a rota canônica 400-200-100-300-500-600. As outras possíveis rotas são chamadas de não-canônicas como, por exemplo, a rota 400-200-300-500-600.

Para encaminhar um pacote usando o esquema de representação de rotas, o algoritmo de encaminhamento precisa olhar ambos os endereços (fonte e destino). Assim, ao observar o endereço de destino o roteador será capaz de saber se o domínio de destino já foi alcançado. Se não, o roteador decide se o "ponto de retorno" foi atingido ou não, ao verificar também o endereço da fonte. Se os dois endereços compartilham de um prefixo comum pertencente ao espaço de endereçamento do domínio atual, então o "ponto de retorno" foi alcançado. Ainda, se os dois endereços não compartilham de um prefixo comum, mas o domínio atual é o Núcleo, então o "ponto de retorno" foi alcançado. Antes de atingir o "ponto de retorno", os pacotes são encaminhados "para cima" de acordo com endereço da fonte. Após o "ponto de retorno", os pacotes são encaminhados "para baixo" de acordo com o endereço de destino. Na NIRA, esse mecanismo de encaminhamento é chamado de Encaminhamento *Valley-Free*. Com a rota canônica 400-200-100-300-500-600, o "ponto de retorno" está no AS 100 ou no AS 300.

Descoberta de Rotas: NIRA oferece dois serviços para auxiliar na descoberta de rotas, a saber o Protocolo de Propagação de Informação de Topologia (*Topology Information Propagation Protocol* - TIPP) e o Serviço de Resolução Nome-para-Rota (*Name-to-Route Resolution Service* - NRRS). O objetivo do TIPP é facilitar a descoberta de informações de topologia nos domínios que fornece serviços para o sistema final. Normalmente, o sistema final pode utilizar este serviço para encontrar os segmentos de rota que alcançam o núcleo. O TIPP propaga para um sistema final seus endereços inter-domínio e os segmentos de rotas associadas com estes endereços. O NRRS ajuda um sistema final a solucionar o problema de inicialização (como enviar o primeiro pacote para um outro sistema final). Para tanto, o serviço NRRS assume que um sistema final sabe o nome de seu correspondente e retorna as informações de topologia de seus segmentos de rota. O NRSS foi projetado com um serviço distribuído de busca de nomes. Um sistema final deve armazenar seus segmentos de rota em um servidor pré-definido (servidores de rota). Esses servidores de rota são organizados hierarquicamente em um espaço de nomes. Similarmente à infra-estrutura do serviço DNS, em um NRRS o resolvedor contém uma lista pré-definida com os segmentos de rota do NRRS raiz. O sistema final tem uma lista pré-definida dos segmentos de rotas de seu resolvedor. Em cada nível de resolução, são retornados os segmentos de rota dos servidores de rota que são responsáveis do espaço de nomes no nível inferior. O processo de busca pára quando se encontram os segmentos de rotas relacionados com o nome requisitado.

Compensação para o Provedor: a arquitetura prevê dois modelos de compensação. Ambos requerem que os usuários tenham acordos contratuais previamente definidos com os provedores antes da utilização dos serviços. O primeiro modelo, chamado de Relações Diretas de Negócio (*Direct Business Relationships* – DBR) seria similar ao modelo atual da Internet, onde os acordos contratuais são negociados diretamente entre as entidades conectadas, porém considerando o custo de permitir ao usuário de escolher diferentes rotas. No segundo modelo, chamado de Relações Indiretas de Negócio (*Indirect Business Relationships* – IBR), um usuário poderia negociar com provedores de serviços não diretamente conectados a ele.

3.4.4. Arquitetura IPNL

A arquitetura IPNL (*IP next layer*) [16] propõe uma extensão à atual arquitetura da Internet para incorporar o mecanismo NAT de maneira natural. Seu principal benefício é oferecer o isolamento de redes, de modo a não obrigar renumeração em caso de troca de provedor, além de permitir que redes se conectem a diversos provedores sem necessitar a troca de rotas BGP. Para isso modificações são necessárias somente nos NATs e sistemas finais atuais, permitindo que os roteadores permanecem inalterados. Esta proposta retira a semântica fim a fim dos endereços IPv4 atuais criando uma nova camada, chamada IPNL, entre as camadas de rede e de transporte. Esta mudança no endereçamento faz com que seja necessário alterar também o roteamento, que em IPNL pode ser baseado em endereços IPNL e em FQDNs (*fully qualified domain names* - nomes de domínio totalmente qualificados, como www.gprt.ufpe.br). Uma característica fundamental de IPNL na questão de transparência é permitir que estações atrás de NAT sejam acessadas de fora, mesmo com o isolamento da rede.

A Figura 3-4(a) mostra alguns elementos da arquitetura IPNL. A topologia IPNL é a mesma que a da Internet atual: regiões com endereços privados, conectadas a um

núcleo da Internet com endereços globais, através de NAT. Os NATs são chamados de *nl-router*, a região de endereçamento global da Internet é chamada de *região central* e regiões com endereços privados são chamadas de *regiões privadas*. Um *nl-router* que conecta uma região privada à região central é chamado de *frontdoor nl-router*, ou simplesmente *frontdoor*. Um *nl-router* que conecta duas regiões privadas é chamado de *nl-router interno*.

Para os roteadores IP em cada região, um *nl-router* parece com um sistema final normal. Para os *nl-router*s, uma região parece como uma rede de acesso múltiplo (ex: LAN), ou seja, para a camada IPNL a camada IP se comporta como uma camada de enlace de dados. Somente o endereço IPNL tem uma semântica fim a fim, de modo que quando um pacote atravessa a rede de origem a destino, ele pode incorporar vários endereços IPs temporários, um para cada região. Isso é idêntico ao comportamento dos endereços de enlace atualmente. Os endereços IP para uma determinada região não têm significado em outras regiões. Na Internet atual a situação é diferente, porque os endereços IP globais têm significado nas regiões internas, enquanto que o contrário não é verdadeiro. Por fim, a Figura 3-4(b) mostra o endereço estendido e a Figura 3-4(c) mostra a pilha de protocolos com a camada IPNL.

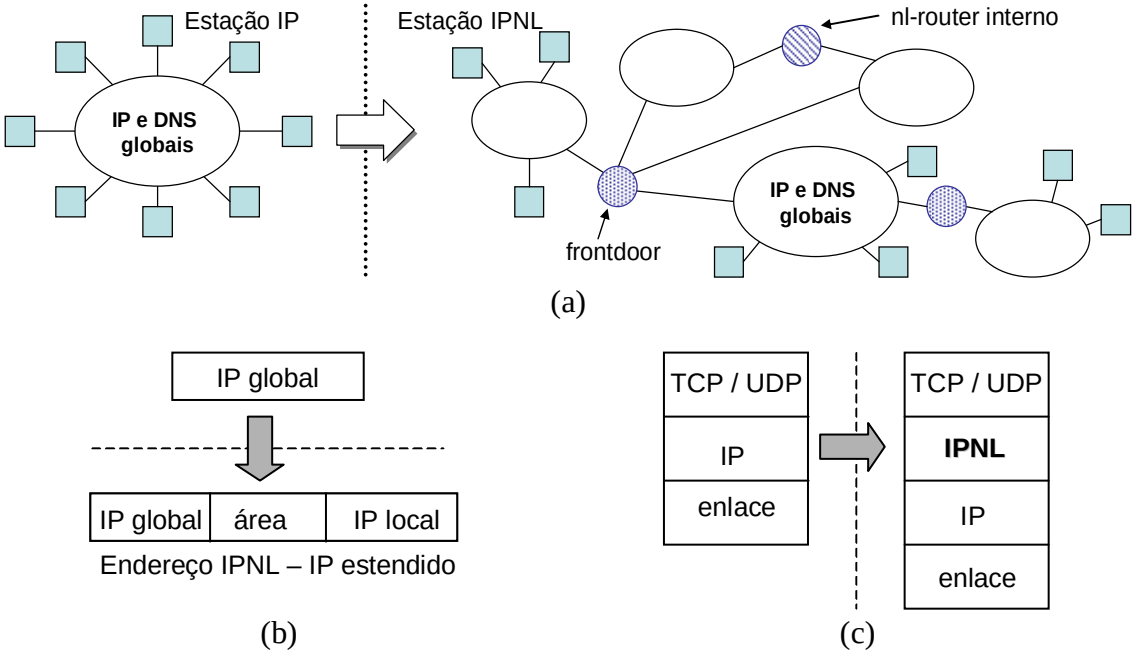


Figura 3-4 – Arquitetura IPNL; a) topologia; b) endereçamento estendido; c) nova camada na pilha de protocolos

Os cabeçalhos IPNL podem carregar dois tipos de endereços para roteamento. Um é o FQDN do sistema final e o outro é o endereço IPNL. Um pacote pode ter somente um dos dois ou ambos em conjunto. O FQDN é o identificador primário de uma conexão fim a fim, que deve ser estável enquanto a conexão durar. No entanto, durante uma conexão, um FQDN de um sistema final pode ser mapeado para vários endereços IPNL sem prejudicar a semântica da conexão. Um conceito importante é justamente a possibilidade de se fazer um roteamento pelo FQDN. Isto é possível devido a uma alteração no DNS. Conforme visto na seção 3.3.1, um endereço IP tem atualmente a função dupla de localizador e identificador do nó. Em IPNL, essa sobrecarga de funções

é exercida pelo FQDN. Uma vantagem dessa abordagem é que ela é robusta a seqüestro de pacotes, uma vez que o algoritmo de roteamento sempre entrega os pacotes ao destino através do seu localizador. Se ele também é a identidade do nó, então existe uma segurança de que os pacotes estão indo para o lugar certo (desconsiderando a possibilidade de ataques do tipo *man-in-the-middle)*.

3.4.5. Arquitetura RBA

A arquitetura baseada em papéis (*role-based architecture* - RBA) [2] rompe com o modelo em camadas tradicionalmente usado na Internet (seção 3.5.6). RBA não é baseada em pilhas de protocolos, mas a comunicação é organizada em unidades funcionais chamadas de *papéis*. Papéis não são organizados hierarquicamente, de tal modo que eles podem ser interconectados de maneira mais ampla do que camadas de protocolos tradicionais. Um papel é uma entidade abstrata e fornece uma descrição funcional de um elemento fundamental de comunicação, que realiza uma função específica no encaminhamento e/ou processamento dos pacotes.

Em uma abordagem sem camadas, as violações de camadas deveriam ser substituídas por interações entre papéis explicitamente projetadas. O objetivo é fazer com que papéis sejam os elementos fundamentais da arquitetura, de preferência bem definidos e conhecidos. Para permitir a interoperabilidade, uma rede real usando RBA não precisaria de muitos papéis bem conhecidos (dezenas ou centenas), mas que fossem bem definidos e padronizados.

Uma vantagem de RBA é permitir que sistemas intermediários se integrem naturalmente na arquitetura, o que não ocorre no modelo em camadas atual. RBA permite que um sistema final sinalize de maneira robusta e extensível para um sistema intermediário que ele deseja ativar ou desativar uma determinada funcionalidade, como por exemplo, permitir que uma estação externa o contacte diretamente ou impedir que uma página seja redirecionada para um cache de web. O modelo em camadas não permite que essas funções sejam integradas facilmente na arquitetura.

Uma arquitetura sem camadas apresenta alguns problemas adicionais que devem ser tratados adequadamente, com relação à estrutura de metadados, regras de processamento e encapsulamento. A estrutura dos metadados no cabeçalho dos pacotes não forma mais uma “pilha” (*stack*), mas um “monte” (*heap*) de cabeçalhos de protocolos. Ou seja, o cabeçalho dos pacotes contém blocos de metadados de tamanho variável que podem ser inseridos, acessados, modificados e removidos em qualquer ordem pelas unidades de protocolo. Esta questão, por sinal, é importante, pois o modelo em camadas especifica uma ordem rígida no processamento dos cabeçalhos. Em um modelo sem camadas não existe uma ordem definida, de modo que os vários protocolos podem inclusive processar o cabeçalho simultaneamente. Obviamente, algum tipo de organização é necessário entre os papéis, dependendo da semântica de cada um deles. Um paradigma que se quebra com RBA é o do encapsulamento, que deixa de ser uma regra universal para ser utilizado somente quando existe uma necessidade semântica forte entre os protocolos. No entanto, esse encapsulamento não é idêntico ao modelo em camadas, pois os cabeçalhos são visíveis a todos os papéis. Portanto, regras mais rígidas para ocultação da informação devem ser criadas, sempre que necessárias.

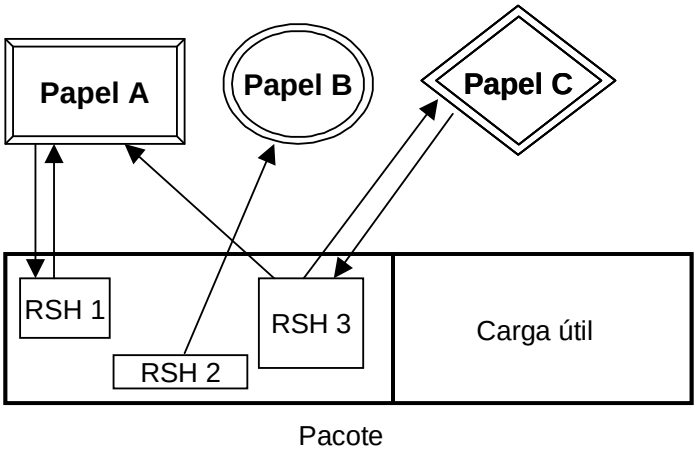


Figura 3-5 – Papéis e cabeçalhos (RSH) em RBA

A Figura 3-5 mostra um pacote em RDA e os papéis que processam os seus metadados no cabeçalho, que por sua vez são divididos em pedaços chamados de cabeçalhos específicos dos papéis (*role-specific headers* – RSH). Na figura, três papéis processam (lêem e escrevem) três RSHs distintos, sendo que um mesmo papel pode processar mais de um RSH e um RSH pode ser processado por mais de um papel. Em RBA um sistema final não sabe *a priori* se o pacote que ele envia irá encontrar um nó intermediário que desempenha um determinado papel. Por exemplo, se o pacote tiver um RSH específico que determina que ele não deve ser redirecionado a um cache, então caso o nó redirecionador consulte o RSH ele não deverá realizar esta função, mas deixar o pacote passar sem nenhuma alteração de destino. Além disso, qualquer nó no caminho pode adicionar um novo RSH ao pacote.

Uma vez conhecidos o mecanismo geral de funcionamento da arquitetura RBA, pode-se estabelecer claramente os seus objetivos:

1. *Expansão*: RBA é inerentemente capaz de ser estendida, tanto do ponto de vista abstrato como de implementação.
2. *Portabilidade*: os papéis não devem estar vinculados aos nós da rede onde serão executados;
3. *Sistemas intermediários*: a arquitetura permite que os sistemas finais se comuniquem explicitamente com os sistemas finais e estes uns com os outros.
4. *Acesso a metadados*: uma vez que não existe uma ordem clara de processamento, RBA procura organizar o modo como os metadados são acessados, para evitar inconsistências.
5. *Auditoria*: um sistema final pode averiguar se o processamento solicitado foi realmente efetuado, através da análise dos RSHs.

3.4.6. Arquitetura Plutarch

Plutarch [13] é um novo arcabouço para redes de próxima geração, que difere da atual arquitetura da Internet principalmente no sentido de que ele adota a idéia da heterogeneidade para atingir inovações revolucionárias. A arquitetura homogênea atual e suas vantagens não são abandonadas, mas mantidas como uma das possíveis arquiteturas entre muitas outras. A proposta de Plutarch é que novas arquiteturas de rede devem se

concentrar em mecanismos que permitam a interoperação entre várias redes heterogêneas, em vez de exigir um conjunto de protocolos único para todos os casos.

Existem algumas motivações para a proposta de um arcabouço como Plutarch. O motivo principal é o problema concreto de conectar redes onde um único protocolo (IP) é impossível ou mesmo indesejável, como no caso de redes de sensores. O segundo motivo é que o modelo abstrato em que Plutarch se baseia consegue capturar o estado da Internet atual melhor do que os modelos baseados nos princípios originais da Internet (seção 3.2.4). Finalmente, um modelo baseado em contextos explícitos oferece um arcabouço mais claro para se debater mudanças futuras na arquitetura do que a tradição atual da Internet pode permitir (devido, por exemplo, aos invariantes, como o endereço IP, no caso da Internet – ver seção 3.2.3).

Conforme mencionado anteriormente, Plutarch divide o mundo em *contextos*, cada um contendo um conjunto de computadores, roteadores, comutadores (*switches*) e enlaces de rede entre outros. Dentro de cada contexto a homogeneidade é esperada, por exemplo, entre endereços, formatos de pacotes, protocolos de transporte e serviços de nomes. Contextos distintos apresentam diferenças em pelo menos uma dessas regiões. A comunicação entre contextos é possível através de *funções intersticiais*[4] que fazem o mapeamento entre o conjunto de funcionalidades de cada contexto. Essas funcionalidades podem ser divididas em quatro áreas principais:

1. *Endereçamento*: O mapeamento entre contextos diferentes de endereçamento é uma função bem conhecida (ex: uso de NAT). As APIs para essas funções deveriam ser claramente expostas, para permitir que os mapeamentos sejam configurados, mantidos e gerenciados de modo automático.
2. *Nomeação*: Atualmente o DNS oferece um único espaço de nomes, com um gerenciamento hierárquico por delegação. Plutarch assume que novos serviços, como VoIP, irão possibilitar o surgimento de abordagens alternativas de mapeamento entre diversos sistemas de nomes, por razões como escalabilidade ou sobrecarga administrativa.
3. *Roteamento*: Estilos diferentes de protocolos de roteamento são apropriados em redes diferentes. Por exemplo, a conexão de uma rede ad-hoc sem fio com roteamento sob demanda com um Sistema Autônomo da Internet que usa OSPF e BGP requer um mapeamento mais complexo do que simplesmente as redes trocarem rotas via BGP (que em geral não é apropriado para a rede ad-hoc).
4. *Transporte*: Um único protocolo de transporte é usado para tratar de todas as tecnologias de rede. Um exemplo disso são as implementações específicas do TCP para redes sem fio, que freqüentemente são baseadas em sistemas intermediários (*proxies*). No entanto, em alguns casos seria vantajoso otimizar protocolos de transporte para redes específicas, ao custo de ter que tratar cuidadosamente caso a caso.

Em cada área, algumas funções intersticiais são necessárias, mas o conjunto de tais funções não é limitado *a priori* por Plutarch. O mais importante é que as áreas (nomeação, endereçamento, roteamento, transporte) devem ser suportadas fim a fim entre redes radicalmente heterogêneas, através da inclusão de interações explícitas nas

[4] Da biologia, “entre tecidos”.

bordas das redes. Através dessas transições explícitas os projetistas de Plutarch que dois benefícios serão alcançados: 1) o modelo da rede irá refletir mais precisamente a realidade da rede; 2) o modelo da rede será mais extensível, permitindo que novos serviços sejam incorporados em todas as camadas da arquitetura[5].

Plutarch prevê que as funções gêmeas de endereçamento e nomeação deveriam ser implementadas de acordo com o argumento fim a fim. Porém, a Internet atual impõe mecanismos localizados no meio da rede (ou seja, não são puramente fim a fim) para endereçamento (no caso do Ipv4, por projeto) e nomeação (o DNS, por um acidente de evolução). A conseqüência disso é que nomes e endereços devem ser globais para funcionar corretamente. Conforme discutido anteriormente, esse modelo é insuficiente para capturar a essência da Internet atual (com sistemas intermediários) e provavelmente também não é apropriado para conectar redes completamente diferentes.

Em vez de impor endereços globais pelo uso do IPv6 (o IPv4 já se mostrou incapaz disso), Plutarch propõe um esquema onde nem nomes nem endereços têm uma semântica global. Isto gera flexibilidade nos sistemas finais retirando a imposição da homogeneidade (unicidade do protocolo IP na cintura da ampulheta) e movendo as decisões de endereçamento e nomeação para os sistemas finais. Esse modelo apresenta duas características que o tornam convincente. Primeiro, ele captura de maneira organizada a realidade da Internet atual, em particular o uso de sistemas intermediários. Isso porque redes que dependem da intermediação de sistemas intermediários podem ser modeladas como contextos distintos, com funções intersticiais bem definidas entre elas. Em vez dos sistemas intermediários violarem um princípio básico da arquitetura, a sua própria existência é incorporada por ela. Em segundo lugar, ele permite que tecnologias futuras sejam incorporadas facilmente, sem a necessidade de aplicar a pilha de protocolos inteira da Internet, permitindo que a Internet atual conviva elegantemente com elas.

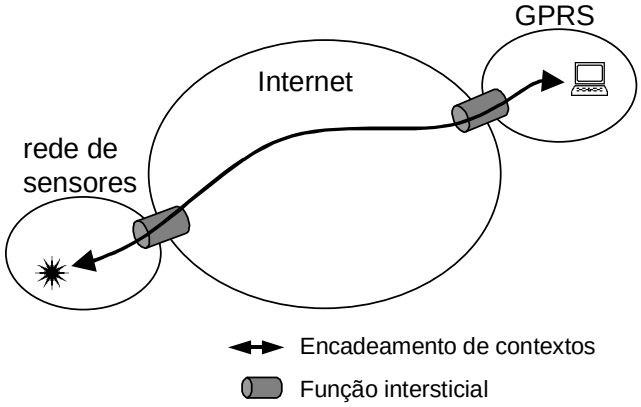


Figura 3-6 – Conexão entre contextos em Plutarch

A Figura 3-6 mostra um exemplo da de uso de Plutarch que poderia facilmente ocorrer na Internet atual. Um usuário com um computador portátil em uma rede GPRS tenta acessar a sua rede de sensores (de pesquisa) através da Internet. Numa condição normal, o usuário não teria acesso à rede de sensores, por eles não implementarem a pilha

[5] É interessante notar que o atual modelo de ampulheta da Internet não suporta facilmente mudanças nas camadas de rede e de transporte.

TCP/IP, por exemplo. Neste caso, uma conexão fim a fim entre o usuário e a rede de sensores poderia ser feita em Plutarch através de três estágios. No primeiro estágio, os nomes para cada contexto (três, no caso) têm que ser resolvidos individualmente. No segundo estágio, os endereços dos três contextos devem ser mapeados em cadeia, através de funções intersticiais e os contextos devem ser explicitamente adicionadas à lista de contextos do computador do usuário. No terceiro estágio, a comunicação ocorre entre as aplicações através das associações entre os contextos. Informações adicionais sobre este e outros exemplos da aplicação de Plutarch podem ser obtidas em [13].

3.4.7. Infra-estrutura SFR

Conforme descrito anteriormente (seção 3.3.1), a forte relação da Web com o serviço DNS tem engessado sua flexibilidade em termos de migração e replicação de conteúdo. Qualquer proposição para quebra desta relação implica na implantação de uma novo serviço de resolução de referências (*Reference Resolution Service* – RRS) em substituição (ou auxiliar) ao DNS. Os requisitos para um novo RRS devem prioritariamente preencher a lacuna deixada pela tecnologia atual (URL baseada no DNS), a saber "referência persistente a objetos" e "referência livre de disputa". O primeiro requisito significa que as referências não devem estar atreladas a nenhum domínio. Vejamos como exemplo o seguinte caso. Suponha que uma página pessoal (objeto) esteja hospedada em um provedor *aaa.com.br*, e que posteriormente migra para o provedor *bbb.com.br*. Com o RRS atual (i.e., DNS) o registro de referência ao objeto é controlado pelo primeiro provedor e a manutenção da persistência implica na permissão (improvável) para a atualização do registro, mesmo que o autor não esteja mais filiado a ele (ex: aaa sendo UOL, bbb sendo TERRA). O segundo requisito está relacionado com a questão legal de propriedade de nomes na Web, que atualmente acontece com registros no DNS.

Em [52] os autores apresentam uma proposta para um RRS com referências sem semântica (*Semantic Free References* – SFR). O SFR é um RRS de propósito geral para referências persistentes e livre de disputa, baseado nos seguintes princípios:

- *Espaço de nomes sem semântica*: em outras palavras, referências não deveriam conter informações sobre instituições, domínios ou provedores onde elas estão localizadas, ou até mesmo serem legíveis ao usuário;
- *RRS com interface mínima*: os serviços oferecidos pelo RRS deveria ser restrito a apenas a resolução de referências. O mapeamento entre nomes legíveis ao usuário e a respectiva referência deve ser feita por sistemas auxiliares;

SFR permite funcionalidades não presentes nativamente na Web hoje, tais como migração de objetos sem a necessidade de atualização dos apontadores ou o aparecimento de apontadores quebrados. Além disso, o processo de replicação de objetos é facilitado sem a necessidade de recorrer a CDN etc.

Desvencilhar-se de um serviço largamente utilizado na Internet, como a Web sobre o DNS, é uma tarefa árdua, por mais que se apresente inúmeras vantagens de uma nova proposição. Desta forma, para uma proposta ser realizável, ela deveria contemplar ou herdar as principais vantagens do sistema legado. No caso do SFR, esta proposta tem que lidar com o fato que a a estrutura hierárquica do DNS fornece unicidade às URLs e ainda possibilita acesso local aos objetos do servidor Web no caso de falhas na conexão

com a Internet (considerando que o servidor DNS está localmente disponível antes do enlace de acesso). Além disso, em alguns casos, a facilidade de leitura dos nomes dos sistemas finais dá ao usuário uma certa confiança nos objetos que ele busca (ex: objetos na página www.ufpe.br). Assim, uma vez que o RRS com SFR não é hierarquicamente organizado nem possui uma interface "legível" para o usuário, esses problemas tem que ser resolvidos com cuidado.

Em relação a escalabilidade, o SFR deve oferecer um esquema de busca rápido e eficiente. Porém a busca de referências em um espaço de nomes sem semântica não é escalável, mesmo com a utilização de DHT, a latência ainda pode ser muito grande. Um outro desafio relacionado refere-se à integridade da referência que deve prevenir que dois objetos distintos recebam a mesma referência.

A proposta de um RRS com SFR é prover uma infra-estrutura compartilhada que ofereça o serviço de mapeamento de um rótulo sem semântica (que referencia um objeto) para o meta-dado associado a este objeto. Nessa infra-estrutura, provedores inserem o meta-dado do objeto e associa-o com um rótulo, enquanto os usuários fazem o processo inverso, submetendo o rótulo à infra-estrutura e recebendo como resposta o meta-dado do objeto. A infra-estrutura do SFR usa DHT para mapear *strings* de 160 bits, *SFRTags*, para registros de objetos, *o-records*. Um registro de objeto *o-record* é mostrado na Figura 3-7. A infra-estrutura não armazena objetos, apenas os *o-records*. O SFR pode utilizar qualquer tecnologia de busca escalável, mas sua implementação de teste usa o *Chord* [51]. O campo *location* é definido no momento de inserção do *o-record* e mantém um ou mais valores que descrevem a localização do dado correspondente ao *SFRTag*. O campo *location* pode ser um par endereço IP e porta, um nome de domínio ou um outro *SFRTag*. Essa infra-estrutura de resolução não limita em funcionalidade as aplicações, visto que o conteúdo e forma do campo *oinfo* é definido pela aplicação e pode incluir dados do tipo de protocolo, nome do caminho no servidor, etc. Por último, o campo *ttl* permite estratégias de *cache*.

Uma infra-estrutura SFR é composta por servidores (portal), clientes e *relays* SFR (Figura 3-8, originalmente apresentada em [52]). As aplicações armazenam e requisitam *o-records* correspondentes a *SFRTags* usando o cliente SFR. Os clientes por sua vez acessam a infra-estrutura SFR através dos *Relays* SFR, que realizam as funções de *cache*.

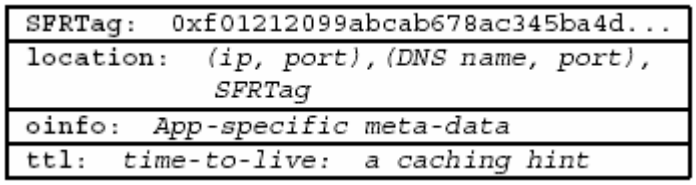


Figura 3-7 - o-record

Em relação à integridade, os rótulos *SFRTags* e os registros *o*-record apresentam as propriedades: a) SFRTag é o resultado de uma função *hash*, sem significado humano (legibilidade); b) Provedores de conteúdo podem criar referências sem consultar uma autoridade de nomeação; c) Somente o criador do registro *o-record* pode atualizá-lo;

O processo de busca nessa infra-estrutura é simples. Uma aplicação envia a solicitação para uma determinada *SFRTag* para o seu *Portal* ou *Relay* e se o rótulo está na infra-estrutura, o nó DHT responsável retorna o *o-record*. Para reduzir a latência ou

fazer balanceamento de carga entre os nós do SFR, a infra-estrutura permite que os *Relays* façam *cache* de *o-records*, que os nós DHT façam *cache* de localização e que os Portais mantenham *cache* dos objetos mais populares. Para o caso de acesso local aos objetos no caso de falhas na conexão no enlace de acesso à Internet, o SFR dispõe de um mecanismo nos *relays*, conhecido como *org-store*. Neste caso, os *relays* mantém uma cópia no *org-store* dos *o-records* criados (ou modificados) dentro da organização, antes de criar (ou atualizar) nos nós DHT da infra-estrutura.

Uma aplicação direta de uma RRS com SFR seria a Web sobre SRF. Na Web sobre SFR, o tratamento dos nomes inteligíveis para o usuário é realizado fora do RRS. Nesta situação, os portais de busca funcionariam da mesma maneira que hoje, com a diferença que eles retornariam rótulos sem semântica ao invés de URL baseadas no DNS. O objetivo de uma Web sobre SFR é permitir migração e replicação nativa de objetos. Toda as informações necessárias para alcançar um objeto na Web (endereço IP e porta do servidor, nome do caminho no servidor) podem ser encapsuladas pelo *SFRTag*, visto que o criador do objeto insere no *o-record* as informações necessárias para encontrá-lo e armazena o registro do objeto na infra-estrutura SFR.

Um dos benefícios diretos da Web sobre SFR é a facilidade de migração. Por exemplo, se um objeto referenciado por um rótulo *SFRTag*, muda para outro servidor Web, em outro nome de caminho, o provedor do conteúdo (dono do objeto) precisa apenas mudar os campos *location* e *oinfo* no *o-record*. As páginas que apontam para o objeto não necessitam de alterar suas referências. O outro benefício citado anteriormente, refere-se a replicação flexível de objetos. Em resposta a uma solicitação para uma *SFRTag*, a infra-estrutura pode retornar várias localizações e caminhos.

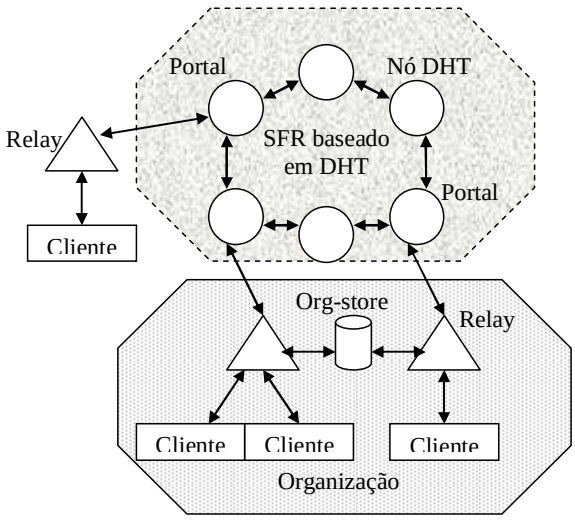


Figura 3-8 - Infra-estrutura SFR e seus componentes

Concluindo, uma das desvantagens do SFR é sua implantação na rede, pois os navegadores Web teriam que utilizar a infra-estrutura do SFR para resolver rótulos em metadados (ex: endereços IP e nomes de caminho) e isto evidentemente não aproveita o legado dos navegadores Web atuais, requerendo uma modificação universal. Porém, algumas estratégias de implementação parcial são apresentadas no artigo seminal [52].

3.4.8. Infra-estrutura I3

A Infra-estrutura de Indireção para a Internet (*Internet Indirection Infrastructure* - *i3*) [30] é uma rede sobreposta (*overlay network*) de propósito geral para facilitar a implantação de serviços como *multicast*, *anycast* e mobilidade. Ao invés de enviar pacotes explicitamente para o destinatário, cada pacote é associado a um identificador (rótulo), o qual é usado pelo receptor para receber o pacote. Este nível de indireção introduzido pela rede *i3*, portanto, desacopla o ato de enviar do ato de receber e permite que a rede suporte uma variedade de serviços de comunicação.

Este modelo de serviços pode ser visto como uma abstração do modelo de comunicação baseado em *Rendezvous*. Pacotes formam um par (*id, dados*) onde *id* um identificador de *m* bits e *dados* consiste da carga útil do pacote. Para indicar seu interesse nos pacotes, os receptores usam "gatilhos" (*triggers*) na forma (*id, addr*), onde *id* representa o identificador do gatilho e *addr* representa o endereço de uma estação, consistindo de um endereço IP e uma porta. Um gatilho (*id, addr*) é "registrado" na rede *i3* para indicar que todos os pacotes com um identificador *id* devem ser encaminhados (no nível IP) pela rede *i3* para a estação identificada como *addr*.

Baseado neste modelo, a criação de um grupo *multicast*, por exemplo, é equivalente a fazer com que todos os interessados no grupo registrem "gatilhos" com o mesmo identificador. A proposta da rede *i3* também inclui uma forma mais complexa de pacotes e gatilhos com empilhamento de identificadores (ao invés de um simples identificador), como meio para suportar níveis adicionais de indireção e permitir a implantação de outros serviços.

A rede *i3* trata a mobilidade das estações e consegue manter a conectividade fim-a-fim entre as estações móveis da seguinte forma. Quando uma estação se move de uma sub-rede para outra e troca o endereço $\boldsymbol{E_1}$ para outro endereço $\boldsymbol{E_2}$, ela deve atualizar os gatilhos de (*id*, $\boldsymbol{E_1}$) para (*id*, $\boldsymbol{E_2}$). A estação emissora não precisa tomar conhecimento da mobilidade do receptor, uma vez que os pacotes são roteados com base no identificador. Dessa forma, ambas as estações emissora e receptora podem se mover simultaneamente. A proposta não discute o mecanismo para troca o endereço $\boldsymbol{E_1}$ para outro endereço $\boldsymbol{E_2}$ durante a mobilidade, pois esta é uma questão tratada no nível IP.

Quanto ao seu funcionamento, a rede *i3* consiste de um conjunto de servidores e sistemas finais. Os servidores que armazenam gatilhos e encaminham pacotes (usando IP) para outros servidores *i3* e para sistemas finais. Identificadores e gatilhos possuem significado apenas na rede *i3* e cada servidor armazena um subconjunto de gatilhos. Em um determinado instante, um gatilho está armazenado em apenas um servidor e cada sistema final conhece um ou mais servidores. Sistemas finais contactam a rede *i3* apenas para enviar pacotes ou inserir gatilhos.

Quando uma estação deseja enviar um pacote, ele o encaminha para um dos servidores conhecidos e se o servidor contatado não contém um gatilho que dispara a entrega do pacote para alguma estação interessada, o pacote é encaminhado para outro servidor. Este processo continua até que o pacote alcança um servidor que armazena um gatilho que casa com o pacote, quando então o pacote é entregue a sistema final através do IP. Para saber para qual estação encaminhar o pacote, a rede *i3* usa DHT para indexar o identificador (os autores usaram o esquema de busca *Chord* [51], mas argumentam que outros mecanismos como *CAN*, *Pastry*, *Tapestry* ou outros podem ser

usados [30]). Dessa forma, o pacote é encaminhado pela rede *i3* até o servidor responsável pelo identificador. No nível de roteamento IP, nenhuma mudança é necessária. É importante mencionar que a rede *i3* não armazena pacotes: ela apenas os encaminha. A rede *i3* provê um serviço do tipo "melhor esforço", tal como na Internet atual, e não implementa confiabilidade sobre a rede IP.

Na Internet atual um nome DNS é mapeado para uma estação como o resultado de uma consulta explícita que precede a transferência. Na rede *i3*, o mapeamento identificador-estação é integrado e automático (não requer consulta explícita): quando uma estação deseja acessar outra, o identificador do receptor pode ser obtido através de um *hash* sobre o nome DNS (ou URL ou uma chave publicamente conhecida) associado ao receptor. Dessa forma, o envio do pacote consiste em computar o identificador do receptor e encaminhar para um servidor *i3*. Uma implementação da rede *i3*, feita pelos autores da proposta, usa um identificador de 256 bits, dos quais 128 são usados pela aplicação e demonstrou-se que esta escolha é suficiente para obter *hashes* disjuntos, com baixa probabilidade de colisão (ou seja, que aplicações em estações diferentes obtenham o mesmo identificador). Por se tratar de uma rede de sobreposição, não há necessidade de mudança no endereçamento do nível IP e a implantação da rede *i3* em escala da Internet poderia se dar de forma gradativa e incremental.

3.4.9. Arquitetura SOS

Keromytis *et al.* propuseram uma arquitetura chamada *Secure Overlay Services* – SOS (serviços sobrepostos seguros) [41] para mitigar o efeito e o volume de ataques de DoS na Internet atual. SOS é rede sobreposta *(overlay network)* composta por nós que se comunicam entre si usando a rede subjacente (rede IP), com o objetivo de gerenciar a comunicação entre a origem e o destino para torná-la segura sobre a rede IP. Freqüentemente, os nós irão executar funções de roteamento para despachar pacotes entre eles. A arquitetura assume que os nós participantes da rede sobreposta são conhecidos (públicos) e, portanto, também podem estar sujeitos a ataques. Entretanto, certos papéis que cada um dos nós da rede sobreposta assume são mantidos em segredo.

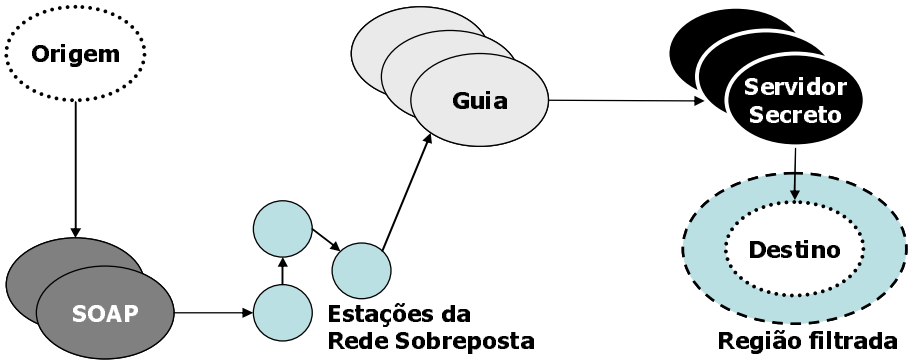


Figura 3-9 – A arquitetura básica do SOS

A **Figura 3-9** mostra uma visão geral da arquitetura SOS que protege uma estação (ou sub-rede) destino, de maneira que este receba apenas tráfego legitimado. Fundamentalmente, o objetivo da infra-estrutura SOS é distinguir o tráfego autorizado e não-autorizado, permitindo que apenas este último alcance o destino, enquanto que o

primeiro deve ser descartado (ou ter sua taxa reduzida). Dessa forma, a rede SOS se torna um grande *firewall* distribuído que distingue o tráfego malicioso e legítimo.

O procedimento para estabelecer a rede SOS, determinar o papel dos nós participantes e ingressar na rede SOS é o seguinte. Primeiro, uma estação destino decide participar da rede SOS e instala um filtro nos seus arredores e seleciona um certo número de nós participantes para agirem como seus "servidores secretos", ou seja, aqueles únicos habilitados a encaminhar tráfego para esta estação participante.

Quando um nó da rede SOS é informado que irá agir como "servidor secreto" para uma estação participante ela, após verificar a autenticidade desse pedido, calcula várias chaves (usando funções de *hash*) baseadas no endereço de rede daquela estação destino. Cada chave irá identificar uma quantidade de nós da rede sobreposta que irão agir como "guias" para aquela estação destino. Tendo identificado os "guias", os "servidores secretos" os contactam, notificando-os para assumirem tal compromisso. Os "guias", após verificarem a autenticidade desse pedido, guardarão a informação necessária para encaminhar o tráfego para a estação destino através daqueles "servidores secretos".

Quando uma estação de origem deseja transmitir para a estação destino, o encaminhamento de um pacote na arquitetura SOS é feita em cinco estágios: 1) o ponto de origem encaminha um pacote para um nó especial da rede sobreposta chamado "ponto de acesso seguro da rede sobreposta" (*secure overlay access point* - SOAP). SOAP é um nó que recebe pacotes cuja legitimidade não foi ainda verificada e executa essa verificação através de algum protocolo seguro (ex: IPsec ou TLS); 2) o SOAP encaminha o pacote (através de outros nós) para o nó "guia"; 3) o "guia" encaminha o pacote para o "servidor secreto", cuja identidade só é conhecida por ele; 4) o "servidor" secreto encaminha o pacote para o destino; e 5) o filtro ao redor do destino permite a passagem apenas do tráfego oriundo de um dos "servidores secretos". Todos os pacotes são encaminhados entre os nós da arquitetura com o uso de DHT (os autores sugerem o *Chord* [51], mas argumentam que outros métodos podem ser usados).

Este esquema é robusto contra ataques de DoS pelos seguintes motivos. Primeiro, se uma estação SOAP (o ponto de entrada na rede SOS) é atacada e falha, as estações de origem podem escolher outra estação SOAP disponível. Se um nó interno da rede SOS é atacado, o nó simplesmente é considerado ausente da rede e o mecanismo DHT (*Chord*) se auto-organiza, provendo novos caminhos sobre a rede para alcançar os "guias". Na rede SOS nenhum nó é mais sensível ou mais importante do que outro e alguma redundância permite que até mesmo os "guias" e os "servidores secretos" sejam atacados (por acaso ou propositadamente) sem causar queda de funcionalidade na rede sobreposta. Segundo, se a identidade de um "servidor secreto" for descoberta e este for alvo de um ataque, ou se algum ataque partir de um endereço IP de um "servidor secreto" em direção a uma estação destino, este pode dispor de mecanismos para trocar o seu conjunto de "servidores secretos".

Observa-se que se um atacante conseguir uma autorização legítima para envio de tráfego para alguma estação destino, este poderá conduzir um ataque de DoS distribuído (DDoS) usando múltiplas estações SOAP como multiplicadores de tráfego. Neste caso, os autores discutem que um "guia" ou um "servidor secreto" pode usar algum

mecanismo de *pushback*[6] para solicitar aos SOAP que revoguem a autorização da estação de origem. A resistência oferecida pela arquitetura SOS aos ataques de DoS é maior quanto maior for o número de nós na rede sobreposta.

3.4.10. Prevenção de DoS através de "Aptidão para uso de Recursos"

Em [37], Anderson *et al.* sugerem uma nova abordagem restringir abusos de DoS na Internet: Prevenção de DoS através de "Aptidão para Uso de Recursos" (AUR). Basicamente, ao invés de qualquer estação poder livremente enviar quaisquer dados para qualquer outra estação a qualquer momento, primeiramente ela deve obter uma permissão da estação destinatária. Caso o receptor concorde em receber o tráfego do emissor, ele deve emitir uma "ficha" (que o autor chama de *token* ou *capability*) que "habilita" o emissor a enviar o tráfego em sua direção. Este mecanismo, portanto, permite rotular cada pacote do tráfego legítimo.

O argumento dos autores é que qualquer solução completa para ataques de DoS deve permitir o controle do uso dos recursos por parte do seu proprietário, o que inclui a possibilidade de impor limites para o uso do recurso. Procedendo desta forma, ataques de IP *spoofing* podem ser combatidos, pois ambos emissor e receptor concordam com o envio e recebimento do tráfego e, caso a autorização concedida não esteja presente nos pacotes recebidos, o tráfego deve ser descartado. Na prática, cada receptor deve gerar um certificado e enviá-lo ao solicitante como sendo a permissão para o envio de dados. O certificado é assinado com a chave privada do receptor e deve constar em cada pacote subseqüente. Portanto, se cada pacote passa a incluir o certificado, roteadores ao longo do caminho poderão verificar que aquele pacote é autorizado e poderão descartar o tráfego não autorizado. O certificado é solicitado e recebido através de uma conexão segura estabelecida para este fim.

Na prática, a idéia é a criação de servidores de Solicitação-de-Envio (*Request-to-Send* - RTS), que ficam nas fronteiras das redes (poderiam estar junto dos roteadores que executam BGP) e emitem autorizações (*tokens*) aos solicitantes. Os RTS são acoplados a um outro tipo de servidor, denominado "ponto de verificação" (*Verification Point* – VP), que figuram no meio do caminho dos dados e fazem o papel de controladores de acesso, verificando a existência de *tokens* válidos. Os domínios autônomos que possuem clientes que desejam filtrar o tráfego devem incluir o endereço IP do seu servidor RTS nos anúncios BGP. Os anúncios em cadeia criam uma hierarquia de RTSs e permitem aos remetentes descobrirem para quais RTS devem encaminhar suas solicitações de *token* quando desejam comunicação com um receptor específico.

Se uma estação remetente deseja enviar algum tráfego, ele envia ao receptor, através da cadeia de RTS dos domínios do caminho, uma solicitação para transmitir. O receptor decide enviar um *token* ou não, em função da disponibilidade de recursos, que pode também envolver políticas de acesso com base no solicitante ou tipo de tráfego etc. Se desejar autorizar, ele fabrica uma seqüência de *tokens* $h_1, ..., h_K$, onde h_i são valores construídos com uma função de *hash* de uma via (ex: MD5), fáceis de computar e

[6] O *pushback* consiste de um mecanismo de comunicação no qual um roteador sob ataque de DoS solicita aos roteadores anteriores a ele, considerando o sentido do tráfego, que filtrem aqueles pacotes, evitando que eles encaminhem tráfego que será descartado adiante, contribuindo assim para a preservação da largura de banda naqueles enlaces.

difíceis de “quebrar”. Cada valor h_i indica a autorização para transmitir *n* pacotes nos próximos *t* segundos. Então, o receptor envia o último valor h_K, junto com um valor randômico s_0 que representa a seqüência inicial, para o remetente através da mesma cadeia de RTS. Cada servidor RTS associa esses valores com um fluxo a ser permitido pelo VP a ele acoplado, ou seja, a autorização é dada por todo o caminho reverso do pedido de solicitação.

Com a autorização estabelecida e de posse do *token* inicial h_K, o remetente está apto a transmitir. Cada pacote é rotulado com o *token* e um número de seqüência e encaminhado através dos VPs do caminho, que verifica a validade do tráfego e decide encaminhá-lo ou não em direção ao receptor. Quando *n* pacotes são encaminhados ou *t* segundos são decorridos, o VP invalida o fluxo. Dessa forma, a largura de banda fica reservada apenas para os fluxos autorizados. Para evitar que o fluxo seja desautorizado antes que o remetente consiga enviar todos os dados, o receptor envia para o remetente o componente h_{K-1} do *token* sempre que receber aproximadamente *n* pacotes, renovando a autorização para *n* pacotes adicionais. O remetente passa então a usar o novo valor h_{K-1} para os próximos *n* pacotes e o VP, ao receber um novo *token* h_{K-1} com um número de seqüência válido, tem como validar o novo valor do *token* e assumi-lo como o *token* vigente para o fluxo usando a mesma função de *hash* adotada pelo emissor do *token*. Observe que isto é possível porque cada token h_{K-x} é computado com base na seqüência do primeiro pacote que ele autoriza, ou seja, o pacote $s_0+n(x+1)$. Esse mecanismo requer uma nova autorização após *nK* pacotes transmitidos, o que requer algum critério para o estabelecer o valor de *K*.

A proposta discute também mecanismos para garantir que os próprios servidores RTS não sejam submetidos a ataques de DoS, sugerindo que estes somente aceitem tráfego de clientes locais ou de servidores RTS vizinhos (conhecidos a partir de anúncios BGP). A proposta como um todo apresenta uma solução que pode ser implementada de forma incremental, ou seja, pode ser adotada através de parcerias entre domínios sem interferir na Internet atual até que todos os domínios estejam totalmente integrados.

3.5. Síntese das Propostas

A seção 3.3 apresentou as principais questões de projeto que hoje representam problemas não devidamente solucionados na atual arquitetura da Internet. Por outro lado, a seção 3.4 apresenta novas arquiteturas com propostas de mudanças cujo objetivo é compatibilizar a Internet atual com os seus princípios fundamentais ou permitir uma evolução da direção de demandas, tecnologias ou problemas não identificados inicialmente. Esta seção procura apresentar uma síntese das principais modificações propostas pelas arquiteturas apresentadas na seção 3.4, relacionando-as e classificando-as de acordo com as questões de projeto apresentadas na seção 3.3. A Tabela 3-1 apresenta uma relação de todas as propostas com as questões de projeto, que serão detalhadas nas próximas sub-seções.

Tabela 3-1 – Relação entre propostas e questões de projeto

Proposta	Função					
	Endereçamento, Nomeação	Roteamento	Segurança	Mobilidade	Transparência	Camadas
LNA	★			★	★	
FARA	★	★		★		
NIRA		★				
IPNL	★	★			★	
RBA					★	★
Plutarch	★	★			★	
SFR	★					
i3	★	★		★		
SOS			★			
AUR[7]			★			

3.5.1. Endereçamento e Nomeação

A questão da camada de rede na arquitetura atual da Internet que acopla, através do uso do endereço IP, a localização do nó na rede (endereçamento) à sua identidade (identificação) em um único atributo, gerou um grande número de propostas para sua atualização ou sua inteira substituição. Diversas soluções para os problemas da proposta original do IP vêm sendo apresentadas, algumas com objetivos específicos de tratar mobilidade, segurança e roteamento etc, e outras de propósito mais geral onde diversos problemas são supostamente solucionados com uma única mudança, como FARA.

A arquitetura FARA provê um arcabouço consistente e de alto-nível com a definição um conjunto abstrato e modular de componentes e suas relações. Utilizando dois níveis conceituais, entidades e associações em um nível e o substrato de comunicação em outro nível, o modelo elegantemente separa localização de identificação. Desta forma, devido à generalidade da arquitetura e através de um processo de instanciação, é possível desenvolver estratégias independentes e distintas para endereçamento, nomeação e mobilidade, entre outras.

LNA cria um novo endereço fim a fim e desacoplam a identificação da localização. O EID é para identificação e o endereço IP para localização/roteamento (esse é o aspecto de roteamento). Em comparação com FARA, LNA adota estratégias similares. Porém, FARA sendo um arcabouço para instanciação de modelos, tem a vantagem da abstração, onde diversas soluções específicas podem ser derivadas.

IPNL mantém o acoplamento de localização e identificação, através do endereço IPNL e inclusive permite que o próprio FQDN seja usado para essa finalidade (além do endereço IPNL). Ou seja, IPNL cria também um novo endereço, o IPNL. IPNL permite o uso de endereços IP distintos dentro de cada região, que vão sendo trocados por novos endereços IP à medida que um pacote vai passando de uma região para outra.

Plutarch permite endereços completamente distintos dentro de cada contexto. Um contexto de Plutarch é basicamente uma região de IPNL, mas que permite redes mais heterogêneas. Por exemplo, Plutarch permite que redes de sensores tenham

[7] Prevenção de DoS através de "Aptidão para uso de Recursos"

endereços que não sejam IP, porque, por exemplo, os sensores não têm capacidade de implementar uma pilha TCP/IP. As funções intersticiais devem fazer o mapeamento desses endereços.

Em termos de nomeação, a Internet dispõe de um mecanismo único e rígido para resolução de nomes em endereços IP (DNS). Assim, a movimentação e replicação de conteúdo (ex:, objetos Web) só é possível através de mecanismos auxiliares a estes serviços, restringindo sua flexibilidade. Para esta questão de projeto, a infra-estrutura SFR aparece como uma importante alternativa para solucionar os problemas de migração e replicação de objetos, para as aplicações que utilizem alguma forma de serviço de resolução de referências (RRS). Um exemplo típico é a Web, onde é necessário um RRS, neste caso o DNS, para mapeamento de URLs às suas localizações na rede. Ao invés de adotar uma estratégia de substituição integral do DNS ou de funcionalidade equivalente, a proposta do SFR aborda a questão partindo do ponto de vista que as referências não deveriam ter semântica legível ao usuário ou conter informações sobre domínios onde elas estão localizadas. O SFR mantém as funções originais do DNS e utiliza-se de seus recursos, podendo ser interpretado como serviços complementares.

3.5.2. Roteamento

A arquitetura de roteamento atual da Internet apresenta diversos problemas, tais como falta de um adequado nível de competição de mercado entre os sistemas autônomos, presença de um modelo não-unificado de segurança e roteamento, escalabilidade, instabilidade e convergência. Além disso, diversos trabalhos apontam que uma arquitetura de roteamento adequada deveria suportar recursos especiais (e.g., suporte à mobilidade), porém com a utilização de procedimentos simples com consumo mínimas de recursos nos roteadores.

Uma solução para o problema de competitividade entre os sistemas autônomos é descrita no projeto de uma nova arquitetura de roteamento chamada NIRA. A inovadora arquitetura NIRA tem como principal objetivo permitir ao usuário a possibilidade de escolha das rotas no nível de domínio. Vários desafios de ordem práticas foram abordados nesta proposta, tais como os mecanismos de descoberta e representação de rotas, sempre observando os requisitos de escalabilidade, robustez, eficiência etc., que foram cuidadosamente considerados.

As conseqüências da criação de um novo endereço fim a fim da arquitetura IPNL são alterações nos mecanismos de roteamento, com a possibilidade de se fazer um roteamento por FQDN. Já no modelo de arquitetura FARA, o substrato de comunicação é o componente responsável pela entrega de pacotes às entidades. Apesar de ser um modelo para instanciação e não fornecer detalhes sobre estratégias de encaminhamento, a arquitetura assume que o substrato de comunicação deve possuir mecanismos não orientados à conexão para entrega dos pacotes das associações

3.5.3. Segurança

Ataques de DoS são os mais temidos na Internet, porque são fáceis de fazer e difíceis de combater. Muitas propostas já foram feitas para combatê-los, mas não se conhece um mecanismo que seja completamente eficiente na arquitetura da Internet atual.

A abordagem da proposta SOS não resolve completamente o problema de DoS na Internet, mas consegue grandes avanços. A idéia é criar um nível adicional (rede

sobreposta) sobre o nível IP para estabelecer filtros de tráfego ao redor de uma estação, tal como um grande *firewall* distribuído. Além disso, as estações precisam solicitar autorização para enviar dados para outra estação, através dos pontos de acesso seguros (SOAP). Outra idéia interessante da proposta SOS é que os nós da rede sobreposta assumem papéis aleatoriamente, e a idéia dos "servidores secretos" que formam a malha de filtro de uma estação reduz sobremaneira a possibilidade de ataques contra a infra-estrutura, principalmente quando a rede SOS é grande. Além disso, se isso ocorrer, a estação atacada é desligada sem interferir na rede SOS, graças à grande resiliência conferida à rede sobreposta pelo mecanismo DHT.

A proposta para prevenção de DoS através de “Aptidão para uso de Recursos” (AUR) também se baseia na idéia de restringir a comunicação apenas para estações autorizadas. Diferentemente da rede SOS, entretanto, esse mecanismo não propõe uma rede sobreposta, mas a inclusão de dois elementos cruciais na borda dos domínios autônomos: o servidor RTS (*request to send*) e o VP (*verification point*). Esses servidores podem ser fisicamente separados ou em uma única estação (geralmente um roteador), mas atuam de forma integrada. O servidor RTS é quem intermedia o pedido de autorização para envio de tráfego para uma estação daquele domínio, ou participa do encaminhamento de um pedido de autorização que cruza o seu domínio em direção ao domínio vizinho. A autorização é concedida pela estação destino e o RTS informa ao VP alguns parâmetros que irão garantir a passagem do tráfego oriundo da estação emissora. A autorização é a abstração de uma "ficha" implementada por *hashes* (ex: MD5), os quais são difíceis de forjar, mas fáceis de verificar. Cada pacote do fluxo deve carregar a ficha que é verificada pelo VP e o tráfego pode ser liberado, barrado, ou suavizado a uma taxa definida pela estação destino.

Considerando as propostas SOS e a AUR, ambas são consideradas "proativas" (evitam o DoS ao invés de apenas combatê-los) e podem ser implantadas de forma incremental e gradativa, mas o impacto na estrutura atual da Internet é menor com a SOS, que se trata de uma rede sobreposta e, portanto, não faz exigências de alteração na camada IP. O esquema DHT usado nas redes sobrepostas, entretanto, pode introduzir atrasos de roteamento quando os nós logicamente próximos estão na verdade geograficamente distantes. Embora uma análise mais detalhada seja requerida, a proposta AUR pode introduzir menor atraso médio, mesmo tendo que verificar a autenticidade de cada pacote do fluxo.

3.5.4. Mobilidade

Mobilidade não foi um requisito considerado pelo IP durante sua especificação original, em função de que praticamente não havia essa necessidade à época. O IP móvel (MIPv4), portanto, é um ajuste na Internet original, que, apesar de elegante, ainda apresenta alguns problemas, como discutido na seção 3.3.4.

Muitos dos problemas enfrentados pelo MIPv4 são resolvidos naturalmente no MIPv6 [53], visto que o IPv6 oferece mobilidade nativa. Entretanto, o fato de que a estação móvel recebe um novo endereço IP da rede visitada é basicamente a fonte de boa parte dos problemas, pois a Internet atual usa o endereço IP como localizador da rede e identificador do sistema final. Outra parte dos problemas podem ser atribuídos à falta de transparência introduzida pelos intermediários (*firewalls*, NATs) e VPNs.

A Arquitetura de Nomes em Camadas [1] propõe a separação semântica da identificação do serviço (ou informação) da estação onde ela reside e introduz o conceito de identificador de serviço (SID) e identificador da estação final (EID). Com a introdução de níveis adicionais para nomeação, essa arquitetura permite localizar estações através do EID, independentemente da sua localização física ou topológica, o que acaba sendo um benefício para a mobilidade. Ou seja, se uma estação se movimenta para outra rede e recebe outro endereço IP, ela manterá a ligação com o seu EID, e apenas precisa atualizar o seu IP para a camada de resolução de EIDs.

Outra proposta que traz benefícios para a mobilidade é a arquitetura *i3* [30], pois sugere uma rede sobreposta capaz de manter a conectividade fim-a-fim entre as estações móveis através da atualização de "gatilhos" (vide seção 3.4.8). Quando uma estação se move de uma sub-rede para outra e troca o endereço E_1 para outro endereço E_2, ela deve atualizar os gatilhos de (*id*, E_1) para (*id*, E_2), para que todos os pacotes que contém o rótulo *id* sejam encaminhados para o novo endereço. A estação emissora não precisa tomar conhecimento da mobilidade do receptor, uma vez que os pacotes são roteados com base no identificador. Dessa forma, ambas as estações emissora e receptora podem se mover simultaneamente.

3.5.5. Transparência Fim a Fim

De acordo com a seção 3.3.5, o princípio da transparência da Internet original pode ser definido por duas características: 1) a existência de um endereço com semântica fim a fim; 2) a rede se comportar como uma “caixa preta”, sem alterar o conteúdo dos pacotes. Essa seção discute as propostas de mudanças, com relação à sua abordagem dessas duas características.

A arquitetura LNA (e também DOA [31], que é uma especialização de LNA) são direcionadas para tratar de aspectos da camada de rede. LNA restaura a semântica fim a fim dos endereços (perdida na Internet atual), introduzindo um novo endereço, o EID. Ele somente é usado para identificar os sistemas finais, não tendo nenhuma utilização para localização e roteamento. Essa última função continua com os endereços IP, que têm semântica limitada nas regiões. LNA permitem que haja processamento nos pacotes, mas sob controle dos sistemas finais, que delegam essa função explicitamente aos intermediários, incluindo-os na arquitetura de maneira natural.

IPNL também cria novos endereços fim a fim, os endereços IPNL e também permite que o FQDN seja usado como endereço. Na realidade, o endereço mais estável é o próprio FQDN. Os intermediários (NATs) são incorporados sem costuras à arquitetura, como *nl-router*s, que para os roteadores são vistos como sistemas finais. Desse modo, na camada IPNL o pacote não é alterado significativamente. Por outro lado, na camada IP os endereços são alterados a cada região. Como o significado dos endereços IP é restrito a cada região, então se pode dizer que IPNL também preserva a característica da caixa-preta, porque os *nl-router*s são vistos como sistemas finais pelos roteadores (ou seja, o tráfego é destinado a eles na camada IP, quando o sistema final está em outra região).

Em RBA, nenhum tratamento é dado aos endereços fim a fim. No entanto, a arquitetura incorpora mudanças nos pacotes de maneira natural, pelos papéis, sob o controle dos sistemas finais. Desse modo, assim como nas outras propostas, a integração

dos sistemas finais ocorre de maneira natural (e não forçada, como na Internet atual, onde pacotes freqüentemente são "seqüestrados", ex: nos *proxies* transparentes).

Plutarch rompe de maneira radical com o conceito de transparência fim a fim. Por um lado, cada contexto pode ter o seu próprio esquema de endereçamento e o mapeamento entre endereços deve ser realizado pelas funções intersticiais. Por outro lado, essas funções permitem que os pacotes sejam modificados no trânsito de origem a destino, com um controle explícito por parte da administração da rede (e não do sistema final, como ocorre em LNA e RBA).

3.5.6. Modelo em Camadas

O modelo em camadas utilizado na Internet possui várias vantagens, como modularidade, encapsulamento e regras rígidas e organizadas para o processamento dos cabeçalhos. No entanto, esses princípios têm sido violados na Internet atual, porque cada vez mais dispositivos (por exemplo, os intermediários) lêem e escrevem informações nos cabeçalhos de outras camadas. DOA é uma arquitetura que não altera o modelo em camadas, mas tenta evitar a sua violação conceitual. Através da delegação explícita de certas atividades para intermediários, os sistemas finais os identificam como locais legítimos de processamento dos cabeçalhos de várias camadas.

No entanto, DOA não se propõe a solucionar outros problemas que ocorrem na prática com o modelo em camadas, como a falta de eficiência e a incapacidade de alocar de certas funções a camadas específicas. Entre as arquiteturas estudadas, RBA é a única que explicitamente questiona se o modelo em camadas continua sendo a melhor abstração para software de rede e propõe um modelo alternativo baseado em papéis, onde a pilha de protocolos é trocada por um "amontoado de protocolos". Fazendo isso, ela troca a violação entre as camadas por interações explícitas entre papéis e oferece a possibilidade de ganhos de eficiência no processamento. No entanto, não é simples organizar esse processamento no software dos sistemas finais e intermediários, de modo a haver uma colaboração efetiva entre todos os seus componentes. Por outro lado, a implementação de RBA exigiria uma modificação completa na Internet, inclusive no funcionamento dos roteadores.

3.6. Comentários Finais

Apesar do seu enorme sucesso, a arquitetura da Internet é amplamente reconhecida como sendo longe do ideal. O seu crescimento, penetração e importância têm realçado muitas das suas imperfeições e aumentado a urgência pelas respectivas soluções [1]. A necessidade por mudanças arquiteturais nunca foi tão forte, como mostra a explosão de críticas e novas propostas oriundas da comunidade científica (ex: LNA [1], FARA [11], NIRA [32], IPNL [16], DOA [31], RBA [2], Plutarch [13], i3 [30], SOS [41] e muitas outras), as quais abordam vários aspectos da arquitetura da Internet: endereçamento, roteamento, nomeação, modelo de camadas, transparência, segurança, mobilidade.

Este documento apresentou uma visão crítica sobre a Internet, não das questões corriqueiras do dia a dia, como aplicações, usuários, desafios de engenharia de curto prazo, mas das questões abstratas da sua arquitetura, cuja repercussão pode ser sentida a médio e longo prazo. A abordagem adotada procurou seguir uma visão didática em cinco fases, as mesmas que foram trilhadas pelos autores (não necessariamente na mesma ordem) no estudo e descoberta das informações apresentadas. Primeiramente, uma visão

de onde partiu e aonde chegou a Internet. Depois, uma visão da arquitetura original da Internet, as decisões pragmáticas adotadas na Internet atual, as questões que deveriam ser tratadas para seguir um caminho coerente de evolução, as propostas de mudanças e a sumarização e comparação dessas e propostas. Ao final, discutiremos a seguir uma sexta fase, onde se questionam as reais possibilidades de evoluir a arquitetura da Internet de maneira significativa.

3.6.1. É possível mudar a Internet?

Evoluir a Internet atual não é uma tarefa fácil (seção 3.1.1), principalmente na "cintura fina" da ampulheta de protocolos (seção 3.2.5). A Internet original foi projetada para incorporar mudanças na sua estrutura de modo natural, mas aos poucos foi perdendo essa característica. Atualmente, as evoluções mais comuns estão nas partes mais largas da ampulheta, ou seja, nas aplicações (ex: p2p) e nas tecnologias (ex: IP sobre redes óticas). Por isso, algumas perguntas são pertinentes: Até que ponto é possível mudar a arquitetura da Internet? Com que velocidade isso pode ser feito? Qual o custo?

Um bom ponto de partida é a transição de IPv4 para IPv6. Existem várias razões para a Internet migrar para IPv6, mas também vários empecilhos, como por exemplo, a falta de um bom modelo de negócios que efetivamente traga benefícios aos provedores, diante da enorme tarefa e do alto custo que eles devem encarar. Apesar do esforço de definição do novo protocolo ter produzido seu primeiro padrão em dezembro de 1995 (RFC 1883[8]), até o presente momento a implantação comercial do IPv6 ainda pode ser considerada incipiente. Existem muitos esforços sendo feitos para que isto ocorra (ex: o Fórum IPV6, www.ipv6forum.org) e empresas significativas já estão trabalhando no seu suporte (como o Google). No entanto, não se sabe ainda quando haverá uma real transição para este novo protocolo.

A lição importante sobre a transição para o IPv6 é que mesmo uma pequena mudança, mas que afeta grande parte da infra-estrutura da Internet, pode ser considerada de grande complexidade. Principalmente, *deve haver uma forte razão econômica para viabilizá-la*. A mudança para IPv6 é considerada pequena porque afeta principalmente o tamanho do endereço IP, que passa de 32 bits para 128 bits. No entanto, ela implica em uma mudança nos roteadores, ou seja, no núcleo da rede. Como uma grande instabilidade na Internet pode ser gerada, a transição é motivo de preocupação. Por outro lado, uma grande quantidade de sistemas finais (clientes e servidores) deve ser modificada também, gerando uma possível inconveniência aos usuários.

Aparentemente, a melhor forma de se implementar uma modificação consistente na Internet é possibilitar aos seus usuários usufruir dela antes que a arquitetura tenha que ser alterado. Isto é possível em caso de técnicas de sobreposição de redes (*overlay*), que acrescentam suas funcionalidades sobre a camada IP atual. Um bom exemplo são as redes p2p, que constroem redes sobrepostas, com endereçamento e roteamento próprios. Muito se tem discutido de embutir o roteamento p2p nos roteadores, mas por enquanto a comunidade está conseguindo obter os benefícios sem que esse sacrifício seja necessário. Por outro lado, algumas vezes redes sobrepostas podem ter um desempenho abaixo do aceitável, o que pode comprometer a sua aceitação.

---

[8] Posteriormente, substituída pela RFC 2460, de dezembro de 1998 (www.ietf.org/html.charters /ipv6-charter.html).

Considerando as propostas analisadas na seção 3.4, um método de identificar a sua factibilidade é avaliar o nível de evolução (incremental) ou revolução (radical) necessário à sua implantação. Por exemplo, as propostas SOS e i3 são baseadas em sobreposições, de modo que sua implantação seria facilitada. Outras propostas exigem mudanças na arquitetura, mas apenas nas suas bordas. As arquiteturas LNA e DOA não necessitam que os roteadores sejam modificados, mas apenas que os sistemas finais. A vantagem é não afetar a estabilidade dos provedores. Por outro lado, outras propostas assumem mudanças drásticas na arquitetura, ou seja, a sua adoção levaria a uma "outra Internet", pelo menos na camada de rede. Exemplos são FARA, RBA e Plutarch.

Respondendo diretamente à pergunta dessa seção, a resposta é: "Sim, é possível mudar a Internet. Porém, tudo tem seu custo[9]". Nesse caso, o custo pode ser financeiro, operacional, político e até mesmo social. Propostas mais simples, ou que afetam menos drasticamente a cintura da ampulheta têm maior chance de vingar. Finalmente, outra pergunta vem à tona: "Por que tipo de benefício a comunidade da Internet está disposta a arcar com qual custo?". A resposta a essa pergunta é subjetiva.

3.6.2. As Doze Verdades sobre Redes: Humor ou Ironia?

A RFC 1925 [54], de 1º de abril de 1996[10], identifica doze verdades fundamentais sobre redes, que misturam humor com ironia, além de, obviamente, muita verdade aprendida com a experiência. As propostas de novas arquiteturas para a Internet devem ser avaliadas de acordo com essas verdades, para impedir que erros passados não sejam cometidos novamente. A primeira verdade diz que afinal de contas tudo "tem que funcionar", sendo portanto fiel ao lema da Internet que preconiza "código executando". Em outras palavras, não pode apenas ser algo com potencial teórico, mas tem que ser validado pela experiência prática. A oitava verdade diz que "é mais complicado do que você pensa", o que inviabiliza, por exemplo, soluções incompletas ou ingênuas. Nos próximos parágrafos, três verdades que contribuem com a discussão das arquiteturas são analisadas com mais detalhes.

A terceira verdade contém muito humor. "Com impulso suficiente, porcos podem voar bem. Entretanto, isso não é necessariamente uma boa idéia". Propostas extravagantes podem ser colocadas em prática. No entanto, o custo pode facilmente ultrapassar os benefícios em muitos níveis de magnitude. Por exemplo, arquiteturas que promovem grandes mudanças, como Plutarch e RBA, podem apresentar benefícios teóricos, mas o benefício pode não valer a pena.

A quinta verdade diz que "é sempre possível aglutinar várias problemas diferentes em uma única solução complexa e interdependente; na maioria dos casos isto é uma má idéia". Essa verdade é um alerta útil para barrar logo no início a tentação de resolver todos os problemas com uma única solução. Experiências anteriores na área de computação e especificamente de redes provaram que geralmente a proposta gerada é muito complexa (portanto, contrária aos princípios da Internet). Um ótimo exemplo é o modelo RM/OSI da ISO, que apesar de apresentar uma grande superioridade ao modelo da Internet, não resistiu ao inchaço de atribuições como camadas, entidades, protocolos,

---

[9] O ditado inglês "*there is no free lunch*" ("não existe almoço grátis") retrata bem esta máxima.

[10] A IETF publica RFCs no dia 1º de abril, com conteúdo humorístico (www.apps.ietf.org/rfc /apr1list.html e www.rfc-editor.org/rfcfaq.html).

controles e funcionalidades. Outro exemplo na área de linguagens de programação é a linguagem PL/1, criada pela IBM nos anos 1960, cujo objetivo foi combinar as melhores características de COBOL, FORTRAN e ALGOL. É fácil concluir qual foi o seu destino, pois atualmente a maioria das pessoas nunca ouviu falar dela!

A décima primeira verdade encerra um fato que é repetidamente praticado na comunidade acadêmica. “Toda idéia velha será proposta de novo com nome e apresentação diferentes, independente do fato de ela funcionar ou não”. Existe uma grande tentação em se reaproveitar o próprio conhecimento, valorizando anos de trabalho e experiência em determinadas áreas. Alguns exemplos são expandir o número de camadas, aumentar o número de indireções, sugerir soluções baseadas em circuitos virtuais com outro nome, utilizar difusão seletiva (*multicast*) de modo dissimulado, reeditar antigas e ineficazes idéias de gerenciamento e aplicar QoS em todos os problemas possíveis. Isso não significa que essas idéias não funcionam, mas são tipicamente soluções revitalizadas e redirecionadas para solucionar vários problemas novos que surgem.

3.6.3. Conclusões

Alterar a arquitetura da Internet é possível, mas o nível da intervenção depende do preço que se dispõe a pagar. Por outro lado, deve-se ao máximo procurar aprender com as experiências de mais de trinta anos no projeto e operação da rede. Essas são duas conclusões das seções anteriores. Outro ponto a observar é que os princípios fundamentais da Internet são ainda as melhores diretrizes disponíveis para balizar a comunidade, visto a grande preocupação das propostas apresentadas em mantê-los sempre que possível e restaurá-los em caso de quebra da sua semântica original.

A visão crítica proporcionada pela compreensão das doze verdades sobre redes (seção 3.6.2) é imprescindível. Com relação às novas arquiteturas, algumas perguntas devem ser feitas: “vai funcionar?”, “quais novos problemas podem ser introduzidos por elas?”, “a solução não é uma colcha de retalhos mal costurada?” e “os princípios que comprovadamente funcionam estão sendo aplicados?”. Aparentemente, o grande desafio é projetar uma solução simples na camada de rede (a cintura fina), mas ao mesmo tempo poderosa e de grande alcance, com ganchos que permitam que a maioria dos problemas sejam resolvidos nos sistemas finais e principalmente através de soluções incrementais.

Finalmente, os autores deixam um questionamento para o leitor, que trilhou o caminho da evolução da Internet através da leitura deste documento: "Se fosse possível começar de novo, como re-projetar a Internet, considerando as lições aprendidas e as perspectivas de uso que o futuro aponta?"

3.7. Referências

[1] Balakrishnan, H. *et al*., “A Layered Naming Architecture for the Internet”, ACM SIGCOMM, Setembro de 2004.

[2] Braden, R., Faber, T. & Handley, M., “From Protocol Stack to Protocol Heap - Role-Based Architecture”, Hotnets I, Outubro de 2002.

[3] Bradner, S., “IETF Working Group Guidelines and Procedures”, RFC 2418, Setembro de 1998.

[4] Bush, R. & Meyer, D., “Some Internet Architectural Guidelines and Philosophy”, RFC 3439, Dezembro de 2002.

[5] Carpenter, B. & Brim, S., “Middleboxes: Taxonomy and Issues”, RFC 3234, Fevereiro de 2002.

[6] Carpenter, B., “Architectural Principles of the Internet”, RFC 1958, Junho de 1996.

[7] Carpenter, B., “Internet Transparency”, RFC 2775, Fevereiro de 2000.

[8] Castineyra, I., Chiappa, N., & Steenstrup, M., “The Nimrod routing architecture”, RFC 1992, Agosto de 1996.

[9] Clark, D. & Tennenhouse, D., “Architectural Considerations for a New Generation of Protocols”. ACM SIGCOMM, Setembro de 1990.

[10] Clark, D., “The Design Philosophy of the DARPA Internet Protocols”, ACM SIGCOMM 88, Agosto. 1988.

[11] Clark, D., Braden, R., Falk, A., & Pingali, V., “FARA: Reorganizing the addressing architecture”, ACM SIGCOMM Workshop on Future Directions in Network Architecture (FNDA), Agosto de 2003.

[12] Clark, D., *et al.*, “New Arch: Future Generation Internet Architecture”, Final Technical Report, Dezembro de 2003, http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf.

[13] Crowcroft, J., Hand, S., Mortier, R., Roscoe, T. & Warfield, A., "Plutarch: an argument for network pluralism". Computer Communication Review 33(4): páginas 258-266, 2003.

[14] Eriksson, J., Faloutsos, M. & Srikanth, “PeerNet: Pushing Peer-to-Peer Down the Stack”, 2nd Intl Workshop on Peer-to-Peer Systems (IPTPS '03), Fevereiro de 2003.

[15] Fielding, R. *et al.*, “Hypertext Transfer Protocol -- HTTP/1.1”, RFC 2616, Junho de 1999.

[16] Francis, P. & Gummadi, R., “IPNL: A NAT-extended Internet architecture”, ACM SIGCOMM 2001, Agosto de 2001.

[17] Freed, N., “Behavior of and Requirements for Internet *Firewalls*”, RFC 2979, Outubro de 2000.

[18] Hain, T., “Architectural Implications of NAT”, RFC 2993, Novembro de 2000.

[19] Houle, K. & Weaver, G., “Trends in Denial of Service Attack Technology”, CERT Coordination Center, Outubro de 2001.

[20] ISO, ”Information Processing Systems - Open Systems Interconnection - Basic Reference Model” ISO 7498, 1984.

[21] Kent, S., & Atkinson, R., “Security Architecture for the Internet Protocol”, RFC 2401, Novembro de 1998.

[22] Labovitz, C., Ahuja, A., Bose, A. & Jahanian, F., “Delayed Internet Routing Convergence”, IEEE/ACM Transactions on Networking, 9(3), Junho de 2001.

[23] Marsan, C., “Faster 'Net growth rate raises fears about routers”, Network World, 04/02/01.

[24] Meyer, E., “ARPA Network Protocol Notes”, RFC 46, Abril de 1970.

[25] Perkins, C. *et al.*, “IP Mobility Support”, RFC 2002, Outubro de 1996.

[26] Roscoe, T., Hand, S., Isaacs, R., Mortier, R., & Jardetzky, P., "Predicate routing: Enabling controlled networking", 1st ACM Hotnets Workshop, Outubro de. 2002.

[27] Saltzer, J., Reed, D. & Clark, D., “End-to-End Arguments in System Design”. ACM Transactions in Computer Systems 2(4), páginas 277-288, Novembro de 1984.

[28] Shaikh, A., Varma, A., Kalampoukas, L., Dube, R., “Routing Stability in Congested Networks: Experimentation and Analysis”, ACM SIGCOMM, Agosto de 2000.

[29] Srisuresh, P. & Egevang, K., “Traditional IP Network Address Translator (Traditional NAT)”, RFC 3022, Janeiro 2001.

[30] Stoica, I., Adkins, D., Zhuang, S., Shenker, S. & Surana, S. “Internet indirection infrastructure”. SIGCOMM 2002, Agosto 2002.

[31] Walfish, M., *et al.*, “Middleboxes No Longer Considered Harmful”, USENIX OSDI 2004, Dezembro de 2004.

[32] Yang, X., “NIRA: A New Internet Routing Architecture”, ACM SIGCOMM Workshop on Future Directions in Network Architecture (FNDA), Agosto de 2003.

[33] Zhu, D., Gritter, M. & Cheriton, D., “Feedback Based Routing”, ACM SIGCOMM Computer Communication Review, 33(1), Fevereiro de 2003

[34] Internet World Stats, “Internet Usage Statistics - The Big Picture” http://www.internetworldstats.com/stats.htm, acessado em 14/03/2005.

[35] Shirey, R., “Internet Security Glossary”, RFC 2828, Maio de 2000.

[36] Howard, J. D. & Longstaff, T., “A Common Language for Computer Security Incidents”, Technical Report 98-8667, Sandia National Labs, Outubro de 1998.

[37] Anderson, T., & Roscoe T. & D. Wetherall, “Preventing Internet Denial-of-Service with Capabilities”, 2nd ACM Hotnets Workshop, Novembro de 2003.

[38] Patrikakis, C., Masikos, M., & Zouraraki, O., “Distributed Denial of Service Attacks”, The Internet Protocol Journal, 7(4), Dezembro de 2004.

[39] Cook, D., Morein, W., Keromytis, A., Misra, V. & Rubenstein, D., “WebSOS: Protecting Web Servers From DDoS Attacks”, IEEE International Conference on Networks (ICON), Setembro/Outubro de 2003.

[40] Dalal, M. (editor), “Transmission Control Protocol security considerations”, Internet-Draft, Novembro de 2004.

[41] Keromytis, A., Misra, V. & Rubenstein, D., “SOS: An Architecture For Mitigating DDoS Attacks”, IEEE Journal on Selected Areas in Communications (JSAC), Janeiro de 2004.

[42] Clark, D., Sollins, K., Wroclawski, J. & Faber, T., “Addressing reality: An architectural response to demands on the evolving Internet. ACM SIGCOMM Workshop on Future Directions in Network Architecture, Agosto de 2003.

[43] Ahlgren, B., Brunner, M, Eggert, L., Hancock, R. & Schmid, S., “Invariants – A New Design Methodology for Network Architectures”, ACM SIGCOMM Workshop on Future Directions in Network Architecture, Agosto de 2004.

[44] Willinger, W. & Doyle, J., "Robustness and the Internet: Design and evolution", 2002, http://netlab.caltech.edu/pub/papers/part1_vers4.pdf, acessada em 15/03/2005.

[45] Perkins, C. (Editor), “IP Mobility Support for IPv4”, RFC 3344. Agosto de 2002.

[46] Montenegro, G. (Editor), “Reverse Tunneling for Mobile IP, revised”. RFC 3024, Janeiro de 2001

[47] El Malki, K. (Editor), “Low Latency *Handoff*s in Mobile IPv4”, Internet Draft, Junho de 2004.

[48] Vaarala, S., & Klovning, E., “Mobile IPv4 Traversal Across IPsec-based VPN Gateways”, Internet Draft. Janeiro de 2005.

[49] Moskowitz, R., Nikander, P., Jokela, P. & Henderson, T. “Host Identity Protocol”, draft-ietf-hip-base-02, Internet-Draft, Fevereiro de 2005.

[50] Rocha, J., Domingues, M., Callado, A., Souto, E., Silvestre, G., Kamienski, C. & Sadok, D., “Peer-to-Peer: Computação Colaborativa na Internet”, 22º Simpósio Brasileiro de Redes de Computadores (SBRC 2004), Gramado/RS, Maio de 2004.

[51] Stoica, I., Morris, R., Karger, D. R., Kaashock, M. Frans & Balakrishman, H., “Chord: A scalable peer-to-peer lookup protocol for Internet applications”, ACM SIGCOMM 2001, Agosto de 2001.

[52] Walfish, M, Balakrishnan, H. ,& Shenker, S., "Untangling the Web from DNS". 1st Symp. on Networked Systems Design and Impl. (NSDI '04), Março de 2004.

[53] Johnson, D., Perkins, C., Arkko, J., "Mobility Support in IPv6". RFC 3775, Junho 2004.

[54] Calmon, R., “The Twelve Networking Truths”, RFC 1925, Abril de 1996.

[55] Leiner, B., *et al.*, “The Past and Future History of the Internet”, Communications of the ACM, 40(2), Fevereiro de 1997.