

Camada de Rede: Conceitos Básicos

Diego Passos

1 Camada de Rede: Responsabilidades, Serviços e Nomenclatura

A camada de rede é responsável pela comunicação entre dispositivos fim-a-fim. É importante destacar aqui — novamente — a diferença de responsabilidade desta camada para a camada de transporte: enquanto a primeira se preocupa com a comunicação entre **processos**, a segunda se preocupa com a comunicação entre **dispositivos finais**.

Na prática, esta diferença se manifesta nos tipos de serviços providos por estas duas camadas. Talvez o principal serviço provido pela camada de transporte, a demultiplexação identifica qual processo — ou mais especificamente, qual *socket* — deve receber o conteúdo de um pacote recém recebido pelo dispositivo. Por outro lado, as duas funções que estão no cerne da camada de rede — roteamento e encaminhamento — dizem respeito à movimentação dos pacotes pelos vários saltos que definem o caminho entre o dispositivo de origem e o de destino.

Assim como a camada de transporte faz com as mensagens da camada de aplicação, a camada de rede realiza o encapsulamento dos segmentos vindos da camada de transporte em **datagramas**. Portanto, no estudo desta camada utilizaremos extensivamente o termo *datagrama* para nos referirmos aos pacotes processados pela camada de rede.

Ao contrário da camada de transporte, no entanto — ao menos, em uma visão idealizada da Internet —, a camada de rede é implementada tanto nas bordas, quanto no núcleo da rede. Tanto os *hosts*, quanto os roteadores necessariamente implementam protocolos desta camada. Adicionalmente, os cabeçalhos da camada de rede são processados por cada roteador pelos quais passam no caminho até o dispositivo de destino — em oposição aos cabeçalhos das camadas superiores que, a princípio, deveriam apenas ser processados por origem e destino.

Os dois serviços chave providos por esta camada são fortemente interligados, porém claramente diferentes. O **encaminhamento** diz respeito à movimentação interna do pacote em um roteador. Em outras palavras, o encaminhamento consiste no processamento de um pacote que chega a um roteador, da decisão de por qual enlace de saída ele será transmitido e da posterior transmissão pelo enlace escolhido. O **roteamento**, por sua vez, diz respeito ao processo pelo qual roteadores coletam informações que baseiam o encaminhamento. Em outras palavras, o roteamento tem relação com a descoberta e escolha do caminho a ser percorrido por um datagrama, enquanto o encaminhamento é a “execução” deste caminho, através da movimentação do datagrama pelas interfaces dos roteadores.

Embora outras formas de interação entre estes serviços sejam possíveis, em geral, o roteamento coleta e processa informações sobre as rotas disponíveis na rede e as compila na forma de uma **tabela de roteamento**. O *software* responsável por isso é chamado de **protocolo de roteamento** que por sua vez utiliza algum tipo de **algoritmo de roteamento** para escolher a rota mais adequada dentre as várias disponíveis entre dois dispositivos.

Esta tabela de roteamento associa características de um datagrama recebido pelo roteador à porta de saída adequada. O termo genérico *características* foi usado aqui precisamente porque, a princípio, esta decisão poderia ser baseada em qualquer aspecto do datagrama. O mais comum, no entanto, é que esta característica seja o valor de um determinado campo do cabeçalho da camada de rede. Ao receber um datagrama por uma porta de entrada, o roteador obtém o valor deste campo, o utiliza para consultar a tabela e descobre a porta de saída que deve ser usada.

Embora não seja o caso da Internet, em certas redes um terceiro serviço executado por esta camada é o **estabelecimento de conexão**. Nestas redes, antes que os dados propriamente ditos de um fluxo possam ser enviados de um dispositivo para outro, é preciso realizar um processo de abertura de conexão, análogo ao que o TCP realiza com o seu *3-way handshake*. Entretanto, diferente do que ocorre no TCP, este processo de estabelecimento de conexão envolve todos os roteadores no caminho entre os dispositivos de origem e destino.

2 Camada de Rede: Modelos de Serviço

Outro conceito pertinente à camada de rede é o **modelo de serviço**. Diferentes redes proveem diferentes modelos de serviço e, em alguns casos, uma mesma rede pode oferecer múltiplos modelos de serviço diferentes — similar ao que a camada de transporte da Internet oferece, com seus dois protocolos tradicionais, TCP e UDP, fornecendo serviços diferentes às aplicações.

Modelos de serviço, portanto, dizem o que podemos esperar em relação ao desempenho da rede para **um datagrama específico ou para um fluxo de datagramas**. Por exemplo, para datagramas individuais, um certo modelo de serviço pode garantir a entrega — *i.e.*, o pacote nunca será descartado pela rede. Outro exemplo seria uma garantia de atraso máximo, *e.g.*, o datagrama será entregue em, no máximo, 50 ms. Já quando falamos de modelos de serviço para fluxos, podemos pensar em garantia de **entrega ordenada**, garantia de uma **vazão mínima** ou garantia de um **jitter máximo**.

Em uma tecnologia como a ATM, a rede oferece vários modelos de serviço diferentes, incluindo:

1. **CBR**. A taxa do fluxo será constante e a rede se compromete a garantir a taxa solicitada.
2. **VBR**. A taxa do fluxo será variável — dentro de determinados parâmetros previsíveis — e a rede garantirá a taxa solicitada.
3. **ABR**. O fluxo terá uma taxa mínima e, se houver disponibilidade de recursos da rede, esta poderá ser temporariamente aumentada. A rede garantirá a taxa mínima.
4. **UBR**. A taxa do fluxo é imprevisível e a rede não promete qualquer tipo de vazão mínima.

Além das garantias em termos de vazão mínima, estes modelos de serviço também podem incluir garantias da rede sobre entrega de pacotes (no caso de CBR e VBR), atraso máximo (CBR e VBR) e ordenação de pacotes (todos).

A Internet, por outro lado, utiliza na camada de rede um único modelo de serviço. Este modelo, ao contrário os exemplificados para as redes ATM, não faz promessas sobre o desempenho de fluxos ou de datagramas individuais. Este modelo é dito de **melhor esforço**: o desempenho na transmissão dos datagramas é condicionado às condições momentâneas da rede. Em outras palavras, na Internet não há garantias de vazão mínima, atraso máximo, *jitter* máximo ou mesmo de que os datagramas serão efetivamente entregues.

3 Camada de Rede: Redes de Circuitos Virtuais

O termo *datagrama*, usado para denotar os pacotes na camada de rede na Internet, implica algumas características básicas do funcionamento desta rede, a saber:

- **Datagramas processados de maneira independente**. Na Internet (ao menos idealmente), roteadores não guardam estado sobre fluxos de pacotes ativos. O processamento de um datagrama em um roteador intermediário, portanto, é realizado de forma completamente independente de outros pacotes anteriores encaminhados pelo mesmo roteador.
- **Serviço não-orientado a conexão**. Dado que roteadores não guardam estado sobre fluxos de dados ativos, não existe o conceito de conexão, do ponto de vista da camada de rede.

Diz-se, portanto, que a Internet é uma **Rede de Datagramas**.

Uma alternativa a este paradigma são as redes baseadas em **Circuitos Virtuais**. Em oposição às redes de datagramas, as redes baseadas em circuitos virtuais proveem um serviço orientado a conexão, em que cada pacote é encaminhado por um comutador de acordo com regras que podem ser específicas ao fluxo de dados ao qual este pertence.

É importante não confundir: **redes baseadas em circuitos virtuais não são a mesma coisa que redes de comutação de circuitos**. Assim como as redes de datagramas, uma rede baseada em circuito virtual utiliza o paradigma da comutação de pacotes. O circuito, ao qual se refere o nome, diz respeito simplesmente ao estabelecimento de uma conexão entre os dispositivos de origem e destino de cada fluxo.

De certa forma, a oposição entre as redes de datagramas e de circuitos virtuais é análoga à oposição entre UDP e TCP na camada de transporte — um protocolo provê um serviço orientado à conexão, enquanto o outro não. No entanto, como estamos falando da camada de rede, as aplicações não têm escolha: a rede provê um ou outro serviço, e os *hosts* são obrigados a utilizá-lo.

Voltando aos circuitos virtuais, a ideia é que eles emulem o funcionamento de um circuito (como na comutação de circuitos), mas utilizando a comutação de pacotes. O primeiro passo é o estabelecimento

do circuito. Durante esta fase, recursos relativos ao circuito são alocados em cada roteador pelo qual o fluxo de dados passará. Embora isso não seja obrigatório, muitas vezes estes recursos podem incluir *buffers* nos roteadores e tempo reservado de transmissão em certos enlaces, fornecendo, assim, uma previsibilidade do desempenho que o fluxo receberá.

Independentemente da reserva ou não de *buffers* ou banda, um circuito virtual deve consistir minimamente de:

1. **Um caminho entre origem e destino.** Isto é, uma sequência de roteadores intermediários pelos quais **todos os pacotes do fluxo passarão**. Isso significa que não há a possibilidade de pacotes diferentes de um mesmo fluxo seguirem caminhos distintos.
2. **Números de identificação.** Também chamado de *número do circuito virtual*, este valor é inserido no cabeçalho de cada pacote de um fluxo e permite a cada roteador identificar facilmente a qual circuito virtual aquele pacote pertence. Note que um circuito virtual pode ser identificado por vários valores diferentes ao longo do caminho: um para cada enlace. Em outras palavras, o identificador do circuito virtual pode mudar a cada salto.
3. **Entradas nas tabelas de roteamento.** Cada roteador precisa armazenar na sua tabela de roteamento uma associação entre cada circuito virtual que passa por ele e a ação adequada ao encaminhamento do pacote, *e.g.*, a porta pela qual os pacotes daquele circuito virtual devem ser encaminhados.

Todos estes três elementos são alocados/escolhidos no momento do estabelecimento do circuito virtual (*i.e.*, a abertura da conexão). Da mesma maneira, cada roteador precisa manter as informações da sua tabela de roteamento enquanto perdurar a conexão. Isso significa que, **em redes baseadas em circuitos virtuais, roteadores precisam armazenar estado dos fluxos ativos**.

O estabelecimento de um circuito virtual demanda um protocolo específico, geralmente denominado um **Protocolo de Sinalização**. Estes protocolos iniciam o processo de estabelecimento do circuito a partir do dispositivo de origem, que envia uma requisição de estabelecimento para o roteador ao qual está conectado, informando o destinatário desejado — e, possivelmente, requisitos de desempenho do fluxo. O núcleo da rede, então, se encarrega de escolher um caminho e alocar os recursos necessários ao longo do circuito — *e.g.*, números de circuito virtual, entradas nas tabelas de roteamento, espaço em *buffer*, tempo de uso dos enlaces. Repare que neste tipo de rede é possível que, no momento da abertura de uma conexão, não haja recursos suficientes disponíveis. Neste caso, os protocolos de sinalização geralmente incluem suporte para que o núcleo da rede avise ao dispositivo de origem sobre esta situação — o análogo a um sinal de ocupado em uma ligação telefônica.

4 Camada de Rede: Redes de Datagramas e Tabelas de Roteamento

Como em uma rede de datagramas não há estabelecimento de um circuito virtual, a informação usada pelos roteadores para realizar a decisão de encaminhamento — *i.e.*, por qual porta de saída encaminhar o pacote — precisa ser diferente. Neste tipo de rede, esta decisão se baseia, majoritariamente, no endereço do dispositivo de destino (que, portanto, precisa constar no cabeçalho do pacote).

Desta forma, em um dado instante de tempo, dois datagramas com um mesmo destinatário são normalmente encaminhados pela mesma porta de saída de um roteador. Por outro lado, as decisões de roteamento podem mudar dinamicamente, fazendo com que dois datagramas de um mesmo fluxo de dados passando por um dado roteador em momentos diferentes possam ser encaminhados por caminhos diferentes. Entre outras consequências, isso pode resultar na reordenação de pacotes, *i.e.*, um pacote gerado posteriormente pode ser entregue ao destinatário antes de um pacote gerado anteriormente.

Como, normalmente, as decisões de encaminhamento são tomadas com base no endereço do destinatário, a tabela de roteamento em uma rede de datagramas possui duas colunas principais:

1. **Endereço de destino.**
2. **Porta de saída.**

Uma preocupação recorrente em uma rede da escala da Internet é com o tamanho das tabelas de roteamento. Embora ainda não tenhamos discutido o protocolo IP, adiantaremos aqui que, na sua versão comumente usada na Internet atual, os endereços dos dispositivos têm comprimento de 32 bits. Isso significa que, a princípio, há $2^{32} = 4294967296$ de endereços possíveis. Se cada roteador precisasse manter em sua tabela de roteamento uma entrada para cada um destes mais de 4 bilhões de endereços,

haveria um sério problema de escalabilidade, em termos de capacidade necessária de armazenamento — faça as contas: ainda que cada entrada da tabela ocupasse apenas 1 byte, seriam necessários 4 GB de memória para armazená-la.

A saída para este problema é a agregação de endereços em faixas. Ao invés associar um endereço individual a uma porta de saída, as tabelas de roteamento associam faixas de vários endereços subsequentes a uma porta de saída comum. Desta maneira, se na maioria das vezes endereços numericamente próximos tiverem uma mesma porta de saída em comum, este processo de agregação será eficiente em reduzir o tamanho das tabelas de roteamento, melhorando este aspecto de escalabilidade na Internet. De fato, a Internet utiliza um mecanismo de **endereçamento hierárquico** (tópico de aulas futuras) que visa, justamente, este tipo de agregação.

Para evitar ter que armazenar dois endereços por entrada da tabela — um endereço do início da faixa e outro do final —, na prática, tabelas de roteamento geralmente denotam as faixas apenas por **prefixos**. Se escrevermos os endereços de início e final de uma faixa na sua representação binária, podemos pegar os bits mais à esquerda em comum nos dois endereços — isto é, o prefixo — e usá-los para denotar a faixa como um todo. Repare que isso assume que as faixas sempre são da forma $XX \dots XX00 \dots 00$ até $XX \dots XX11 \dots 11$, onde $XX \dots XX$ denotam os bits do prefixo. Em outras palavras, o número de endereços compreendidos em uma faixa é sempre uma potência de 2, embora faixas diferentes possam ter tamanhos diferentes.

Outra característica interessante da maneira pela qual tabelas de roteamento são tratadas na Internet é a ideia de **casamento por prefixo mais longo**. Suponha que um roteador possua duas entradas na sua tabela de roteamento, uma associando o prefixo 0101 à porta de saída 0 e associando o prefixo 01011 à porta de saída 1. Note que os endereços compreendidos na faixa definida pelo segundo prefixo também fazem parte da faixa definida pelo primeiro prefixo. Se o roteador consulta a tabela para decidir por onde encaminhar um datagrama destinado ao endereço 010111..., qual das duas entradas deve ser utilizada? Aplicando o conceito de casamento por prefixo mais longo, escolheremos a segunda entrada por ela possuir, como o nome sugere, um prefixo mais longo. Comumente, dizemos esta entrada da tabela de roteamento é *mais específica*. Esta convenção permite que definamos uma porta de saída única para uma grande faixa de endereços contíguos, mas considerando algumas exceções para faixas menores que precisem ser encaminhadas por portas diferentes.

O casamento por prefixo mais longo também dá origem a uma entrada bastante comum em tabelas de roteamento na Internet, conhecida como **rota padrão**. Esta entrada tem um prefixo de comprimento 0 que, portanto, casa com qualquer endereço de destino — porque os “zero primeiros bits” de qualquer endereço sempre casam com um prefixo nulo. Esta regra funciona como um coringa: se não houver uma entrada específica na tabela de roteamento dizendo por onde encaminhar um dado pacote, a rota padrão será usada.

5 Camada de Rede: Circuitos Virtuais *vs.* Datagramas

Uma vez definidos os princípios básicos de funcionamento das redes de circuitos virtuais e das de datagramas, podemos fazer uma comparação das duas abordagens.

Como em praticamente todos os casos de abordagens alternativas estudadas nesta disciplina, não existe um “vencedor” definitivo entre estes dois paradigmas. Ao contrário, existe um conjunto de compromissos que, uma vez considerados em um cenário específico, ajudam a decidir qual paradigma é mais adequado àquela situação.

As redes de circuitos virtuais têm maior capacidade de oferecer modelos de serviço mais sofisticados, já que o processo de estabelecimento do circuito virtual permite a reserva de recursos ao longo do caminho escolhido. Além disso, por alocar um circuito virtual (e, portanto, um caminho fim-a-fim) estático ao longo da duração do fluxo de dados, estas redes impedem problemas como a reordenação de pacotes. Todas estas características permitem que as redes de circuitos virtuais sejam boas alternativas para aplicações que demandam desempenho previsível e/ou garantido.

Já as redes de datagramas não fazem, a princípio, qualquer reserva de recursos. Em condições normais, nem mesmo pode-se garantir que o mesmo caminho será usado por todos os datagramas de um fluxo. Isso torna a previsibilidade do desempenho bem mais baixa. Para os chamados **serviços elásticos** (como a transferência de um arquivo), isso não é problema: eles se adaptam ao que a rede puder entregar de desempenho naquele momento. Dado que mecanismos de transferência confiável de dados, como os fornecidos pelo TCP, sejam utilizados, estas aplicações funcionam.

Embora possam dar garantias de desempenho, as redes de circuitos virtuais acabam gerando uma maior complexidade no núcleo. Roteadores precisam armazenar estado e dar suporte a reserva de recursos e protocolos de sinalização. Neste sentido, as redes de datagramas são bem mais simples, delegando a maior complexidade às bordas (*i.e.*, sistemas finais). Esta simplicidade do núcleo também

permite uma escalabilidade maior, o que é fundamental em uma rede com a escala da Internet.

Além da simplificação do núcleo — e, talvez, uma razão até mais forte para isso —, a Internet adotou o paradigma de datagramas por conta do seu objetivo histórico de interligar redes diferentes. O termo *diferentes*, aqui, pode assumir vários significados, incluindo a existência de tecnologias bastante variadas de enlaces. Dada esta heterogeneidade de enlaces, seria muito difícil prover as garantias que as redes baseadas em circuitos virtuais geralmente proveem. Colocando de outra forma, é fácil emular um serviço de melhor esforço sobre uma rede com garantias de desempenho, mas o inverso não é verdade. Neste sentido, uma Internet baseada em circuitos virtuais demandaria uma série de requisitos mínimos das tecnologias dos enlaces que a compõem. Assim, uma rede de datagramas permitiu uma Internet mais “democrática”, abrangendo uma gama enorme de tecnologias distintas.