Parte 2
IP e Roteamento

1.9 – Qualidade de Serviço (QoS)

1.9.1 – Requisitos

- ***Fluxo*** → é uma sequência de pacotes desde uma origem até um destino
- *Serviço orientado a conexões na camada de rede* → todos os pacotes que pertencem a um fluxo seguem a mesma rota
- *Serviço não orientado a conexões* → os pacotes podem seguir rotas diferentes
- As necessidades de cada fluxo podem ser caracterizadas por quatro parâmetros principais:
  - confiabilidade
  - retardo
  - flutuação
  - largura de banda

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

1.9.2 – Técnicas para se alcançar boa qualidade de serviço

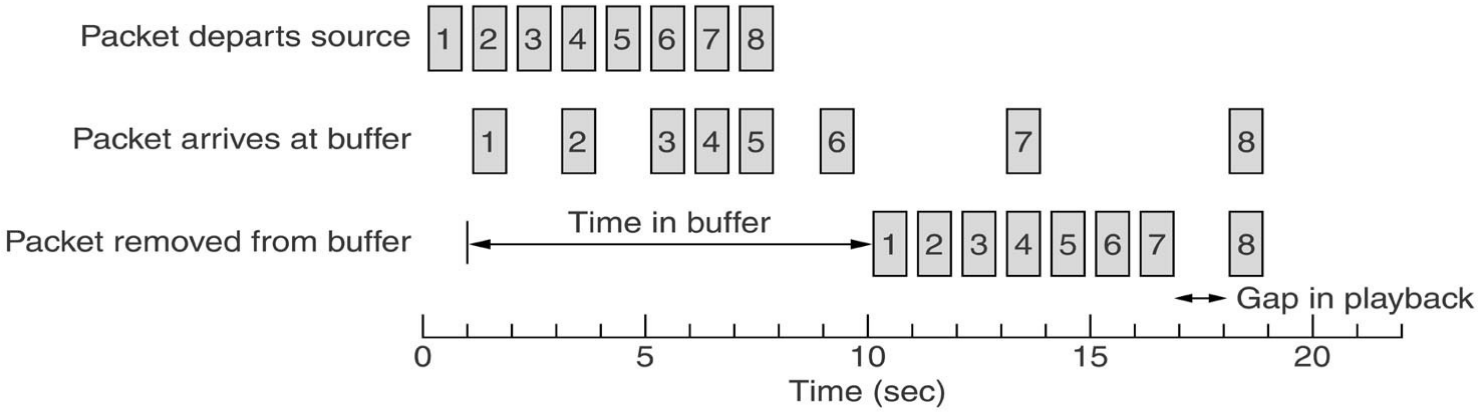
- ✔ Como alcançar a qualidade de serviço?
- ✔ Não há uma fórmula mágica
- ✔ Nenhuma técnica isolada proporciona QoS eficiente e seguro de forma ótima
- ✔ Foram desenvolvidas diversas técnicas, e as soluções práticas muitas vezes combinam várias dessas técnicas

Superdimensionamento

- ✔ Uma solução prática é fornecer muita capacidade de roteadores, com muito espaço de buffer e muita largura de banda
  - os pacotes serão transmitidos com muita facilidade
  - o problema com essa solução é seu custo
  - e.g.: o sistema de telefonia é superdimensionado

Armazenamento em buffer

- Os fluxos podem ser armazenados em buffer no lado receptor, antes de serem entregues
  - não afeta a confiabilidade ou a largura de banda
  - aumenta o retardo
  - suaviza a flutuação
- No caso de áudio e vídeo por demanda, a flutuação é o principal problema e, portanto, essa técnica ajuda bastante



Moldagem de tráfego

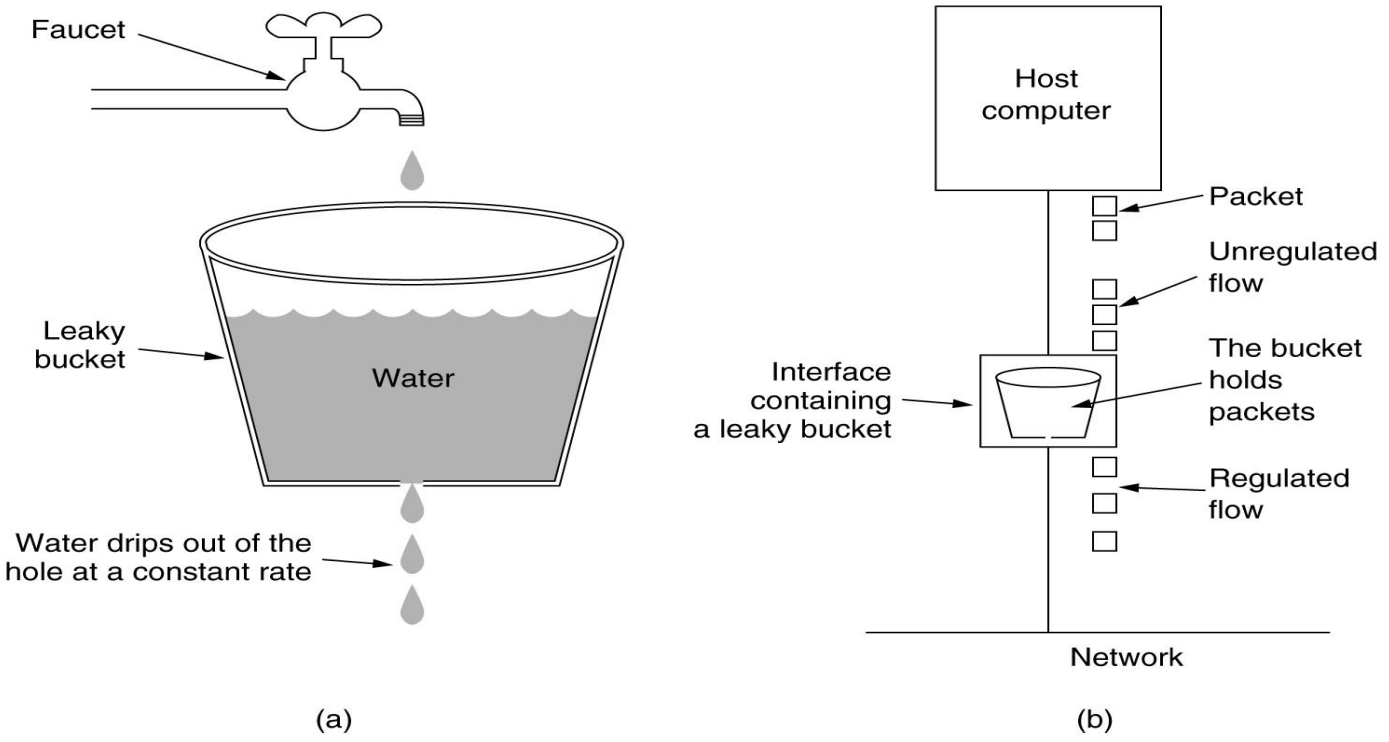
- No exemplo anterior, a origem transmite os pacotes com um espaçamento uniforme entre eles mas, em outros casos, é possível que eles sejam transmitidos de modo irregular, o que pode causar congestionamentos na rede
- A saída não uniforme é comum:
  - se o servidor está manipulando muitos fluxos ao mesmo tempo
  - avanço e retrocesso rápidos
  - autenticação de usuários, etc
- O armazenamento em buffer nem sempre é possível, por exemplo, com videoconferência
- Se fosse feito algo para obrigar o servidor (e os hosts em geral) a transmitir a uma velocidade uniforme, a qualidade de serviços seria melhor
- A técnica denominada **moldagem de tráfego** suaviza o tráfego no lado servidor, e não no lado cliente

- ✔ A moldagem de tráfego está relacionada à regulagem da taxa média (e do volume) da transmissão de dados
- ✔ Quando uma conexão é configurada, o usuário e a sub-rede concordam com um determinado padrão de tráfego (ou seja, um formato) para esse circuito
  - geralmente, esse acordo é chamado **acordo de nível de serviço**
  - se o cliente cumpre sua parte no negócio e envia somente pacotes que estejam de acordo com o contrato, a concessionária de comunicações promete entregá-los pontualmente
  - a moldagem de tráfego reduz o congestionamento e ajuda a concessionária a cumprir sua promessa
  - esse acordo não é muito importante para transferências de arquivos, mas é de grande importância no caso da transmissão de dados em tempo real
    - e.g: conexões de áudio e vídeo

- ✔ Como a concessionária poderá saber se o cliente está cumprindo o acordo e o que fazer em caso negativo?
- ✔ O monitoramento de um fluxo de tráfego é chamado **controle de tráfego**
- ✔ Concordar com um padrão de tráfego e controlá-lo daí em diante é mais fácil com sub-redes de circuitos virtuais do que com sub-redes de datagramas.
- ✔ Mesmo com sub-redes de datagramas, as mesmas ideias podem se aplicar a conexões da camada de transporte

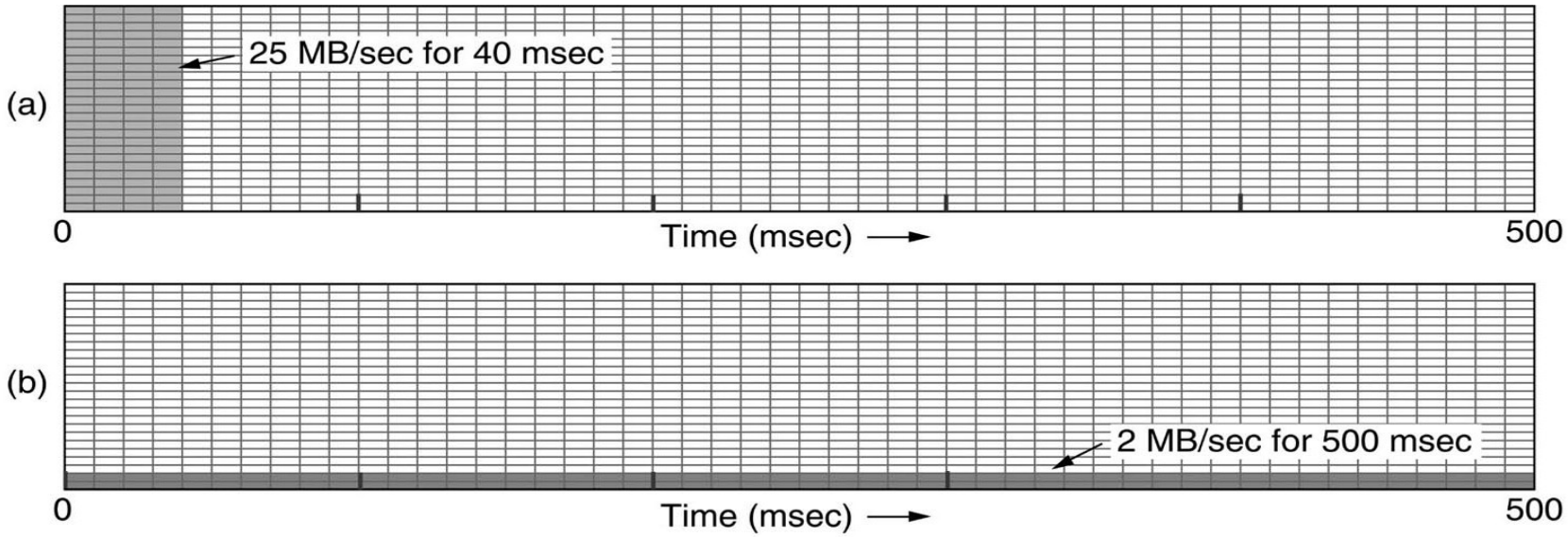
O algoritmo do balde furado

- Seja um balde com um pequeno furo no fundo Independente da velocidade com que a água entra no balde, o fluxo de saída ocorrerá em uma taxa (praticamente) constante
- Quando o balde estiver cheio, a água que entrar escorrerá pelas bordas e se perderá (ou seja, não aparecerá no fluxo de saída sob o furo)
- A mesma ideia pode ser aplicada a pacotes
  - cada host estaria conectado à rede por uma interface que contém um balde furado, com uma fila interna finita
  - se um pacote chegar à fila quando ela estiver cheia, o pacote será descartado sem cerimônia


Faucet
Leaky bucket
Water
Water drips out of the hole at a constant rate
(a)
Host computer
Packet
Unregulated flow
Interface containing a leaky bucket
The bucket holds packets
Regulated flow
Network
(b)

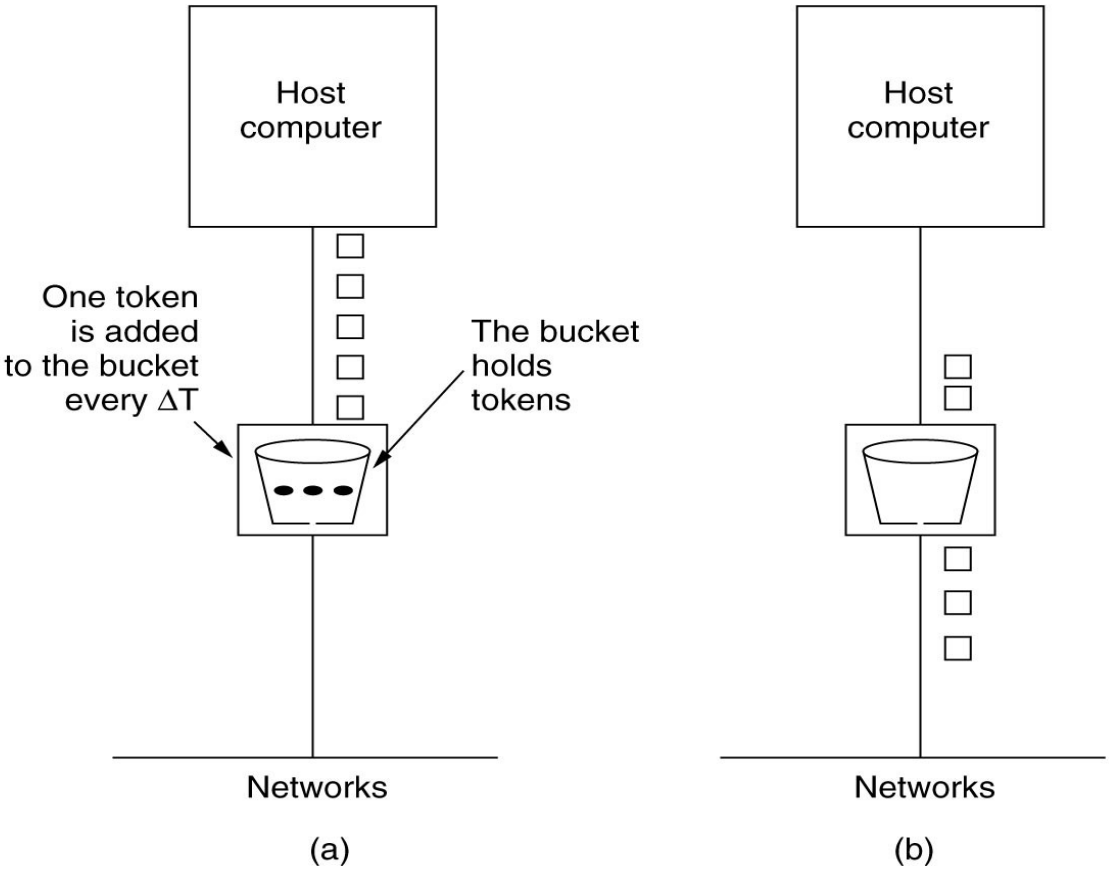
- Essa disposição pode ser interna à interface de hardware ou simulada pelo sistema operacional do host
  - trata-se simplesmente de um sistema de enfileiramento com um tempo de serviço constante
- O host pode inserir um pacote na rede a cada pulso de clock
  - isso pode ser forçado pela placa de interface ou pelo sistema operacional
- Esse mecanismo transforma um fluxo de pacotes irregular, proveniente dos processos do usuário no host, em um fluxo de pacotes regular para a rede, suavizando as rajadas e reduzindo bastante as possibilidades de congestionamento
- Quando estão sendo utilizados pacotes de tamanho variável, a melhor opção é permitir um número fixo de bytes por pulso, em vez de apenas um pacote
  - e.g.: se a regra for 1024 bytes por pulso, é possível admitir em um pulso um único pacote de 1024 bytes, ou dois pacotes de 512 bytes, ou quatro pacotes de 256 bytes, etc

- **Algoritmo de balde furado original:**
  - o balde furado consiste em uma fila finita
  - ao chegar um pacote
    - se houver espaço na fila, ele será incluído na fila
    - caso contrário, ele será descartado
  - a cada pulso do clock, um pacote é transmitido (a menos que a fila esteja vazia)
- **Algoritmo do balde furado de contagem de bytes**
  - a cada pulso, um contador é inicializado em ***n***
  - se o primeiro pacote da fila tiver menos bytes que o valor atual do contador, ele será transmitido, e o contador será decrementado por esse número de bytes
  - é possível enviar pacotes adicionais, desde que o contador tenha um valor suficientemente alto
  - quando o contador fica abaixo do comprimento do próximo pacote na fila, a transmissão é interrompida até o pulso seguinte
    - então a contagem de bytes residuais é reinicializada e o fluxo pode continuar


(a)
25 MB/sec for 40 msec
0
Time (msec)
500
(b)
2 MB/sec for 500 msec
0
Time (msec)
500

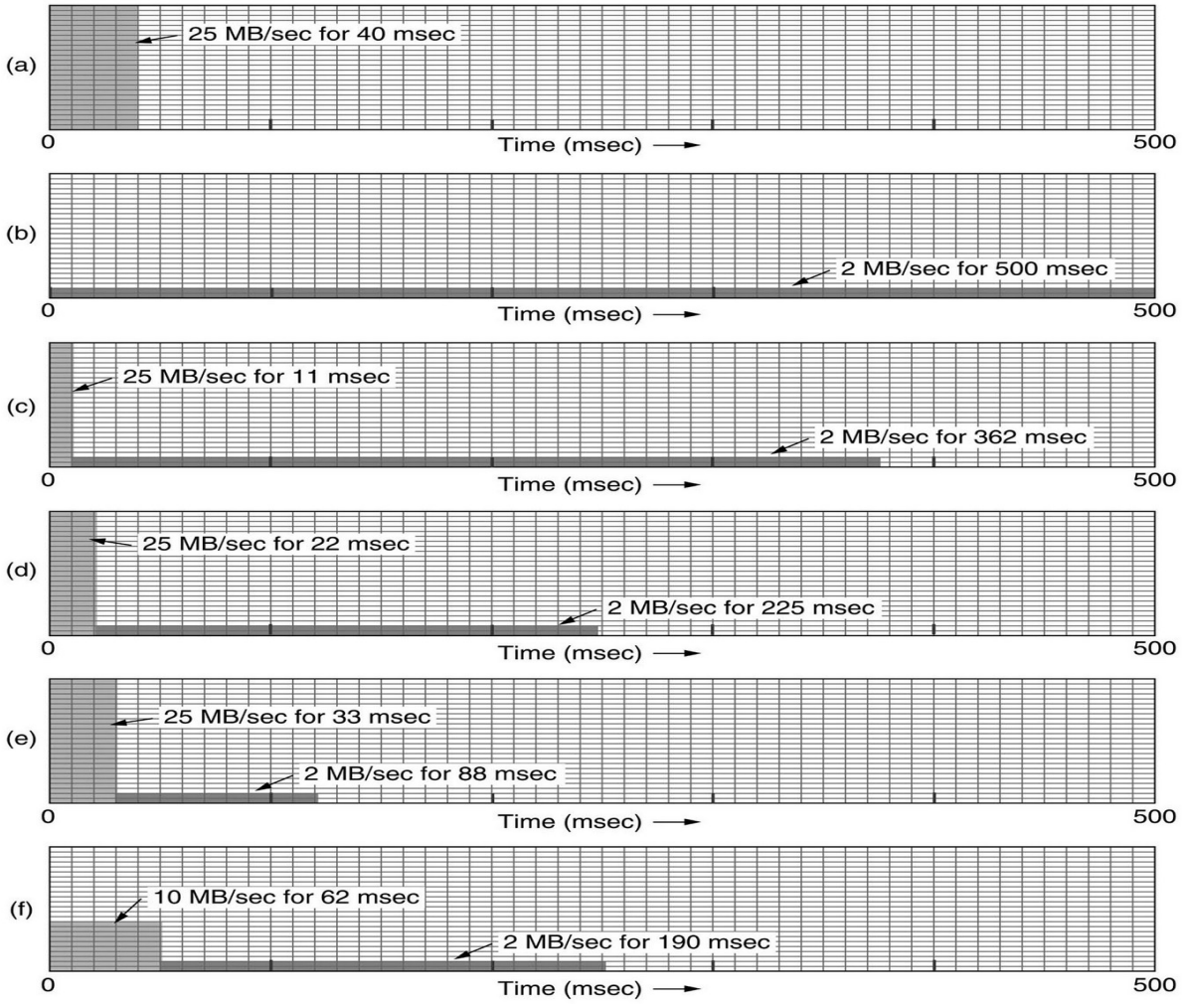
Algoritmo de Balde de Tokens

- ✔ O algoritmo de balde furado impõe um padrão de saída rígido à taxa média, independente da irregularidade do tráfego
- ✔ Em muitas aplicações, é melhor permitir que a saída aumente um pouco sua velocidade quando chegarem rajadas maiores → necessário um algoritmo mais flexível
- ✔ Algoritmo de Balde de Tokens → Nesse algoritmo, o balde furado retém tokens gerados por um clock na velocidade de um token a cada **ΔT** segundos


Host computer
One token is added to the bucket every ΔT
The bucket holds tokens
Networks
(a)
Host computer
Networks
(b)

- ✔ O algoritmo de balde de tokens possibilita um tipo de moldagem de tráfego diferente do algoritmo de balde furado
  - o algoritmo de balde furado não deixa que hosts inativos poupem permissões para enviar rajadas maiores posteriormente
  - o algoritmo de balde de tokens permite economia, até o tamanho máximo do balde, **n**
    - significa que rajadas de até **n** pacotes podem ser enviadas simultaneamente
- ✔ O algoritmo de balde de tokens joga tokens fora (isto é, capacidade de transmissão) quando o balde enche
- ✔ O algoritmo de balde furado descarta pacotes quando o balde fica cheio
- ✔ Variação do algoritmo de balde de tokens → cada token representa o direito de enviar não um pacote, mas **k** bytes
- ✔ um pacote só poderá ser transmitido se houver tokens suficientes disponíveis para abranger seu comprimento em bytes

- ✔ Os algoritmos de balde furado e de balde de tokens também podem ser usados para atenuar o tráfego entre roteadores
- ✔ Podem ser usados também para controlar a saída do host
- ✔ O algoritmo de balde de tokens utiliza uma variável que conta tokens
  - o contador é incrementado em uma unidade a cada intervalo de tempo **ΔT** e é decrementado de uma unidade sempre que um pacote é enviado
    - quando o contador atinge zero, nenhum pacote pode ser enviado
  - na variante de contagem de bytes, o contador é incrementado em **k** bytes a cada **ΔT** e decrementado de acordo com o comprimento de cada pacote enviado
- ✔ Basicamente, o que o algoritmo de balde de tokens faz é permitir rajadas, mas apenas até uma duração máxima controlada


(a)
25 MB/sec for 40 msec
0
Time (msec)
500
(b)
2 MB/sec for 500 msec
0
Time (msec)
500
(c)
25 MB/sec for 11 msec
2 MB/sec for 362 msec
0
Time (msec)
500
(d)
25 MB/sec for 22 msec
2 MB/sec for 225 msec
0
Time (msec)
500
(e)
25 MB/sec for 33 msec
2 MB/sec for 88 msec
0
Time (msec)
500
(f)
10 MB/sec for 62 msec
2 MB/sec for 190 msec
0
Time (msec)
500

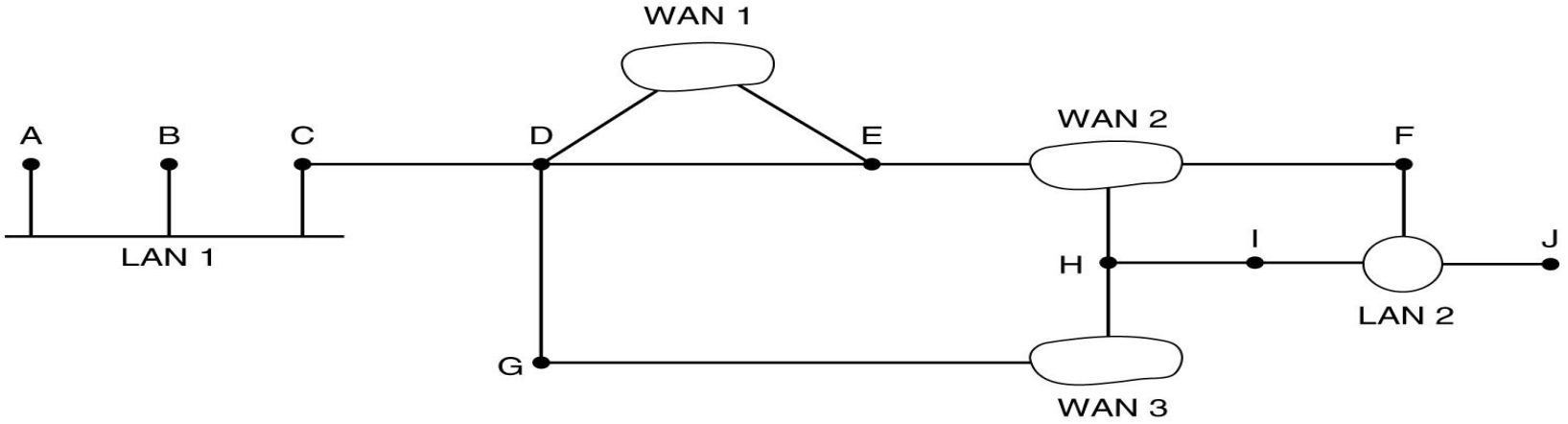
1.10 – OSPF – Interior Gateway Routing Protocol

- A Internet é formada por muitos sistemas autônomos (SA)
- Cada SA é operado por uma organização diferente (autoridade)
- Cada SA pode usar seu próprio algoritmo de roteamento interno
  - e.g.: as redes internas das empresas X, Y e Z em geral serão vistas como três SAs
  - internamente, todas três podem usar algoritmos de roteamento específicos (diferentes para cada uma)
- O fato de haver padrões, mesmo para roteamento interno, simplifica a implementação de fronteiras entre os SAs
- Um algoritmo de roteamento em um SA é chamado **protocolo de gateway interior**
- Um algoritmo para roteamento entre SAs é chamado de **protocolo de gateway exterior**

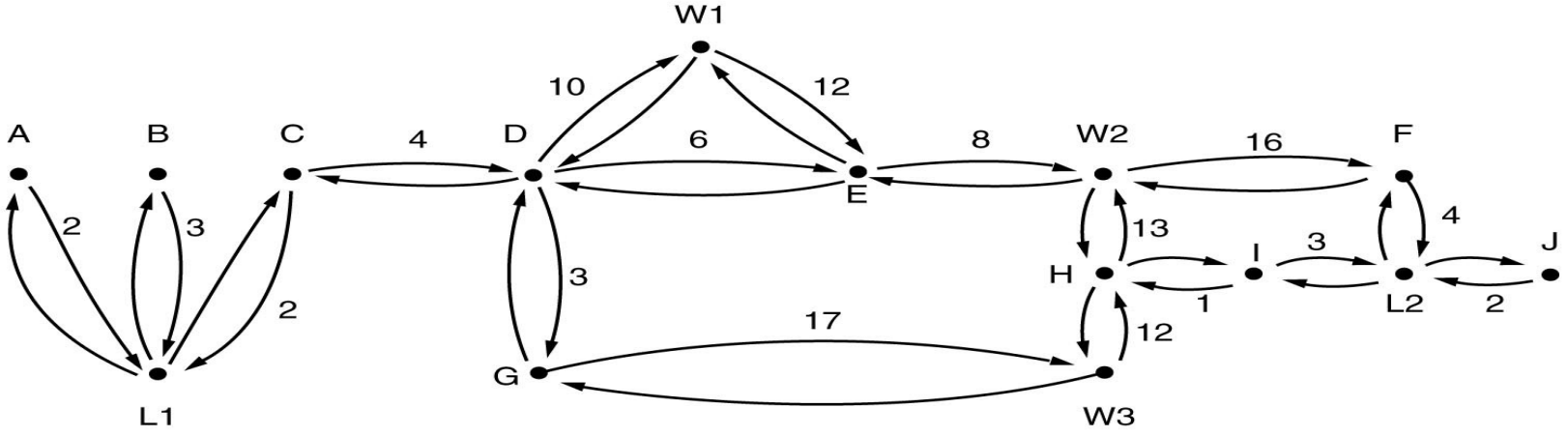
- ✔ O protocolo de gateway interior da Internet original era um protocolo de vetor de distância (RIP)
- ✔ Funcionava bem em sistemas pequenos
- ✔ Sofria do problema da contagem até infinito e convergia lentamente
- ✔ Em 1988, o IETF começou a desenvolver seu sucessor – (**OSPF – *Open Shortest Path First***), que se tornou um padrão em 1990 – RFC 2328
- ✔ Bem aceito por fabricantes de roteadores, tornou-se o principal protocolo de gateway interior

- Lista de requisitos que deveriam ser atendidos:
  - 1 - O protocolo teria de ser amplamente divulgado na literatura especializada. Uma solução patenteada de uma única empresa não funcionaria nesse caso
  - 2 - O novo protocolo teria de admitir uma variedade de unidades de medida de distância (peso)
  - 3 - Ele teria de ser um algoritmo dinâmico, que se adaptasse de forma rápida e automática a alterações na topologia
  - 4 - Ele tinha que admitir o roteamento baseado no tipo de serviço
    - o novo protocolo deveria ser capaz de rotear tráfego de tempo real de uma determinada maneira e outro tipo de tráfego de maneira diferente
  - 5 - O novo protocolo tinha que balancear a carga, dividindo-a por várias linhas
  - 6 - Era necessário o suporte para sistemas hierárquicos
    - nenhum roteador deveria ser obrigado a conhecer a topologia da rede inteira
  - 7 – Nível mínimo de segurança
  - 8 – Capacidade para lidar com outros roteadores através de túnel

- O OSPF é compatível com três tipos de conexões e redes:
  - 1. Linhas ponto a ponto entre, exatamente, dois roteadores
  - 2. Redes de multiacesso com difusão (por exemplo, a maioria das LANs)
  - 3. Redes de multiacesso sem difusão (por exemplo, a maioria das WANs)
- ***Rede de Multiacesso***: é aquela que pode ter vários roteadores e cada um dos quais pode se comunicar diretamente com todos os outros
- O OSPF funciona transformando o conjunto de redes, roteadores e linhas reais em um grafo orientado
- O OSPF calcula o caminho mais curto com base nos pesos dos arcos
- Uma conexão entre dois roteadores é representada por um par de arcos, um em cada sentido
  - seus pesos podem ser diferentes


WAN 1
A
B
C
D
E
WAN 2
F
LAN 1
H
I
J
LAN 2
G
WAN 3

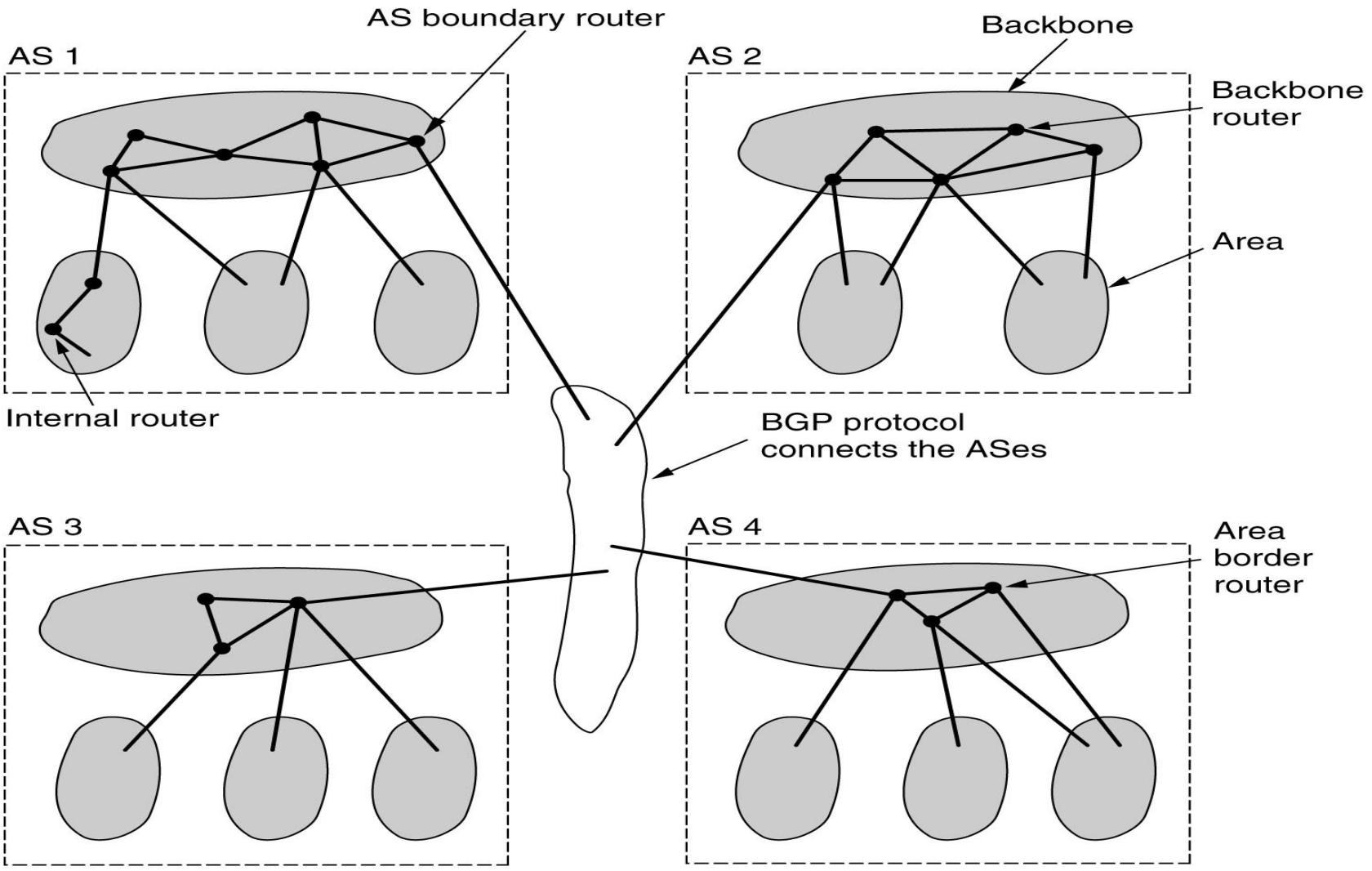
(a)


W1
10
12
A
B
C
4
D
6
8
W2
16
F
E
2
3
13
4
I
3
J
H
3
2
1
L2
2
17
12
G
L1
W3

(b)

- ✔ Muitos dos SAs da Internet são grandes e difíceis de gerenciar
  - o OSPF permite que eles sejam divididos em áreas
  - uma área é uma rede ou um conjunto de redes contíguas
    - essas áreas não se sobrepõem, mas não precisam ser completas
    - talvez alguns roteadores não pertençam a nenhuma área
  - uma área é uma generalização de uma sub-rede
    - fora de uma área, a topologia e os detalhes da sub-rede não são visíveis
- ✔ Cada SA possui uma área de backbone, chamada área 0
- ✔ Todas as áreas estão conectadas ao backbone
  - pode-se ir de uma área do SA para qualquer outra via backbone
- ✔ Cada roteador conectado a duas ou mais áreas faz parte do backbone
- ✔ Como em outras áreas, a topologia do backbone não pode ser vista fora dele

- Em uma área, cada roteador tem o mesmo banco de dados de estado de enlace e utiliza o mesmo algoritmo de caminho mais curto
- Um roteador que se conecta a duas áreas precisa dos bancos de dados de ambas as áreas e deve utilizar o algoritmo de caminho mais curto em cada uma delas separadamente
- Durante a operação normal, talvez sejam necessários três tipos de rotas:
  - 1 - entre áreas;
    - sempre acontece em três etapas:
    - a - vai da origem para o backbone;
    - b - atravessa o backbone até a área de destino;
    - c - vai até o destino
    - esse algoritmo força uma configuração em estrela no OSPF, com o backbone sendo o hub e as outras áreas sendo os raios
  - 2 - na mesma área;
    - mais fácil, o roteador de origem já conhece o caminho mais curto para o roteador de destino
  - 3 - entre sistemas autônomos


AS boundary router
Backbone
AS 1
AS 2
Backbone router
Area
Internal router
BGP protocol connects the ASes
AS 3
AS 4
Area border router

- O OSPF distingue quatro classes de roteadores:
  - 1 - os roteadores internos, que ficam inteiramente em uma área
  - 2 - os roteadores de borda de área, que conectam duas ou mais áreas
  - 3 - os roteadores de backbone, que ficam no backbone
  - 4 - os roteadores de fronteira do SA, que interagem com roteadores de outros SAs
- Essas classes podem se sobrepor:
  - e.g.: todos os roteadores de borda fazem parte do backbone automaticamente
  - e.g.: um roteador que esteja no backbone, mas que não faça parte de nenhuma outra área, também é um roteador interno

- ✔ Não é útil fazer cada roteador de uma LAN se comunicar com todos os outros roteadores da mesma LAN
- ✔ Para evitar essa situação, um roteador é eleito como o roteador **designado** ou **adjacente**
- ✔ Ele troca informações com todos os outros roteadores em sua LAN
- ✔ Roteadores vizinhos que não são adjacentes não trocam informações entre si
- ✔ Um roteador adjacente de reserva é sempre mantido atualizado, a fim de facilitar a transição, caso o roteador designado principal venha a falhar

- Quando é inicializado, um roteador envia mensagens ***HELLO*** por todas as suas linhas ponto a ponto, transmitindo-as por difusão nas LANs até o grupo que consiste em todos os outros roteadores
- Nas WANs, o roteador necessita de algumas informações de configuração para saber quem contactar
- A partir das respostas, cada roteador descobre quem são seus vizinhos
- Os roteadores da mesma LAN são todos vizinhos

- ✔ Durante a operação normal, cada roteador emite periodicamente por inundação mensagens **LINK STATE UPDATE** para cada um de seus roteadores adjacentes.
- ✔ Essa mensagem informa seu estado e fornece os custos usados no banco de dados da topologia
- ✔ As mensagens enviadas são confirmadas através de mensagens **LINK STATE ACK**, a fim de torná-las confiáveis
- ✔ Cada mensagem tem um número de sequência, e assim o roteador pode ver se uma mensagem **LINK STATE UPDATE** recebida é mais antiga ou mais recente do que a atual
- ✔ Os roteadores também enviam essas mensagens quando uma linha é ativada ou desativada, ou quando seus custos se alteram

- As mensagens **DATABASE DESCRIPTION** fornecem os números de sequência de todas as entradas de estado de enlace mantidas no momento pelo transmissor
- Comparando seus próprios valores com os valores do transmissor, o receptor pode determinar quem tem os valores mais recentes
  - Essas mensagens são usadas quando uma linha é interrompida
- Cada roteador pode solicitar informações de estado de enlace um ao outro, usando mensagens **LINK STATE REQUEST**
- O resultado desse algoritmo é que cada par de roteadores adjacentes verifica quem tem os dados mais recentes, e as novas informações são divulgadas por toda a área
- Todas essas mensagens são enviadas como pacotes IP puros

Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

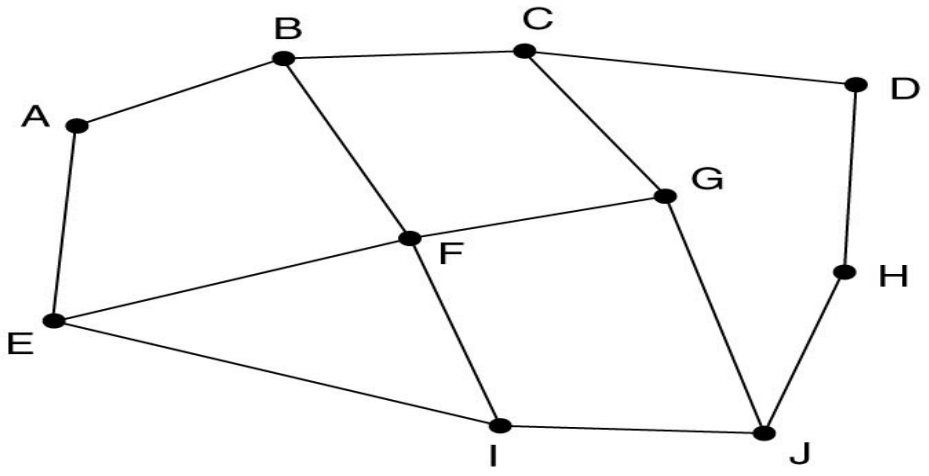
- ✔ Usando o processo de inundação, cada roteador informa todos os outros roteadores de sua área sobre seus vizinhos e custos
- ✔ Essas informações permitem que cada roteador construa o grafo para a(s) sua(s) área(s) e calcule o caminho mais curto
- ✔ A área do backbone faz o mesmo
- ✔ Os roteadores do backbone aceitam as informações dos roteadores de borda de área para calcular a melhor rota entre cada roteador do backbone até cada um dos outros roteadores
- ✔ Essas informações são propagadas para os roteadores de borda de área, que as divulgam em suas áreas
- ✔ Usando essas informações, um roteador para enviar um pacote entre áreas pode selecionar o melhor roteador de saída para o backbone

1.11 – BGP – Border Gateway Protocol

- ✔ Dentro de um único SA, o protocolo de roteamento recomendado na Internet é o OSPF
- ✔ Entre SAs é usado o BGP
- ✔ Os objetivos de um protocolo de gateway interior e os de um protocolo de gateway exterior não são os mesmos
- ✔ Protocolo de gateway interior não precisa se preocupar com política de transmissão → o roteador de gateway externo deve se preocupar muito
- ✔ Os protocolos de gateway exterior foram projetados para permitir a imposição de muitos tipos de políticas de roteamento no tráfego entre SAs
  - as políticas envolvem considerações políticas, econômicas e de segurança

- ✔ As políticas são configuradas manualmente em cada roteador BGP
  - elas não fazem parte do protocolo em si
- ✔ Para um roteador BGP, uma rede consiste de SAs e enlaces
- ✔ Dois SAs são considerados conectados se existe um enlace entre roteadores de borda de cada um deles
- ✔ O BGP agrupa as redes em três categorias:
  - **A** - fazem parte as **redes stub**, que têm somente uma conexão com o grafo BGP
    - não podem ser usadas para tráfego, porque não há ninguém do outro lado
  - **B** – fazem parte as **redes multiconectadas**, que podem ser usadas para tráfego, a menos que se recusem
  - **C** – fazem parte as **redes de trânsito**, tais como backbones, cujo objetivo é tratar pacotes de terceiros

- ✔ Os pares de roteadores BGP se comunicam entre si, estabelecendo conexões TCP
  - comunicação confiável além de poder ocultar todos os detalhes da rede que está sendo utilizada
- ✔ O BGP é um protocolo de vetor de distância, um pouco diferente da maioria das implementações:
  - além do custo para cada destino, cada roteador BGP tem controle de qual caminho está sendo usado
    - além de fornecer a cada vizinho seu custo estimado para cada destino possível, o roteador BGP informa a seus vizinhos o caminho exato que está usando



(a)

Information F receives
from its neighbors about D

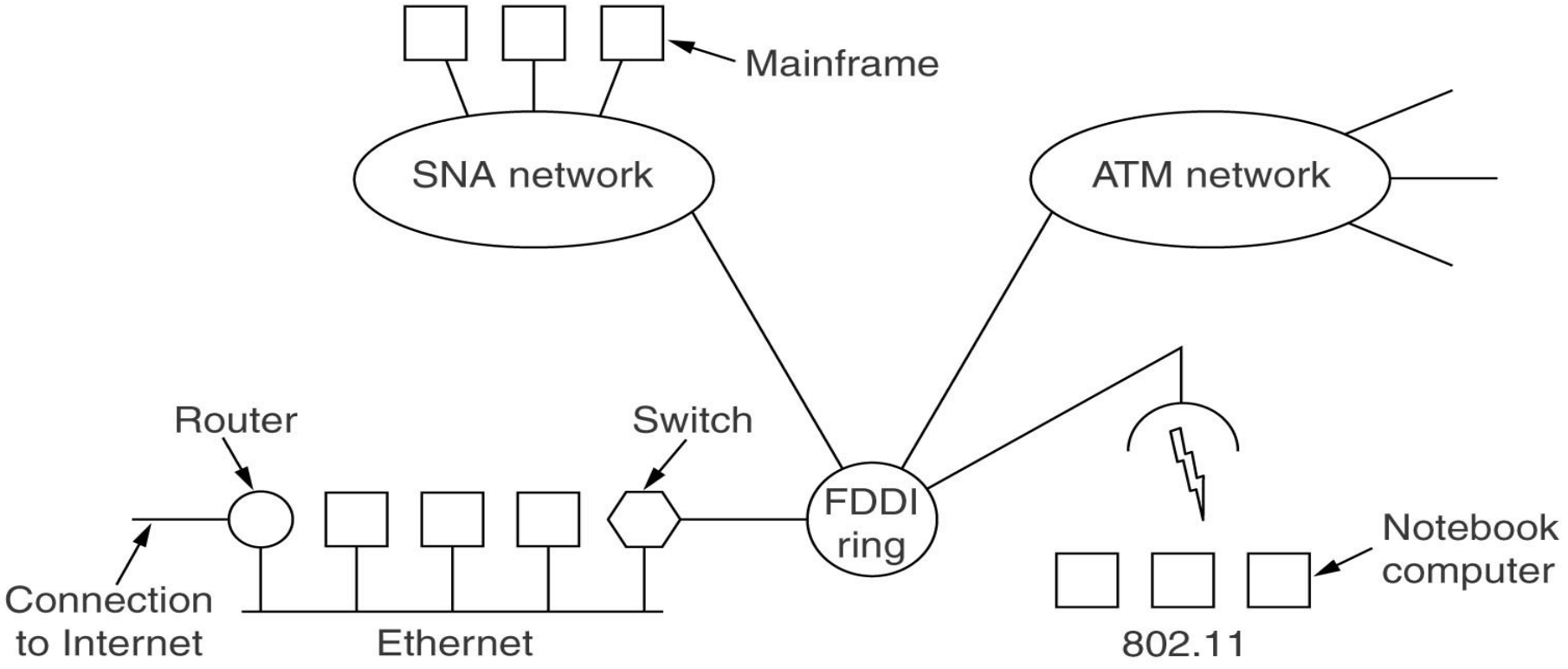
From B: "I use BCD"
From G: "I use GCD"
From I: "I use IFGCD"
From E: "I use EFGCD"

(b)

- ✔ Qualquer rota que viole uma restrição política recebe automaticamente uma contagem infinita
- ✔ O roteador adota a rota com a distância mais curta
- ✔ O BGP resolve o problema da contagem até infinito

1.12 - Interligação de Redes

- ✔ Infelizmente a Internet não é composta de um só tipo de rede: há vários tipos diferentes que precisam ser interligados (Ethernet, 802.11, 802.16, telefonia móvel e fixa, etc).
- ✔ Diferentes redes → diferentes protocolos
- ✔ A finalidade de interconectar todas as redes é permitir que usuários de qualquer uma se comuniquem com usuários de todas as outras
- ✔ Permitir também que usuários de qualquer uma acessem dados armazenados em qualquer das redes


Mainframe
SNA network
ATM network
Router
Switch
FDDI
ring
Notebook
computer
Connection
to Internet
Ethernet
802.11

1.12.1 - Diferenças entre redes

- As redes podem diferir em várias aspectos técnicos:
  - técnicas de modulação (camada física)
  - formato dos quadros (camada de enlace)

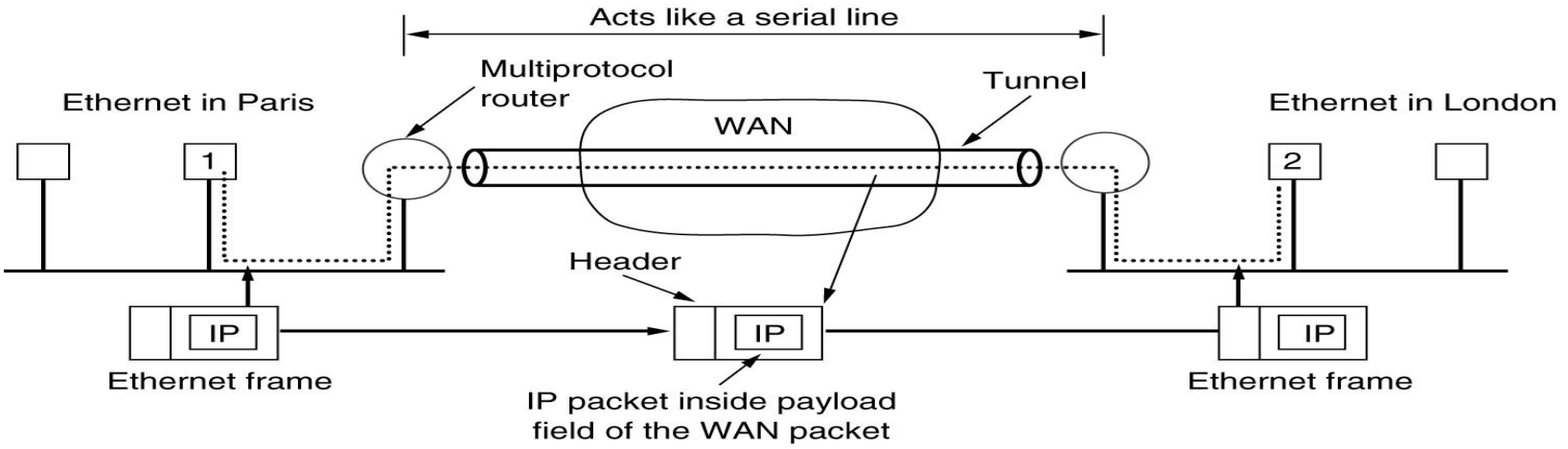
Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

1.12.2 - Como as redes podem ser interconectadas

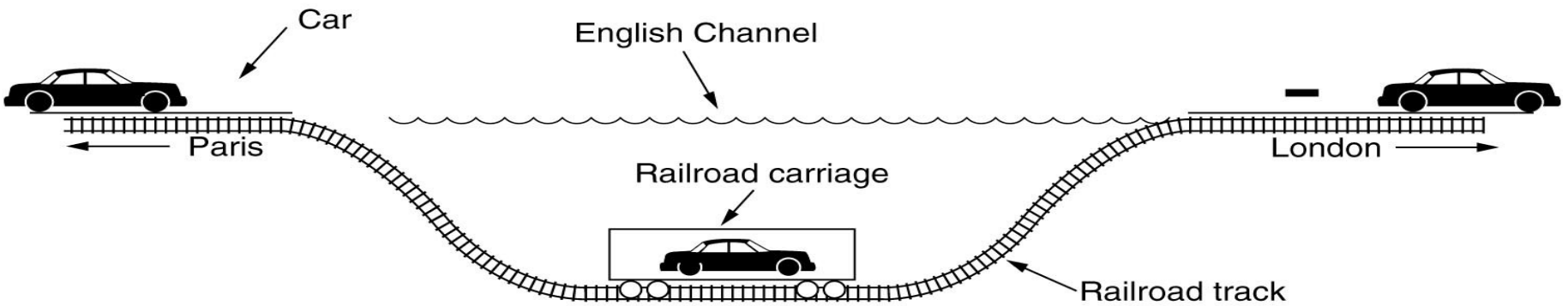
- ✔ Duas técnicas:
  - Utilizar dispositivos que façam conversão de pacotes de vários tipos, como hub, repetidores, switches, bridges, roteadores
  - Acrescentar uma camada de abstração sobre as diferentes redes, que seja comum a todas
- ✔ Independente da técnica, os dispositivos são colocados nas fronteiras entre as redes
- ✔ Exemplo de protocolos diferentes na camada de rede: IPv4 e IPv6
- ✔ Utilizar uma camada mais alta para a troca de pacotes?
- ✔ Traduzir os pacotes?

1.12.3 - Tunelamento

- Utilizado quando os hosts de origem e destino estão no mesmo tipo de rede, mas há uma rede de outro tipo entre eles
  - Ex: uma rede IPv6 de uma universidade em Paris se comunicando com uma rede IPv6 de uma universidade em Londres através de uma rede IPv4
  - O pacote IPv6 é encapsulado com um cabeçalho IPv4

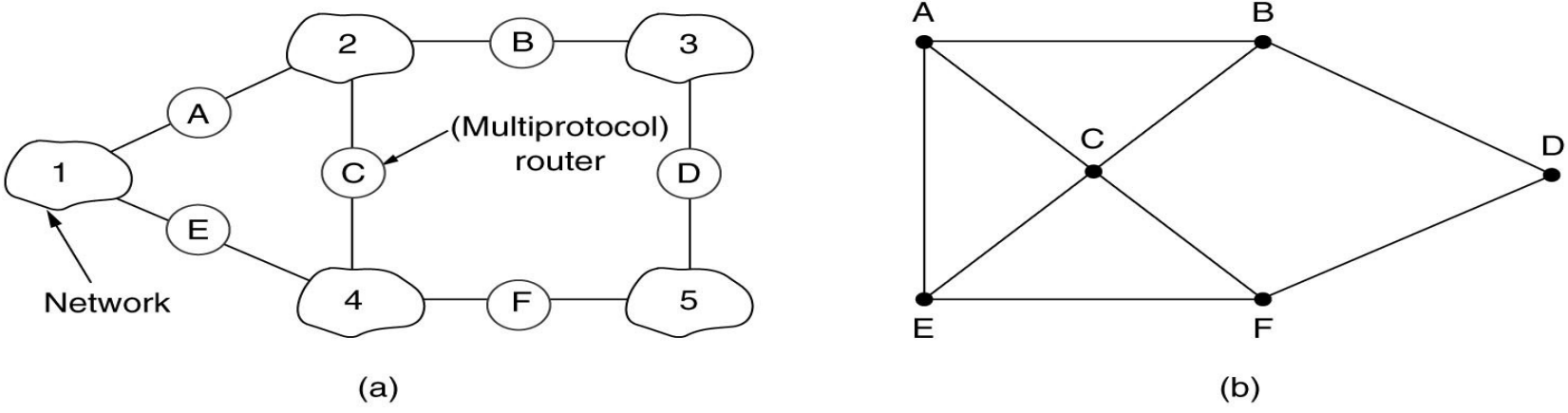


- A WAN pode ser considerada um grande túnel, que se estende de um roteador multiprotocolo a outro
- Nem o pacote IP, nem os hosts finais precisam se preocupar com a WAN, somente os roteadores multiprotocolos
- O caminho entre os dois roteadores multiprotocolos funciona como uma linha serial
- Analogia com um caso real:



1.12.4 - Roteamento entre redes

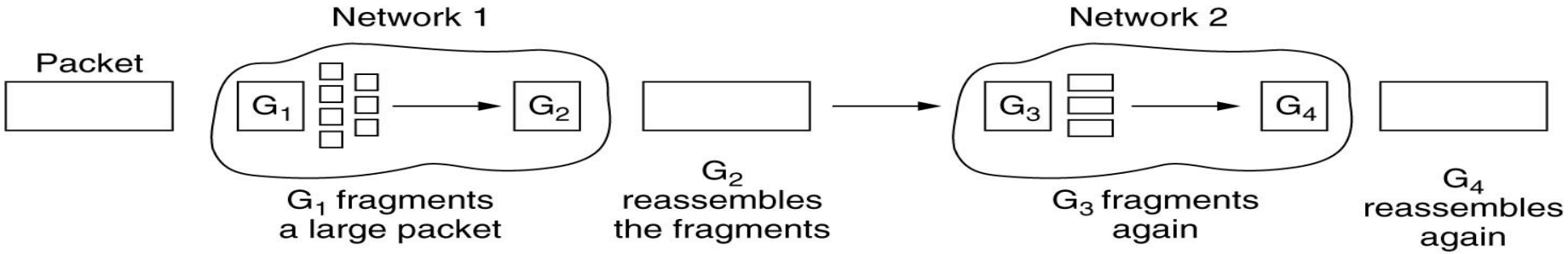
- ✔ O roteamento entre redes é semelhante ao roteamento em uma única rede, mas com complicações
- ✔ É usado um roteamento de dois níveis
  - Em cada rede é usado um protocolo de gateway interior
  - Entre as redes é usado um protocolo de gateway exterior
- ✔ Cada rede é independente (SA), todas elas podem usar diferentes algoritmos



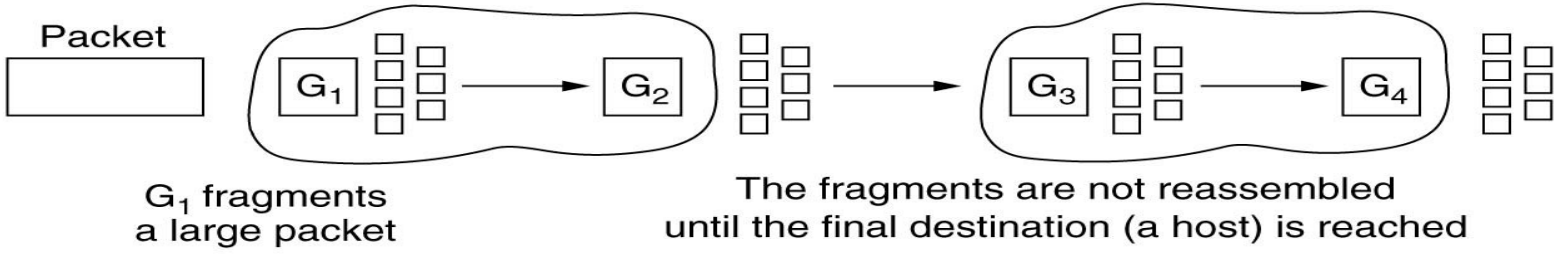
1.12.5 - Fragmentação de pacotes

- Cada rede impõe um tamanho máximo a seus pacotes
- As principais causas são:
  - 1. Hardware (por exemplo, o tamanho de um quadro Ethernet)
  - 2. Sistema operacional (por exemplo, tamanho dos buffers)
  - 3. Protocolos (por exemplo, o número de bits do campo de tamanho do pacote)
  - 4. Compatibilidade com algum padrão (inter)nacional
  - 5. Desejo de reduzir de alguma forma as retransmissões provocadas por erros
  - 6. Desejo de evitar que um pacote ocupe o canal por muito tempo
- O resultado de todos esses fatores é que os projetistas de redes não têm liberdade para escolher o tamanho máximo de pacote que desejam
- As cargas máximas variam de 48 bytes (células ATM) a 65.515 bytes (pacotes IP)

- Um problema surge quando um pacote muito grande tem de trafegar por uma rede cujo tamanho máximo de pacote é muito pequeno
- Uma solução é usar um algoritmo de roteamento que evite o envio de pacotes por redes que não sejam capazes de tratá-los
  - Não é a melhor solução: o que acontecerá se o pacote original for muito grande para ser tratado pela rede de destino?
  - Dificilmente o algoritmo de roteamento poderá ignorar o destino
- Basicamente, a única solução para o problema é permitir que os roteadores dividam os pacotes em fragmentos
  - É bem mais fácil converter um objeto grande em pequenos fragmentos do que executar o processo inverso
  - As redes de comutação de pacotes têm problemas para juntar os fragmentos novamente
- Existem duas estratégias opostas para recombinar os fragmentos e recompor o pacote original: ***fragmentação transparente*** e ***fragmentação não-transparente***


Network 1
Network 2
Packet
G1
G2
G3
G4
G1 fragments a large packet
G2 reassembles the fragments
G3 fragments again
G4 reassembles again

(a)


Packet
G1
G2
G3
G4
G1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached

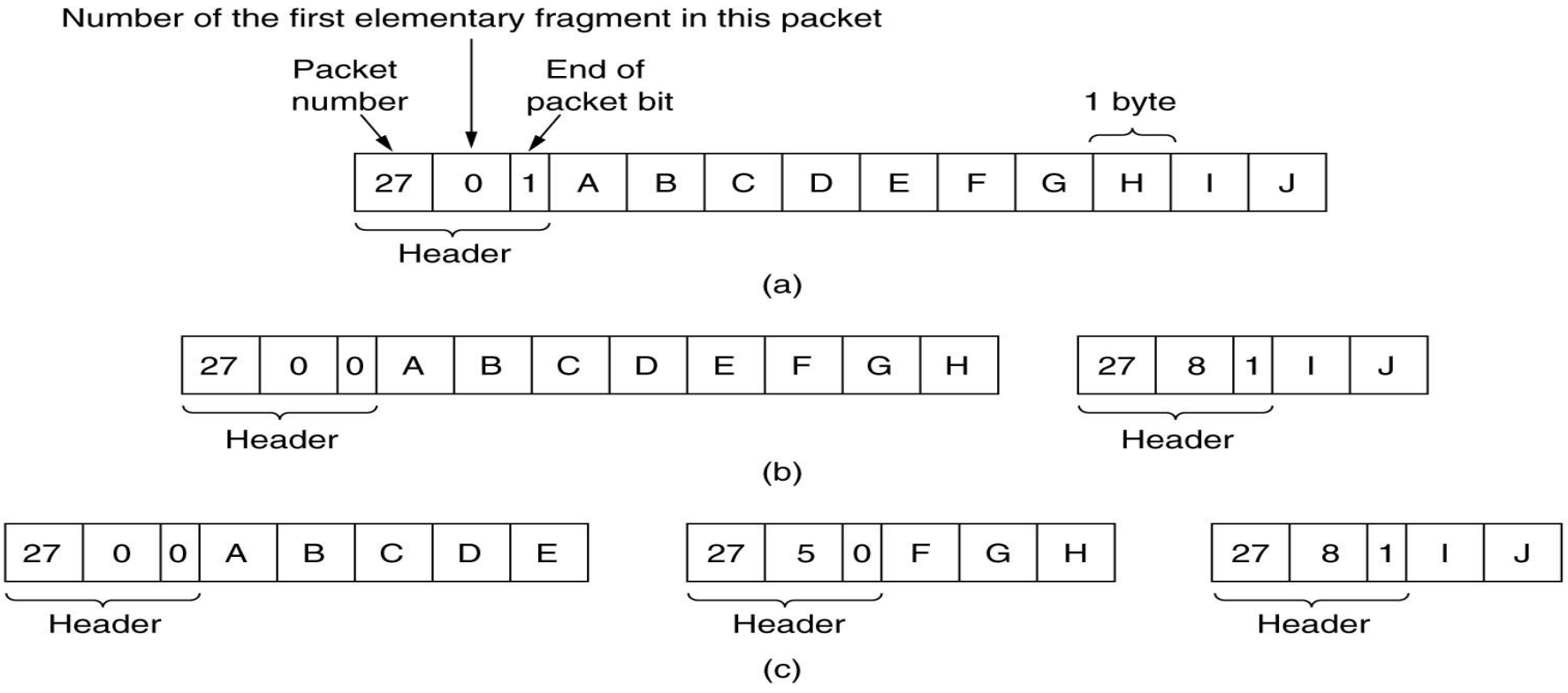
(b)

- A fragmentação transparente é simples, mas apresenta alguns problemas:
  - O roteador de saída deve saber quando recebeu todos os fragmentos → necessário incluir um campo de contagem ou bit de "fim de pacote" em cada pacote
  - Todos os pacotes têm de sair pelo mesmo roteador → como não é permitido que fragmentos sigam rotas diferentes até o destino final, há uma perda considerável em termos de desempenho
  - Um último problema é o overhead necessário para remontar e refragmentar repetidamente um pacote grande que passa por uma série de redes de pequenos pacotes
- A fragmentação não-transparente evita recombinar os fragmentos em roteadores intermediários
  - Quando um pacote é fragmentado, cada fragmento é tratado como se fosse o pacote original
  - A recombinação só ocorre no host de destino
  - O IP funciona dessa maneira

- A fragmentação não-transparente também apresenta alguns problemas:
  - Ela exige que todos os hosts sejam capazes de fazer a remontagem
  - Quando um pacote muito grande é fragmentado, o overhead total aumenta devido ao acréscimo de um cabeçalho a cada fragmento → o overhead permanece igual até o fim do trajeto
- Vantagem da fragmentação não-transparente:
  - Agora é possível usar vários roteadores de saída e obter um desempenho melhor → se o modelo de circuitos virtuais concatenados estiver sendo usado, essa vantagem não terá qualquer utilidade
- Os fragmentos devem ser numerados de forma que o fluxo de dados original possa ser reconstruído:
  - Uma forma de numerar os fragmentos é usar uma árvore → se o pacote 0 tiver de ser dividido, os fragmentos serão chamados de 0.0, 0.1, 0.2 etc. Se esses fragmentos tiverem de ser fragmentados posteriormente, os fragmentos resultantes serão 0.0.0, 0.0.1, 0.0.2,..., 0.1.0, 0.1.1, 0.1.2 etc

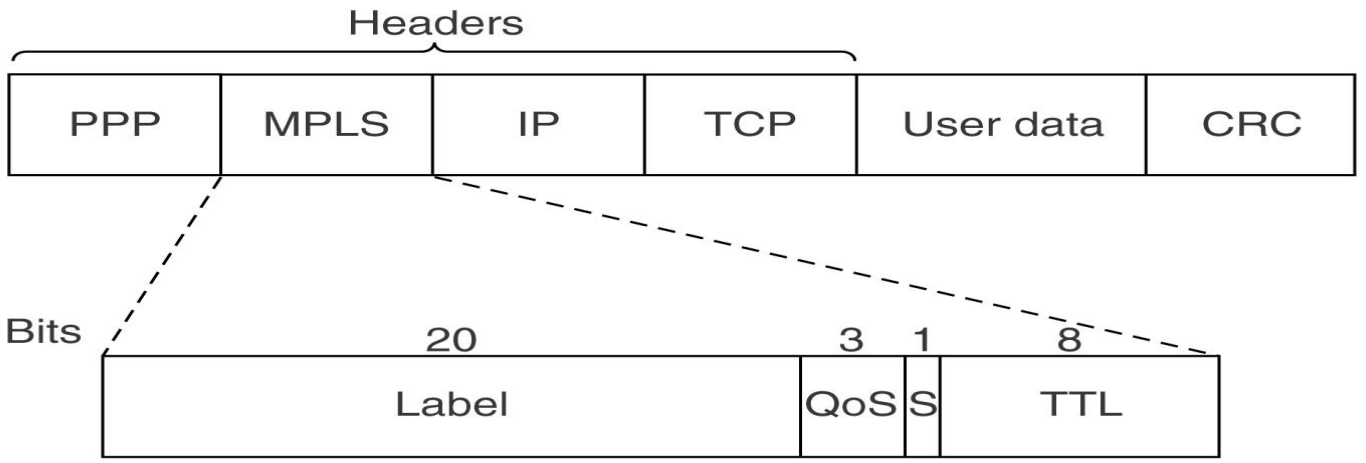
- Quando uma rede perde um pacote, há necessidade de uma retransmissão fim a fim, com efeitos desastrosos para o sistema de numeração
  - Suponha que um pacote de 1024 bits seja inicialmente dividido em quatro fragmentos de mesmo tamanho, 0.0, 0.1, 0.2 e 0.3
  - O fragmento 0.1 é perdido, mas os outros chegam ao destino
  - A origem retransmite o pacote original
  - Dessa vez, a rota seguida passa por uma rede com um limite de 512 bits → são gerados dois fragmentos agora em vez de quatro
  - Quando o novo fragmento 0.1 chegar ao destino, o receptor concluirá que todos os quatro fragmentos foram recebidos e reconstruirá o pacote incorretamente
- Um sistema de numeração diferente e melhor é fazer com que o protocolo defina um tamanho de fragmento elementar suficientemente pequeno para que esse fragmento possa passar por qualquer rede
- Quando um pacote é fragmentado, todas as partes têm tamanho igual ao do fragmento elementar, com exceção do último, que pode ser mais curto

- Outro tipo de numeração:



1.13 - Rótulos de comutação e MPLS

- MPLS (*MultiProtocol Label Switching* — comutação de rótulos multiprotocolo)
- Começa a ser largamente utilizada por ISP's nas suas sub-redes (núcleo da rede)
- Técnica bastante próxima da comutação de circuitos
- Acrescenta um rótulo na frente de cada pacote
- Encaminhamento é baseado no rótulo e não no endereço de destino → roteador despacha o pacote rapidamente
- Onde é colocado o rótulo?
  - Pacotes IP não foram projetados para circuitos virtuais
  - Não há um campo disponível no cabeçalho IP



- ✔ Cabeçalho MPLS possui 4 bytes e 4 campos:
  - Label: rótulo que identifica o trajeto
  - QoS: classe de serviço
  - S: relacionado ao empilhamento de rótulos
  - TTL: tempo de vida do pacote (impede o loop infinito)

- O protocolo MPLS fica entre a camada de enlace e a camada de rede: chamada de camada “2,5”
  - não pode ser da camada 3 pois depende do IP
  - não pode ser da camada 2 pois utiliza vários hops e não apenas um enlace
  - é independente das duas camadas (2 e 3) → pode ser utilizado em pacotes não-IP, pode transportar pacotes IP sobre redes que não usam IP
- A maioria dos hosts e roteadores não entendem o MPLS:
  - os rótulos deve ser anexados ou retirados de um pacote quando este alcança a borda de uma rede MPLS → roteador próprio para isso
- Os rótulos têm de ser remapeados a cada hop

- Uma diferença em relação aos circuitos virtuais tradicionais é o nível de agregação:
  - É possível cada fluxo ter seu próprio conjunto de rótulos na sub-rede. Porém, é mais comum osroteadores agruparem vários fluxos que terminam em um certo roteador ou LAN e usarem um único rótulo para eles
  - Dizemos que os fluxos agrupados sob um único rótulo pertencem à mesma FEC (*Forwarding Equivalence Class* — classe de equivalência de encaminhamento)
  - Essa classe abrange não apenas os lugares para onde os pacotes estão indo, mas também sua classe de serviço, porque todos os seus pacotes são tratados do mesmo modo para fins de encaminhamento
  - Com o roteamento tradicional de circuito virtual, não é possível agrupar vários caminhos distintos com pontos extremos diferentes no mesmo identificador de circuito virtual, porque não haveria como distingui-los no destino final → no caso do MPLS, os pacotes ainda contêm seu endereço de destino final, além do rótulo

- ✔ Criação do “circuito”→ há duas maneiras de criar as entradas da tabela de encaminhamento:
  - Abordagem orientada para dados, quando um pacote chega, o primeiro roteador que ele acessa entra em contato com o roteador situado mais abaixo por onde o pacote deve passar e pede que ele gere um rótulo para o fluxo → esse método é aplicado recursivamente
  - Abordagem orientada por controle, quando é inicializado, um roteador verifica para quais rotas ele é o destino final, cria uma ou mais FECs para elas, aloca um rótulo para cada uma e repassa os rótulos a seus vizinhos, que inserem os rótulos em suas tabelas de encaminhamento e enviam novos rótulos a seus vizinhos, até todos os roteadores se apossarem do caminho

- ✔ O MPLS pode operar em vários níveis ao mesmo tempo, acrescentando mais de um rótulo a um mesmo pacote
- ✔ No nível mais alto, cada concessionária pode usar o MPLS desde a origem até destino
- ✔ Dentro da rede de cada concessionária, o MPLS também pode ser usado em trechos, o que nos leva a um segundo nível de identificação
- ✔ Um pacote pode transportar uma pilha inteira de rótulos
- ✔ O bit S permite a um roteador remover um rótulo para saber se existem rótulos adicionais
  - Ele é definido como 1 para indicar o rótulo inferior e como 0 para todos os outros rótulos. Na prática, esse recurso é usado principalmente para implementar redes privadas virtuais e túneis recursivos