Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática
Departamento de Ciência da Computação
Curso de Ciência da Computação

Redes de Computadores I (parte 2)

Profª:Raquel Mini
raquelmini@pucminas.br

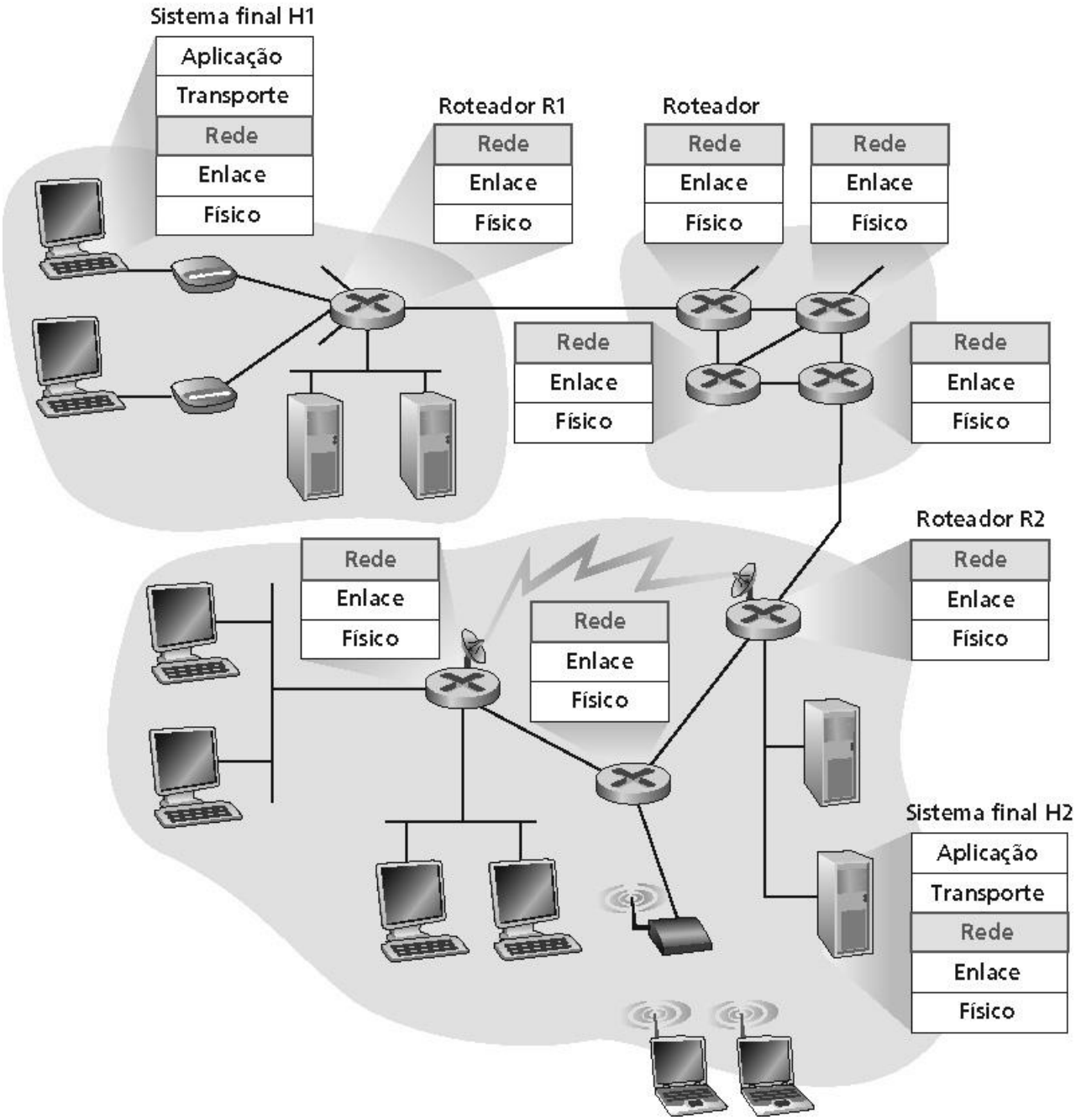
2º semestre de 2020

Camada de Rede

Camada de Rede

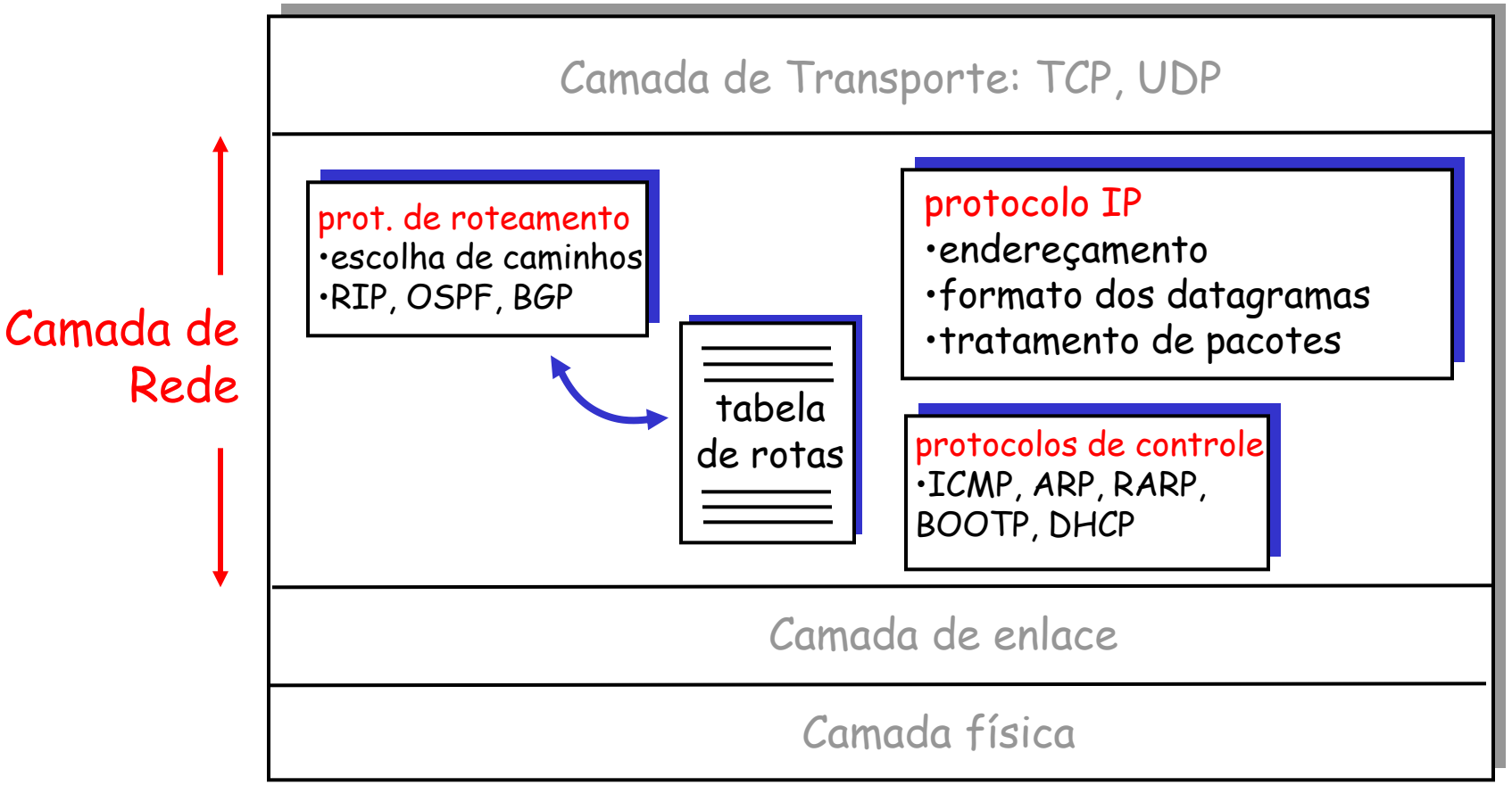
- A camada de enlace tem o objetivo de mover quadros de uma extremidade de um fio até a outra
- A camada de rede tem o objetivo de fazer a transferência de pacotes da origem para o destino
- A camada de rede deve conhecer a topologia da sub-rede de comunicações e escolher os caminhos mais apropriados através dela

Camada de Rede



A Camada de Rede da Internet

A Camada de Rede da Internet



A Camada de Rede da Internet

- Comutação de pacotes *store-and-forward*
- O elemento que mantém a Internet unida é o protocolo IP (*Internet Protocol*)
  - Foi projetado tendo como objetivo a interconexão de redes
  - Fornece a melhor forma possível de transportar pacotes da origem para o destino, independentemente das máquinas estarem na mesma rede ou de haver outras redes entre elas

O Protocolo IP

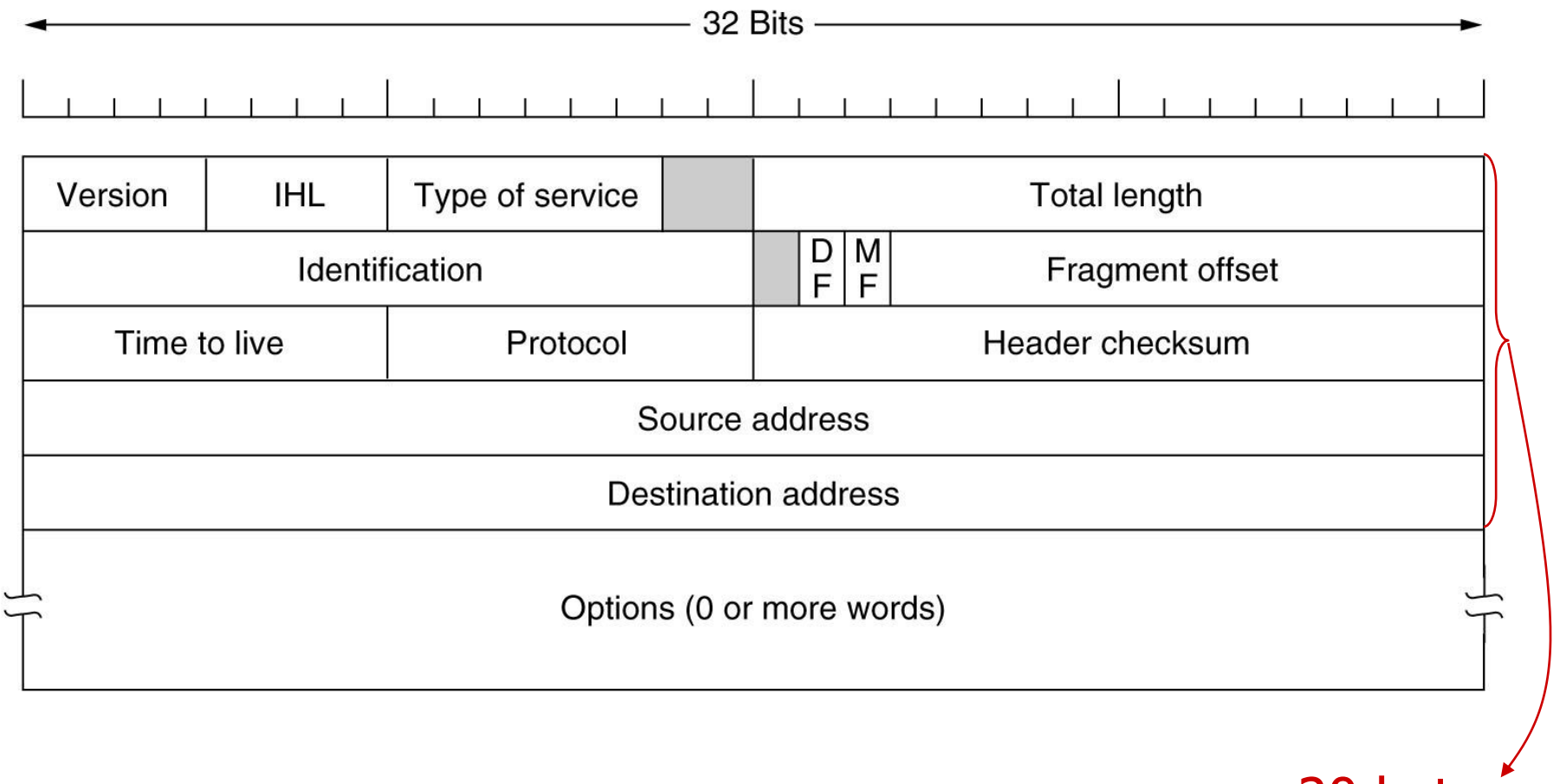
- Entrega será feita com o "melhor esforço" (*best-effort delivery*)
- No IP não há garantia:
  - de que a temporização entre pacotes seja preservada;
  - de que os pacotes sejam recebidos na ordem em que foram enviados;
  - da eventual entrega dos pacotes transmitidos;
- Protocolos de outros níveis devem tratar desses problemas

O Protocolo IP

- IPv4, ou simplesmente IP, permite um pacote de até 64 Kbytes
- O pacote IP consiste:
  - Cabeçalho
    - Fixa: 20 bytes
    - Opcional: tamanho variável
  - Carga útil

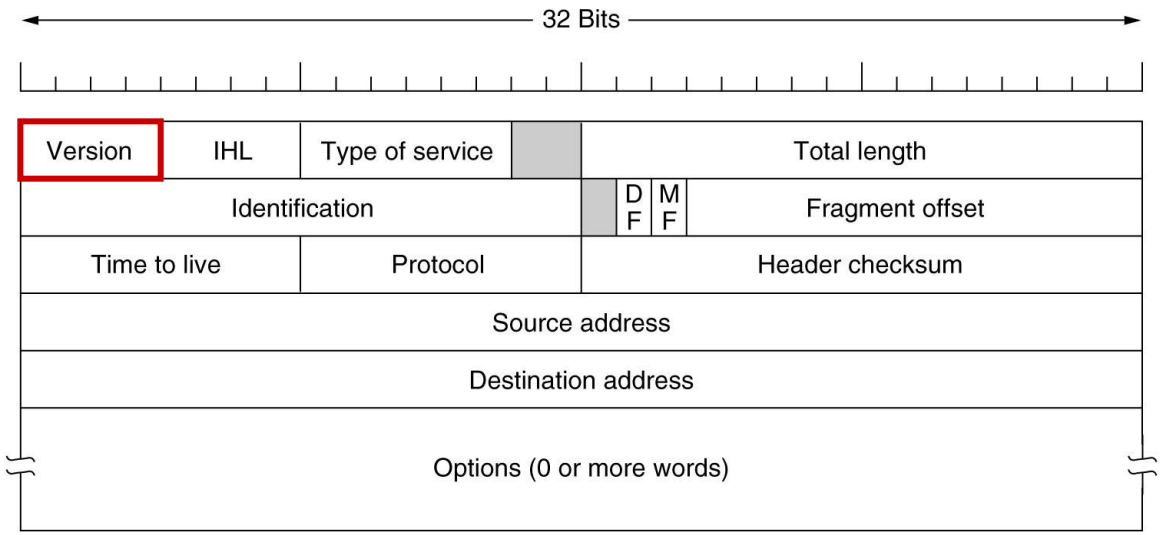
O Protocolo IP

- O cabeçalho IPv4



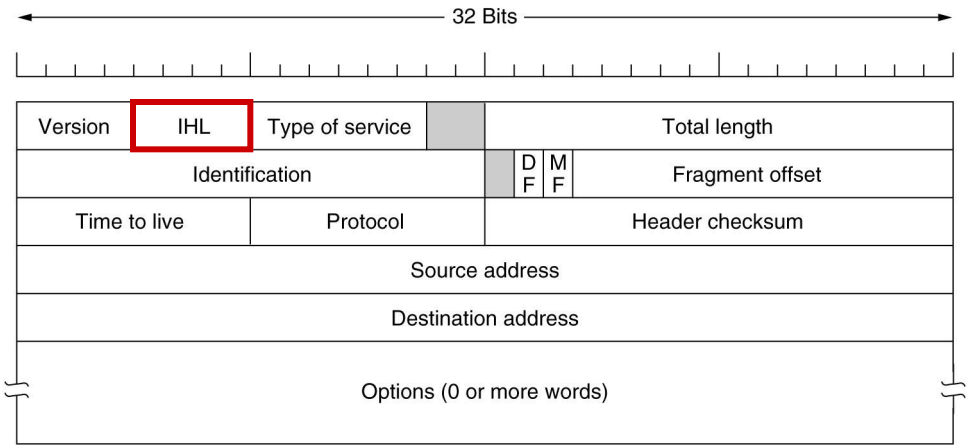
Formato do Pacote IP

- Version (4 bits):
  - Indica o número da versão corrente
  - Permite uma transição "suave" entre versões



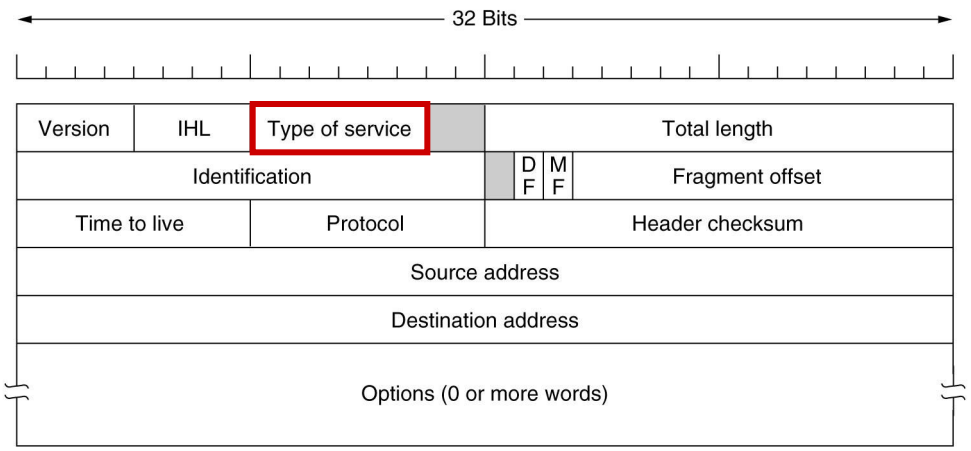
Formato do Pacote IP

- IHL (4 bits):
  - Informa o tamanho do cabeçalho em palavras de 32 bits
  - Mínimo: 5 (sem nenhuma opção)
  - Máximo: 15 (60 bytes sendo 40 bytes no campo Options)



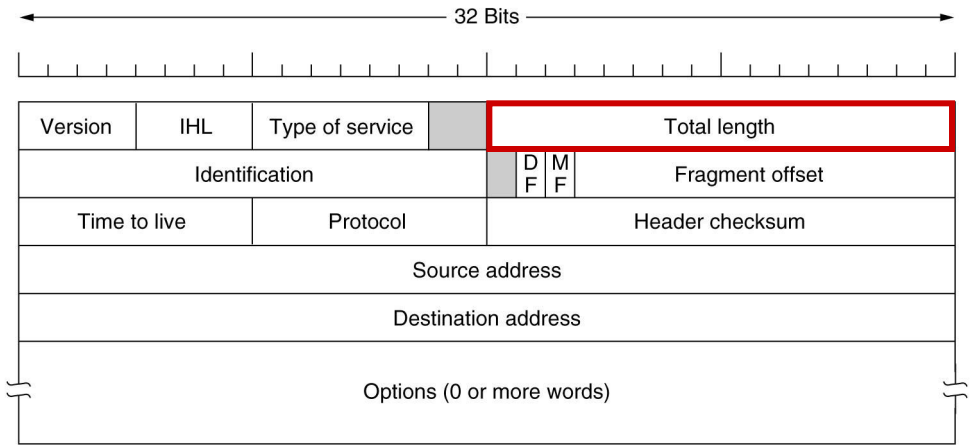
Formato do Pacote IP

- Type of service (6 bits):
  - Fazer a distinção entre diferentes classes de serviço
  - Indica o que é mais importante para a aplicação: menor atraso, maior vazão, maior confiabilidade
  - Na prática, os roteadores tendem a ignorar este campo



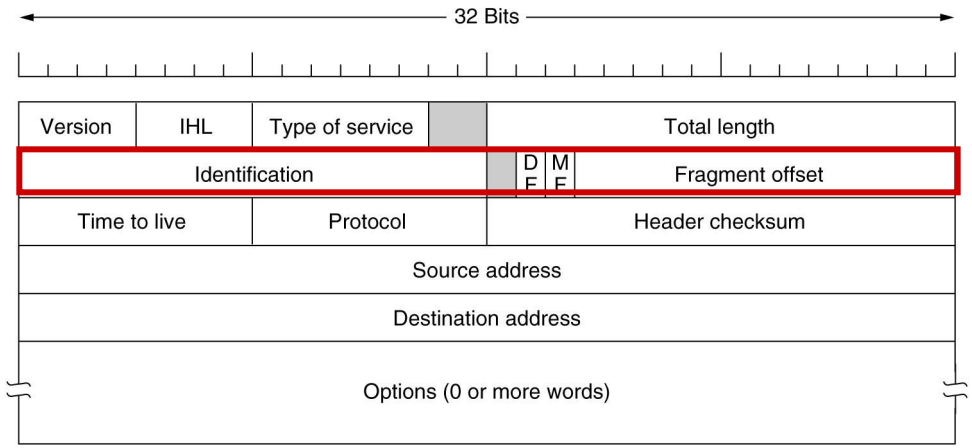
Formato do Pacote IP

- Total Length (16 bits):
  - Tamanho, em bytes, de todo o pacote: cabeçalho e dados
  - Valor máximo 65535



Formato do Pacote IP

- Identification (16 bits), Bit DF, Bit MF e Fragment Offset (13 bits)
  - Usados na fragmentação e remontagem de pacotes IP



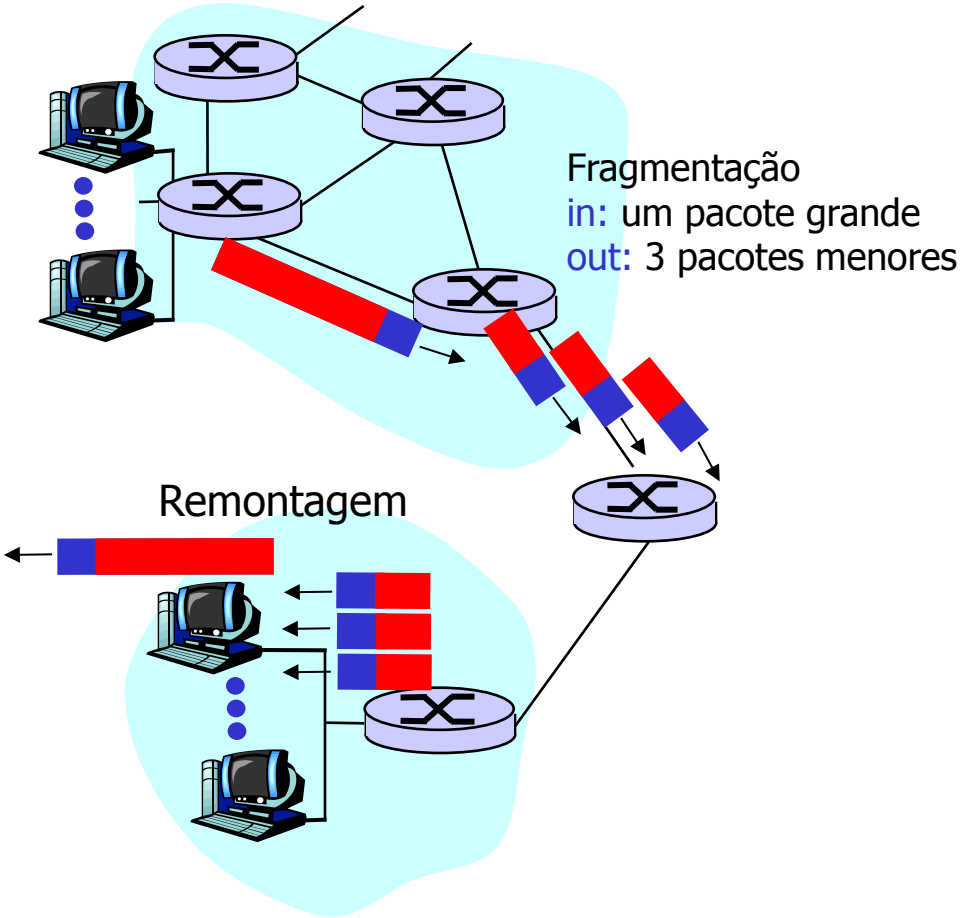
Formato do Pacote IP

- Fragmentação:
  - Nem todos os protocolos da camada de enlace podem carregar pacotes do mesmo tamanho
  - Exemplo: Ethernet não podem carregar mais do que 1500 bytes de carga útil
  - Os pacotes para muitos enlaces de longa distância não podem carregar mais do que 576 bytes
  - A quantidade máxima de dados que um pacote da camada de enlace pode carregar é chamada de **unidade máxima de transferência** (MTU)

Fragmentação de Pacotes IP

- Pacotes IP grandes devem ser divididos dentro da rede (fragmentados)
  - Um pacote dá origem a vários pacotes (fragmentos)
  - "Remontagem" ocorre apenas no destino final
  - O cabeçalho IP é usado para identificar e ordenar pacotes relacionados
  - Se um ou mais fragmentos não chegarem ao destino, o pacote será descartado e não será passado à camada de transporte

Fragmentação de Pacotes IP

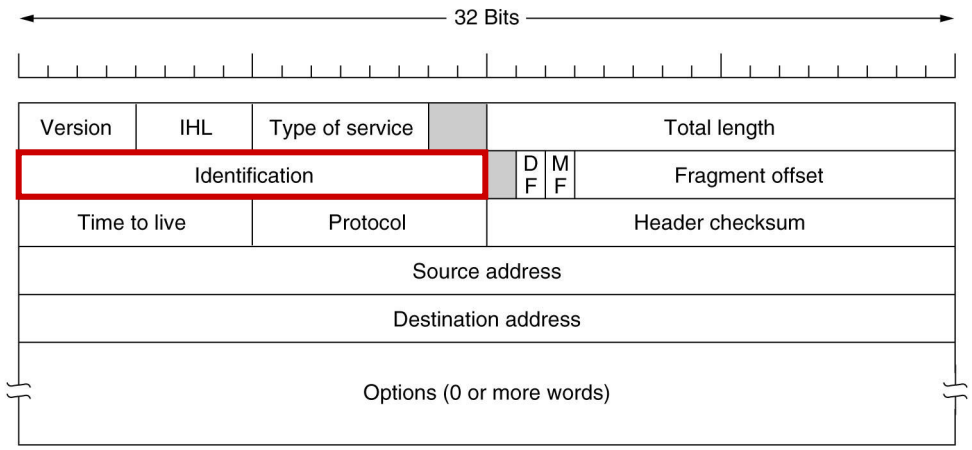


Fragmentação de Pacotes IP

- Cada pacote criado é marcado pelo hospedeiro remetente com um número de identificação
- O hospedeiro remetente incrementa o número de identificação para cada pacote que envia

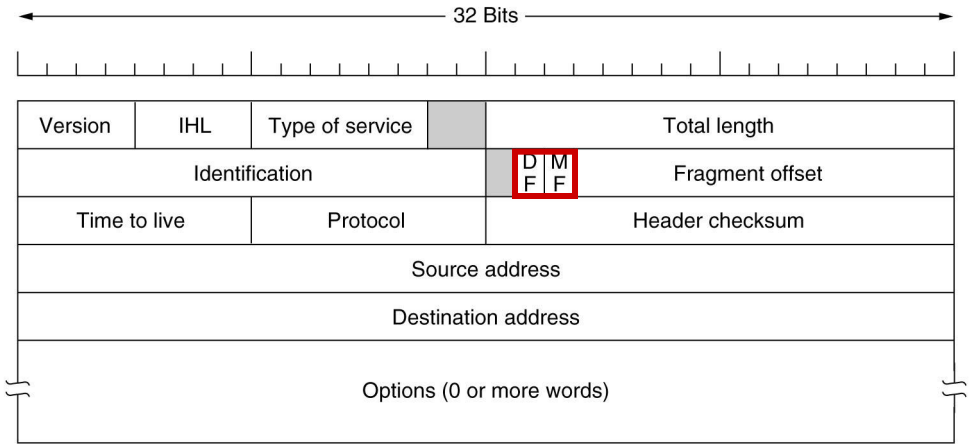
Formato do Pacote IP

- Identification (16 bits):
  - Identifica o pacote ou o fragmento de um pacote
  - Usado pelo destinatário para remontagem
  - Todos os fragmentos de um pacote contêm o mesmo valor de Identification



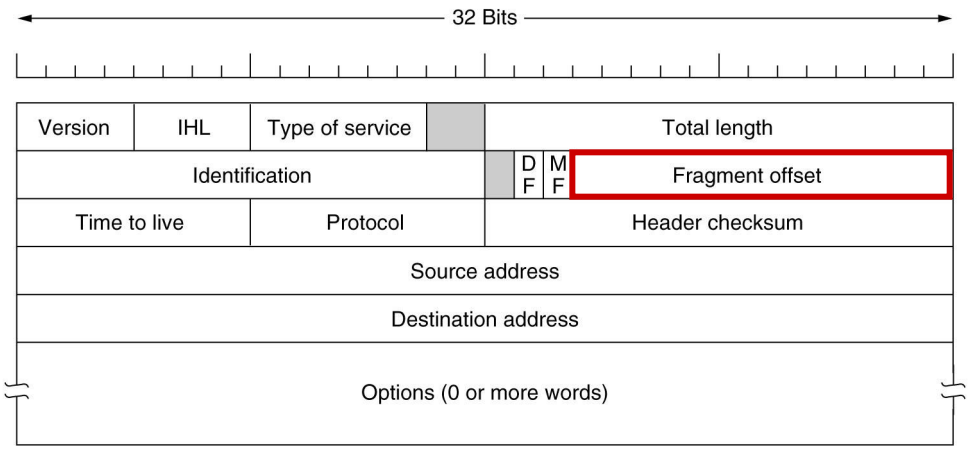
Formato do Pacote IP

- Bit DF (*don't fragment*):
  - Indica que o pacote não deve ser fragmentado
- Bit MF (*more fragments*):
  - Todos os fragmentos de um pacote, exceto o último, marcam este bit



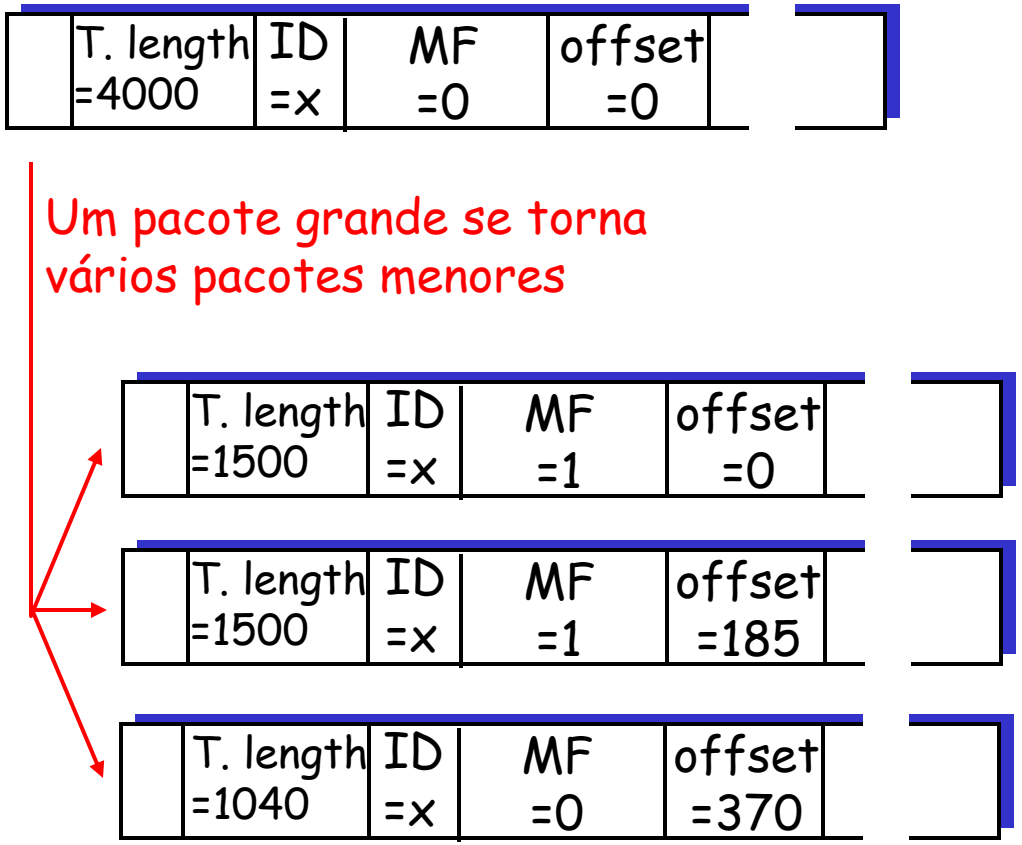
Formato do Pacote IP

- Fragment Offset (13 bits):
  - Indica onde (byte igual ao seu conteúdo multiplicado por 8) o fragmento se encaixa dentro do pacote
  - A carga útil de cada fragmento, exceto o último, deve ser múltiplo de 8
  - Existem no máximo 8.192 fragmentos por pacote, resultando em um tamanho máximo de pacote igual a 65.536 bytes



Fragmentação de Pacotes IP – Exemplo

- Segmento de 3.980 bytes em uma rede com MTU de 1500 bytes



Fragmentação de Pacotes IP

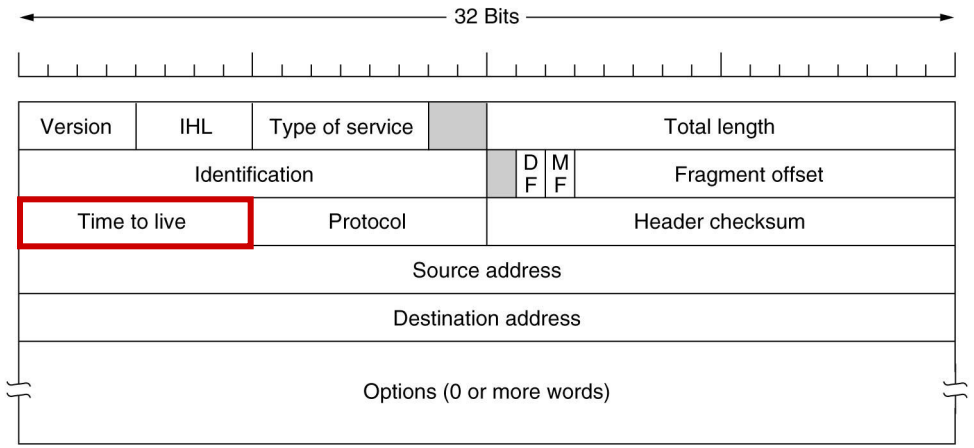
- A fragmentação e a remontagem colocam uma carga adicional sobre os roteadores da Internet (fazem a fragmentação) e sobre os hospedeiros (fazem a remontagem)
  - É desejável que se minimize a quantidade de fragmentação
  - Limita-se os segmentos TCP e UDP a tamanhos relativamente pequenos de modo que a fragmentação dos pacotes seja pouco provável
  - Todos os protocolos suportados pelo IP têm MTU de, no mínimo, 576 bytes
  - Na prática, o TCP e o UDP passam para o IP segmentos de no máximo 556 bytes

Exercício

1. Suponha que o host A esteja conectado a um roteador R1, que R1 esteja conectado a outro roteador R2, e que R2 esteja conectado ao host B. Suponha que uma mensagem TCP contendo 900 bytes de dados e 20 bytes de cabeçalho TCP seja repassada ao código IP do host A para ser entregue a B. Mostre os campos *Total length*, *Identification*, *DF*, *MF* e *Fragment offset* do cabeçalho IP em cada pacote transmitido pelos três enlaces. Suponha que o enlace A-R1 possa admitir um tamanho máximo de quadro de 1.024 bytes, incluindo um cabeçalho de quadro de 14 bytes, que o enlace R1-R2 possa admitir um tamanho máximo de quadro de 512 bytes, incluindo um cabeçalho de quadro de 8 bytes, e que o enlace R2-B possa admitir um tamanho máximo de quadro de 512 bytes, incluindo um cabeçalho de quadro de 12 bytes.

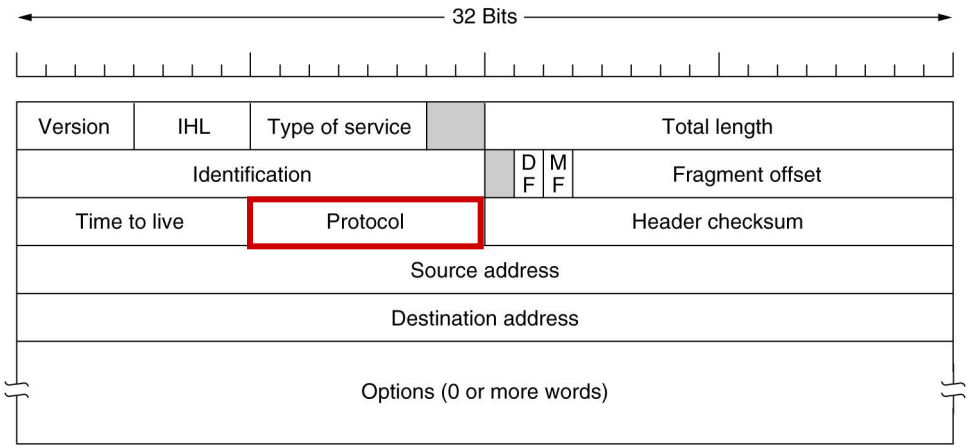
Formato do Pacote IP

- Time To Live (8 bits):
  - Contador usado para limitar a vida útil dos pacotes
  - Indica o número máximo de roteadores (*hops*) que pode passar
  - Quando chega a zero, o pacote é descartado e um pacote de advertência é enviado ao computador de origem



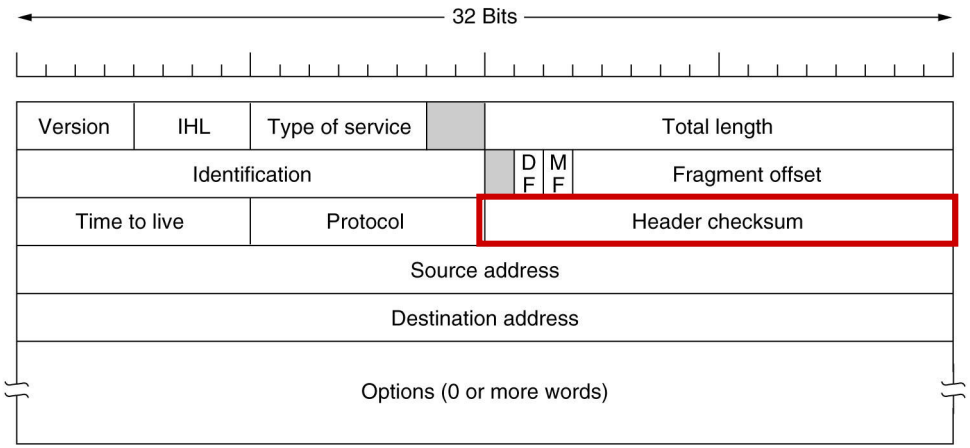
Formato do Pacote IP

- Protocolo (8 bits):
  - Indica o protocolo para o qual deve-se passar o pacote (TCP ou UDP)



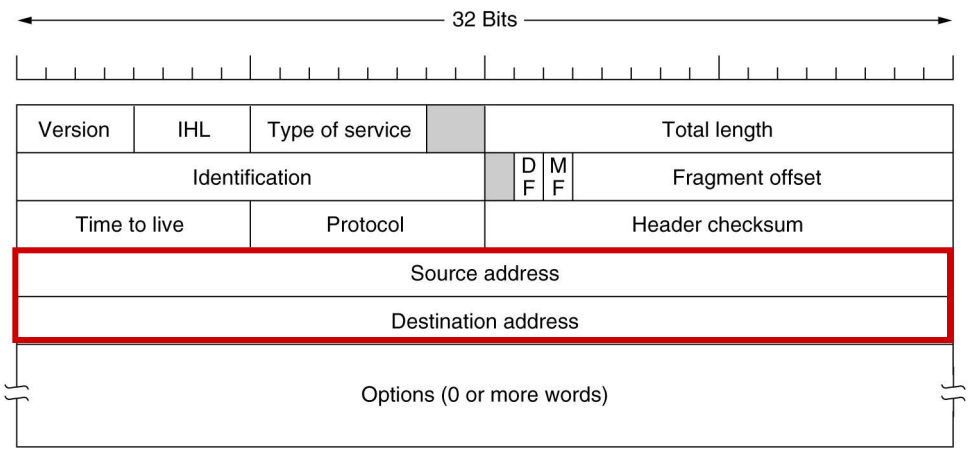
Formato do Pacote IP

- Header Checksum (16 bits):
  - Total de verificação do cabeçalho
  - Deve ser calculado novamente em cada roteador porque o campo Time To Live sempre se altera



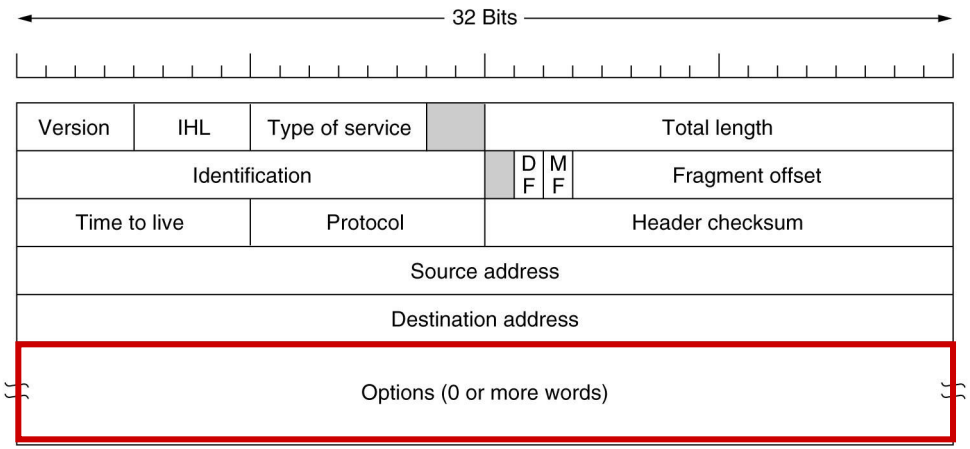
Formato do Pacote IP

- Source Address (32 bits) e Destination Address (32 bits)
  - Endereços IP dos computadores origem e destino



Formato do Pacote IP

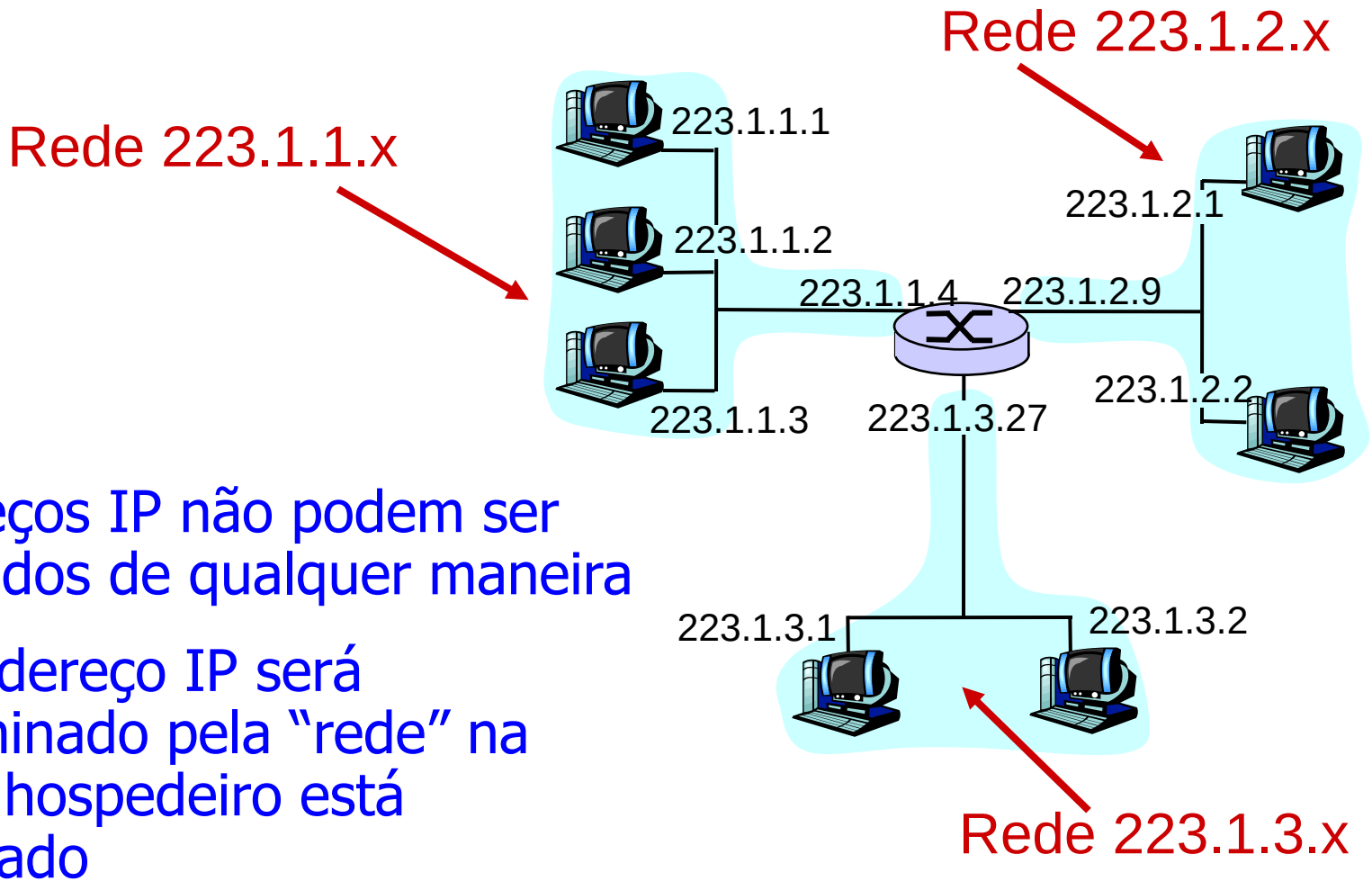
- Options:
  - Forma de incluir informações não presentes na versão



Endereços IP

- Na Internet, cada hospedeiro e cada roteador tem um endereço IP
- Endereço IP de um computador:
  - Número binário único de 32 bits (teoricamente 4.294.967.296 endereços)
  - Usados nos campos *Source Address* e *Destination Address* dos pacotes IP
  - Auxilia no roteamento
- Um endereço IP identifica uma interface de rede (se um hospedeiro tiver duas interfaces de redes, ele precisará de dois endereços IP)

Endereços IP

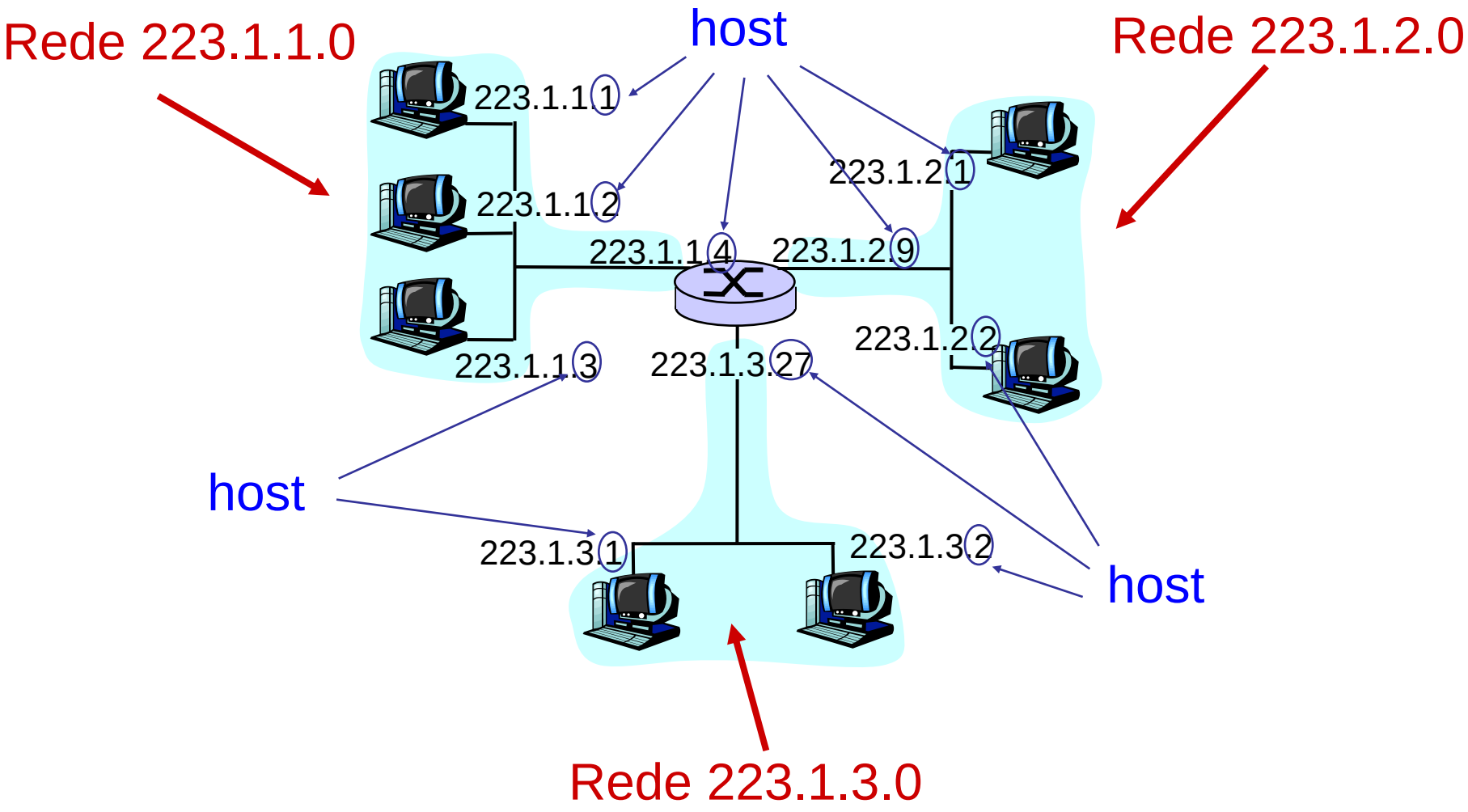


- Endereços IP não podem ser escolhidos de qualquer maneira
- Um endereço IP será determinado pela "rede" na qual o hospedeiro está conectado

Endereços IP

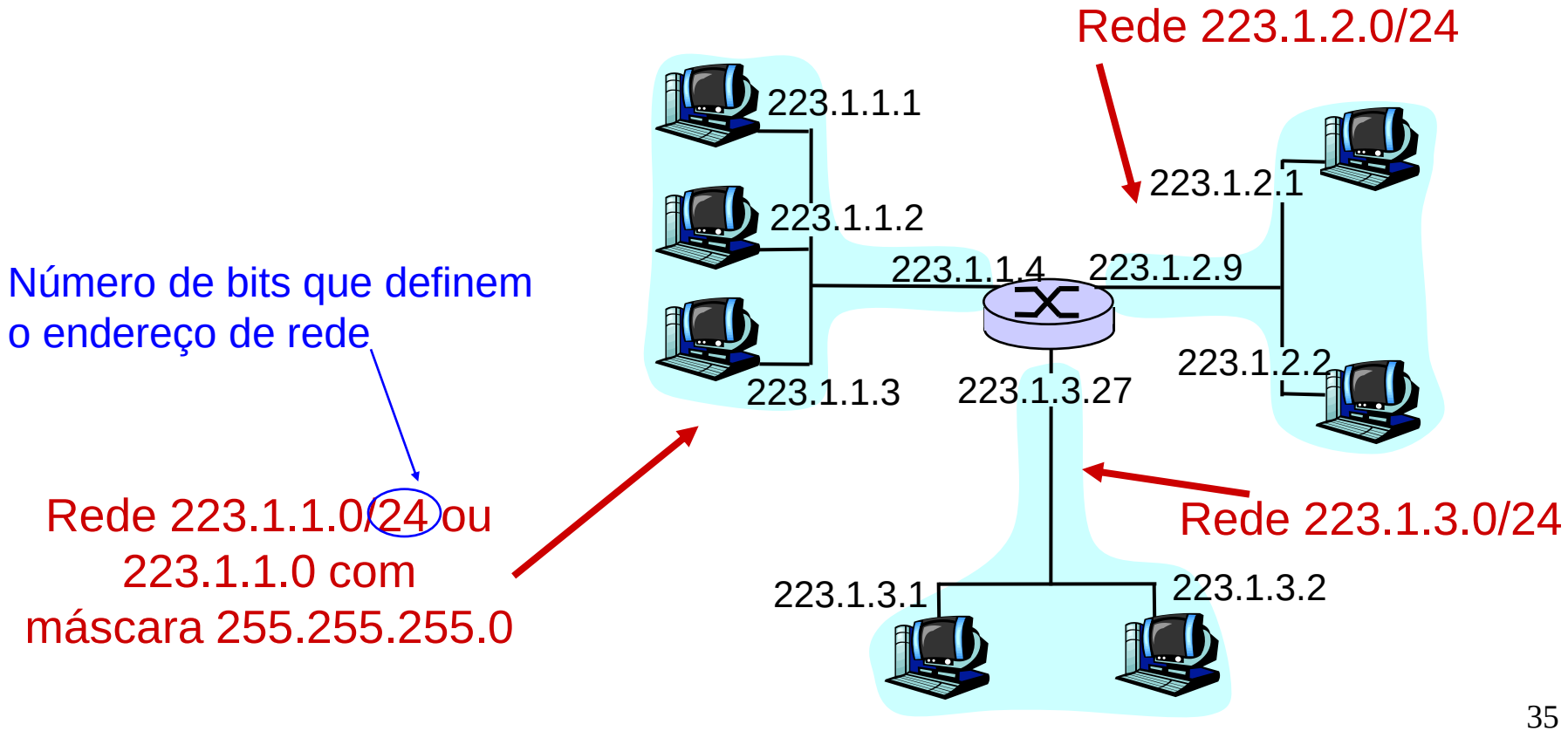
- E endereço IP é dividido em duas partes:
  - Prefixo:
    - Identifica a rede física na qual o computador se encontra (número de rede)
    - Controlado globalmente
  - Sufixo:
    - Identifica o hospedeiro na rede
    - Controlado localmente

Endereços IP

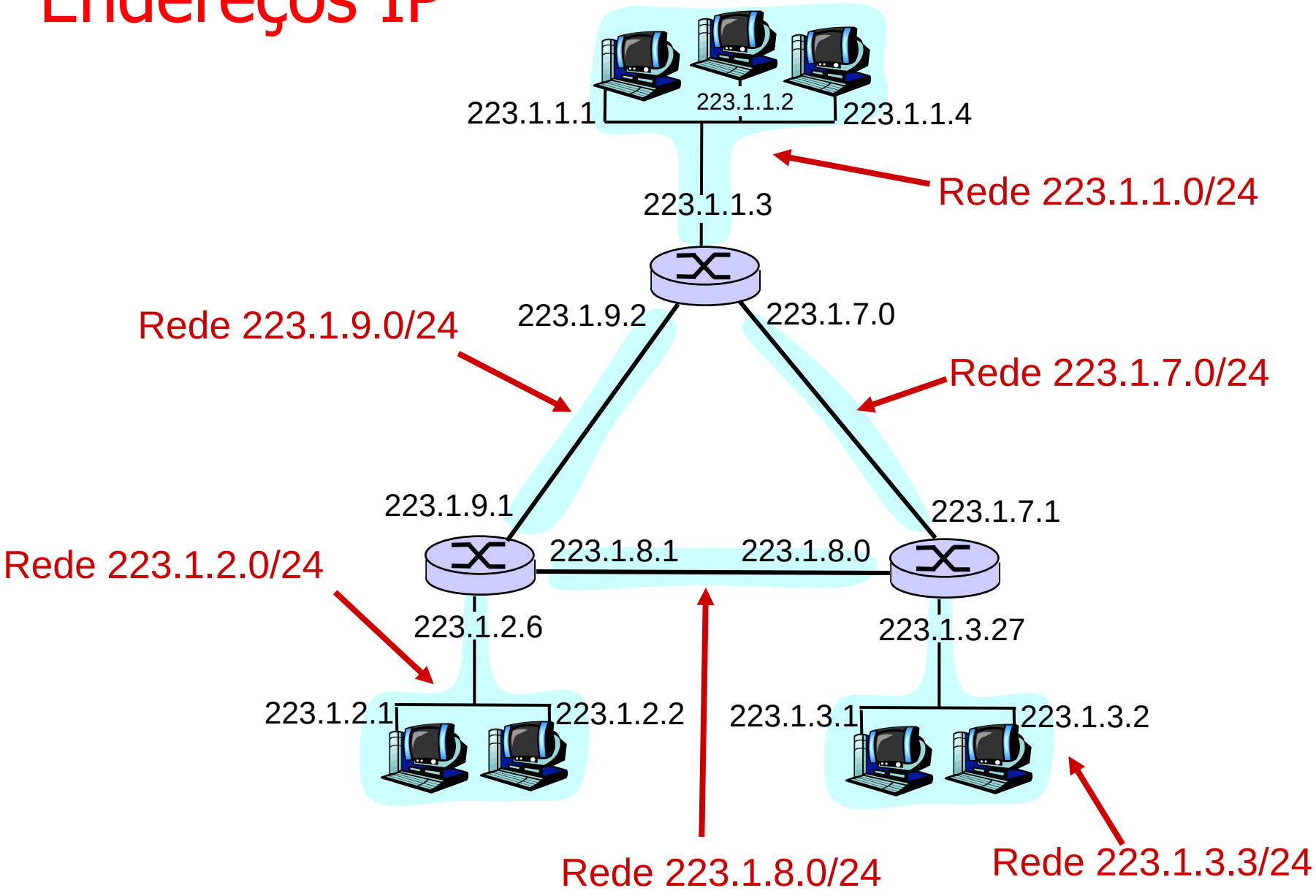


Endereços IP

- Máscara da rede:
  - Mostra a divisão entre endereço de rede e endereço de computador

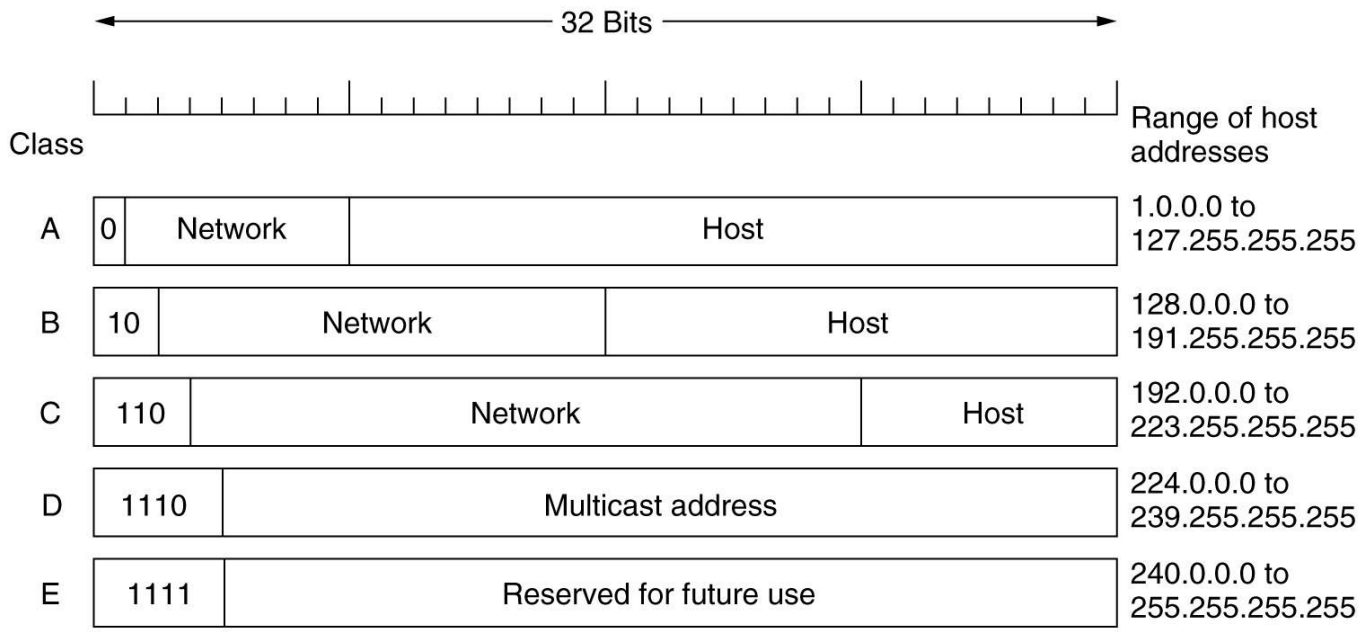


Endereços IP



Endereçamento de Classe Completo

- A arquitetura de endereçamento original da Internet definia classes de endereços



Endereçamento de Classe Completo

- Espaço de endereçamento:

Classe	Nº de Bits no Prefixo	Nº Máximo de Redes	Nº de Bits no Sufixo	Nº Máximo de Hosts por Rede
A	7	128	24	16.777.216
B	14	16.384	16	65.536
C	21	2.097.152	8	256

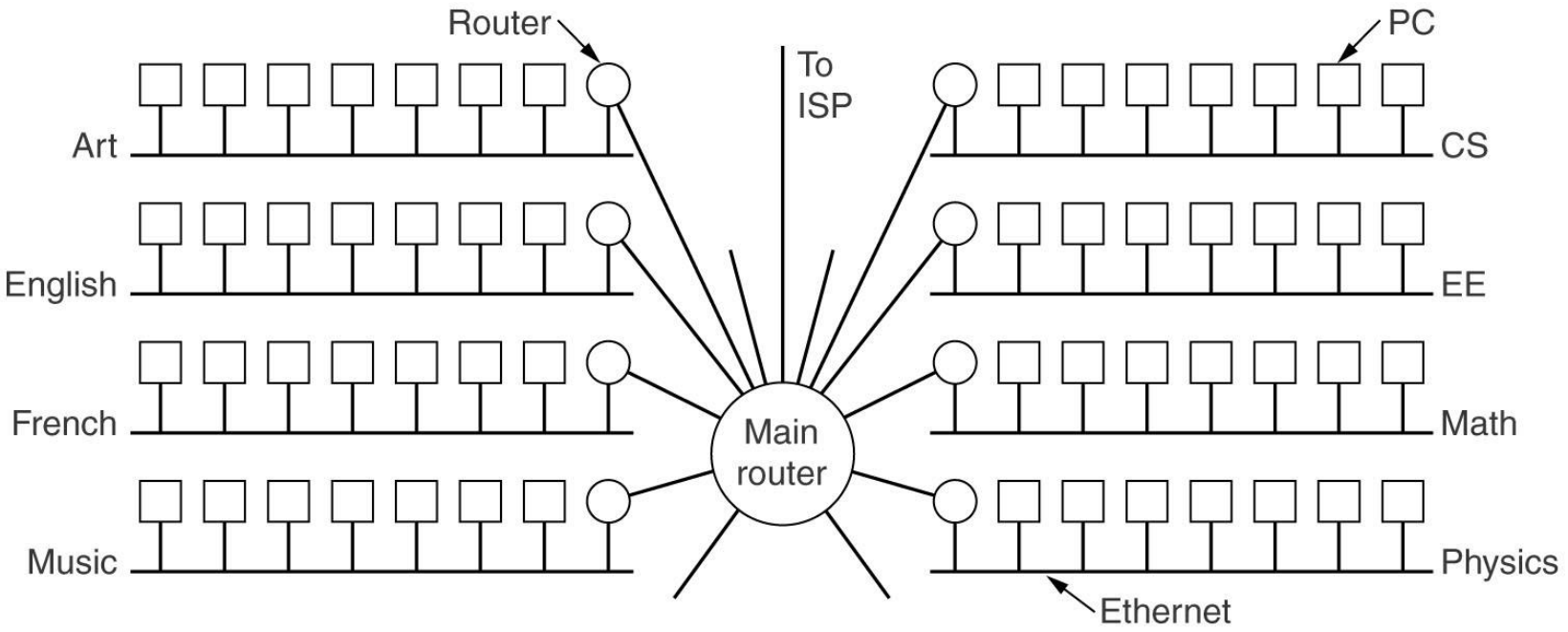
Exercícios

2. Suponha que, em vez de serem utilizados 16 bits na parte de rede de um endereço da classe B, tenham sido usados 20 bits. Nesse caso, quantas redes da classe B existiriam?
3. A máscara de sub-rede de uma rede na Internet é 255.255.240.0. Qual é o número máximo de hospedeiros que ela pode manipular?

Sub-redes

- Todos os hospedeiros de uma rede devem ter o mesmo endereço de rede
- Inicialmente:
  - Um único endereço da classe A, B ou C se refere a uma única rede
- À medida que mais e mais organizações se conectavam à rede:
  - Foi necessário permitir que uma rede fosse dividida em diversas partes para uso interno, mas externamente continuasse a funcionar como uma única rede

Sub-redes



Uma rede de campus consistindo em LANs para vários departamentos

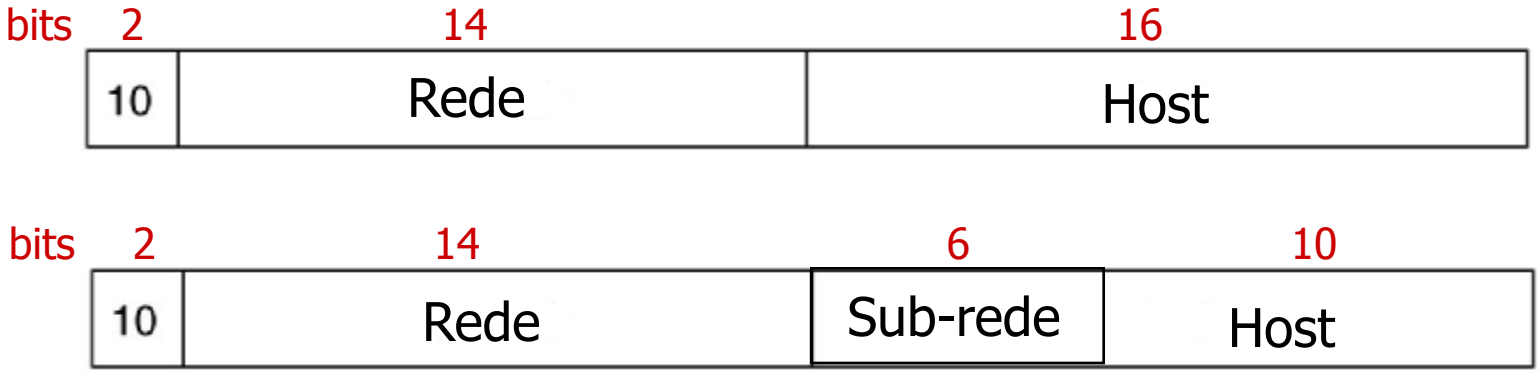
Sub-redes

- Uma rede pode ser dividida em diversas partes
- As partes da rede são chamadas sub-redes
  - Alguns bits são retirados do número do hospedeiro para criar um número de sub-rede
  - Número de bits em uma rede Classe B:

	Sem Sub-rede	**Com Sub-rede**
Fixo	2	2
Rede	14	14
Sub-rede	0	X
Host	16	16-X

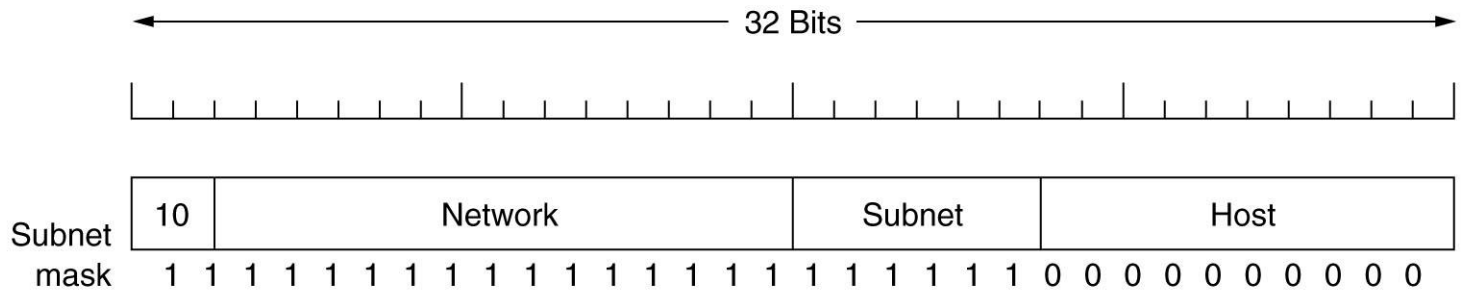
Sub-redes

- Exemplo de uma Universidade
  - Classe B
  - 35 departamentos
  - 64 redes Ethernet, cada uma com o máximo de 1022 hots (0 e –1 não estão disponíveis)



Sub-redes

- Uma **máscara** (*netmask*) indica a divisão entre o número de rede + sub-rede e o hospedeiro
- Exemplo:



  - Máscara: 255.255.252.0 ou /22

Sub-redes

- Exemplo: rede 130.50.0.0 e máscara de sub-rede 255.255.252.0 ou /22
  - Sub-rede 0: 130.50.0.1 a 130.50.3.254
    10000010 00110010 000000|00 00000001 a
    10000010 00110010 000000|11 11111110
  - Sub-rede 1: 130.50.4.1 a 130.50.7.254
    10000010 00110010 000001|00 00000001 a
    10000010 00110010 000001|11 11111110
  - Sub-rede 2: 130.50.8.1 a 130.50.11.254
    10000010 00110010 000010|00 00000001 a
    10000010 00110010 000010|11 11111110
  - Sub-rede 3: 130.50.12.1 a 130.50.15.254
    10000010 00110010 000011|00 00000001 a
    10000010 00110010 000011|11 11111110

Exercício

4. Um ISP possui o seguinte bloco de endereços: 150.164.192.0/18. Ele deseja dividir esse espaço de endereçamento igualmente entre 4 organizações. Informe a quantidade de endereços IP cada organização terá, o endereço inicial, final e a máscara de cada organização.
5. Se o endereço IP 201.40.12.231/26 faz parte de uma rede local, quais são os outros endereços dessa rede? E os endereços de broadcast e o de rede? Justifique sua resposta e mostre todos os cálculos.

Exercício

6. Quantas redes diferentes com máscara 255.255.255.240 podem ser alocadas dentro da faixa de IP 172.16.0.0/16?

Sub-redes

- Fora da rede, a divisão em sub-redes não é visível
- A alocação de uma nova sub-rede não exige a mudança de quaisquer bancos de dados externos
- Como os pacotes IP são processados em um roteador?

Roteamento de Pacotes IP

- Cada roteador mantém uma tabela de roteamento
  - A tabela é inicializada quando o roteador é ligado e deve ser atualizada se a topologia muda ou há uma falha de hardware
  - Cada entrada da tabela especifica um destino e o próximo roteador a ser usado para alcançar esse destino
  - Cada roteador só precisa manter entradas para as outras redes e para os hospedeiros locais

Roteamento de Pacotes IP

- Quando um pacote IP é recebido, procura-se pela entrada cujo endereço case com o AND entre a sua máscara e o endereço de destino do pacote IP
  - Se o destino for de uma rede distante, o pacote será encaminhado para o próximo roteador da interface fornecida na tabela
  - Se o destino for um hospedeiro local, o pacote será enviado diretamente
  - Se a rede do destino não estiver presente na tabela, o pacote será enviado para um roteador predefinido que tenha tabelas maiores (*default gateway*)

Roteamento de Pacotes IP

- Endereço de destino x *Next-hop*
  - Endereço de destino indica para quem deve ser entregue o pacote
  - Endereço de *Next-hop* indica para que roteador o pacote deve ser enviado
  - *Next-hop* não aparece no pacote

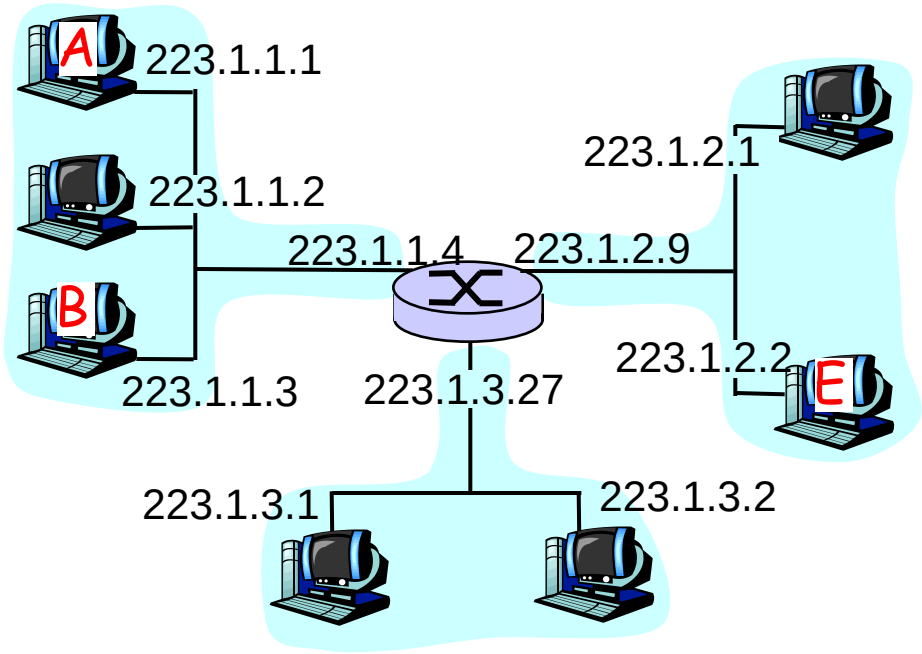
Roteamento de Pacotes IP – Exemplo

- **A** deseja enviar uma mensagem para **B**

pacote IP:

outros campos	endereço IP origem	endereço IP destino	dados

- Os endereços do pacote não mudam ao viajar da fonte ao destino



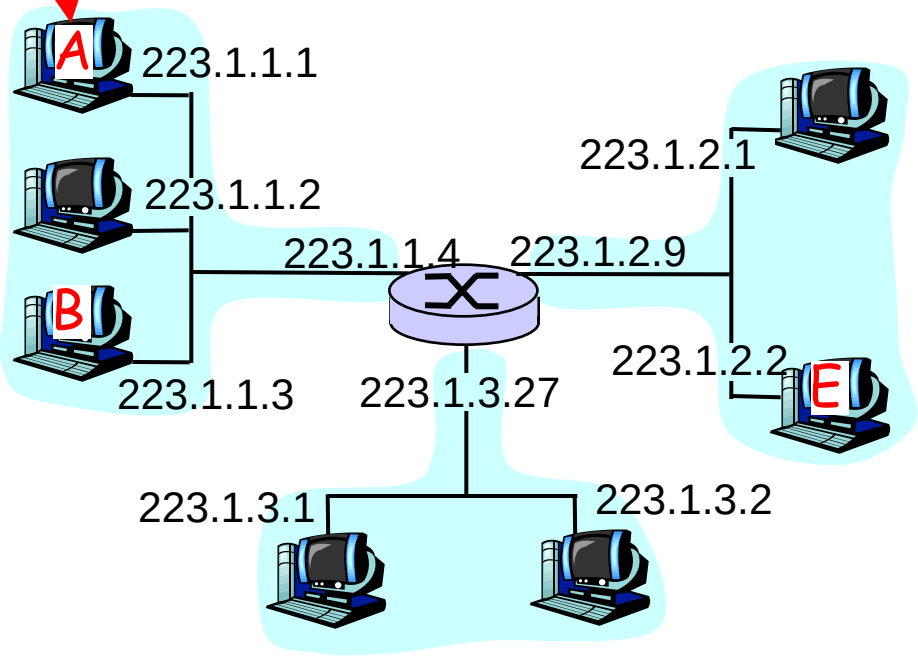
Roteamento de Pacotes IP – Exemplo

outros campos	223.1.1.1	223.1.1.3	dados

- Começando em A, levar pacote IP para B:
  - 223.1.1.3 AND 255.255.255.0 casa com 223.1.1.0
  - B está na mesma rede de A
  - Camada de enlace envia pacote diretamente para B em um quadro da camada de enlace

Tabela de roteamento em A

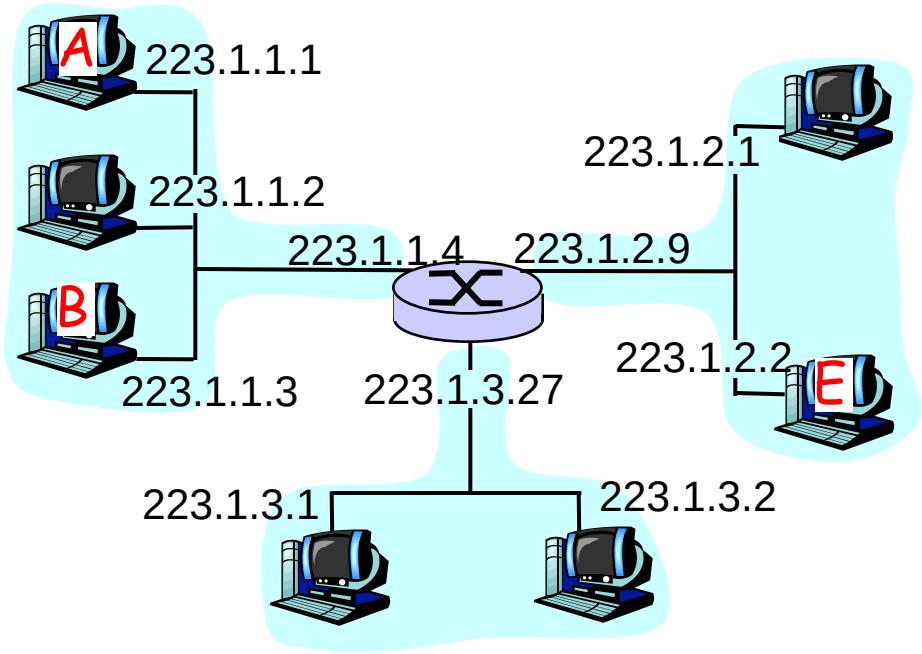
Rede destino	próx. roteador	Núm. saltos
223.1.1.0/24		1
223.1.2.0/24	223.1.1.4	2
223.1.3.0/24	223.1.1.4	2



Roteamento de Pacotes IP – Exemplo

- **A** deseja enviar uma mensagem para **E**

outros campos	223.1.1.1	223.1.2.2	dados



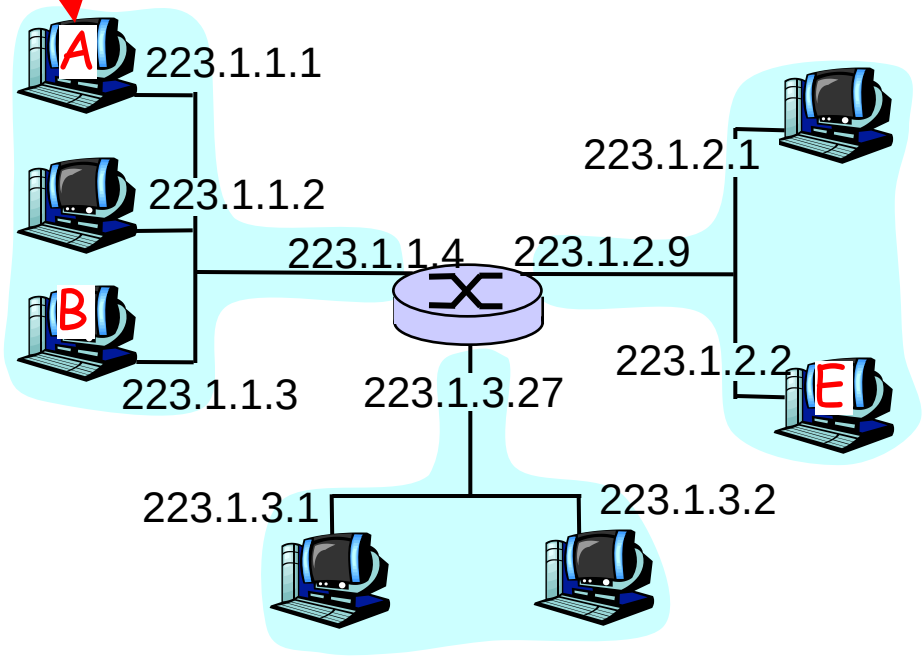
Roteamento de Pacotes IP – Exemplo

outros campos	223.1.1.1	223.1.2.2	dados

- Começando em A:
  - 223.1.2.2 AND 255.255.255.0 casa com 223.1.2.0
  - Próximo roteador para E é 223.1.1.4
  - Pacote chega em 223.1.1.4

Tabela de roteamento em A

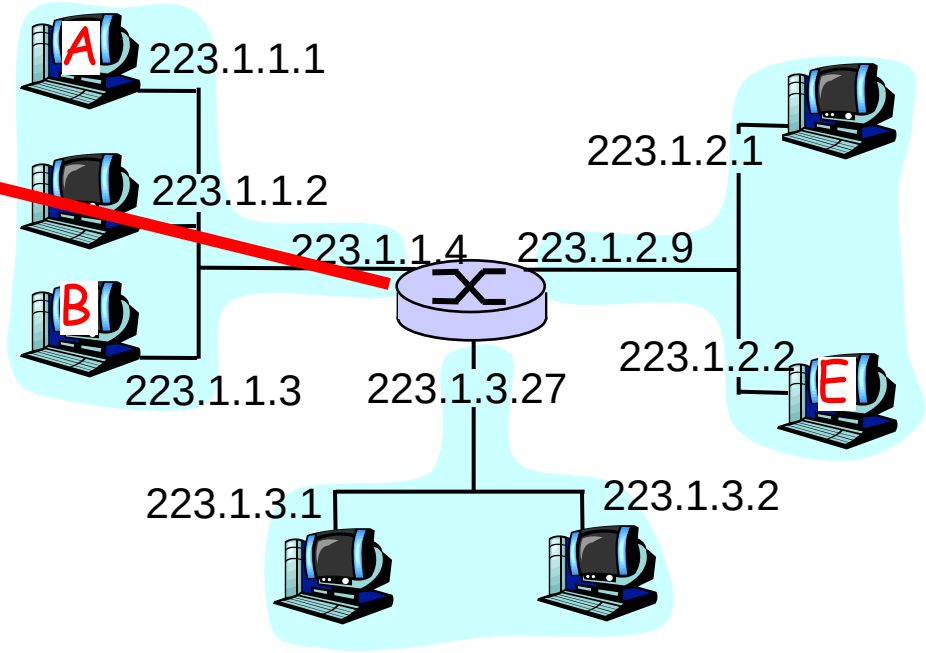
Rede destino	próx. roteador	Núm. saltos
223.1.1.0/24		1
223.1.2.0/24	223.1.1.4	2
223.1.3.0/24	223.1.1.4	2



Roteamento de Pacotes IP – Exemplo

Tabela de roteamento no roteador

Rede destino	Próx. roteador	Núm. saltos	Endereço Interface
223.1.1.0/24	-	1	223.1.1.4
223.1.2.0/24	-	1	223.1.2.9
223.1.3.0/24	-	1	223.1.3.27



- No roteador
  - 223.1.2.2 AND 255.255.255.0 casa com 223.1.2.0
  - Está na mesma rede da interface 223.1.2.9 do roteador
    - Roteador e E estão diretamente ligados
  - Envia o pacote para 223.1.2.2 através da interface 223.1.2.9

Exercício

5. Um roteador tem as seguintes entradas em sua tabela de roteamento:

Endereço/máscara	**Próximo hop**
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Roteador 1
padrão	Roteador 2

Para cada um dos endereços IP a seguir, o que o roteador fará se chegar um pacote com esse endereço?

a) 135.46.63.10 b) 135.46.57.14

c) 135.46.52.2 d) 192.53.40.7

e) 192.53.56.7

Endereçamento sem Classes

- A Internet está esgotando com rapidez os endereços IP disponíveis
- A prática de organizar o espaço de endereços por classes faz com que milhões deles sejam desperdiçados
  - Organização em classes:
    - Classe A: 128 redes de 16.777.216 hospedeiros
    - Classe B: 16.384 redes de 65.536 hospedeiros
    - Classe C: 2.097.152 redes de 256 hospedeiros
  - Um endereço classe B é grande demais para a maioria das organizações e um classe C é pequeno demais
  - Mais da metade de todas as redes da classe B tem menos de 50 hospedeiros

Endereçamento sem Classes

- Solução é usar endereçamento sem classes: CIDR (*Classless Interdomain Routing*)
  - Alocar os endereços IP restantes em blocos de tamanho variável, sem levar em consideração as classes
  - A parcela da rede de um endereço IP pode ter qualquer comprimento de bits, não ficando mais limitada a 8, 16 ou 24 bits
  - Se alguém precisar de 2.000 endereços, ele receberá um bloco de 2.048 endereços

Endereçamento sem Classes

- Exemplo:
  - Existem endereços disponível na rede 194.24.0.0
  - Universidade de Cambridge solicita 2.048 endereços
  - Universidade de Oxford solicita 4.096 endereços
  - Universidade de Edinburgh solicita 1.024 endereços

Endereçamento sem Classes

Universidade	Primeiro endereço	Último endereço	Quantidade	Escritos como
Cambridge	194.24.0.0	194.24.7.255	2.048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1.024	194.24.8.0/22
(Disponível)	194.24.12.0	194.24.15.255	1.024	194.24.12.0/22
Oxford	194.24.16.0	194.24.31.255	4.096	194.24.16.0/20

Endereçamento sem Classes

	Endereço	Máscara
Cam.	11000010 00011000 00000000 00000000	11111111 11111111 11111000 00000000
Edi.	11000010 00011000 00001000 00000000	11111111 11111111 11111100 00000000
Oxf.	11000010 00011000 00010000 00000000	11111111 11111111 11110000 00000000

- Roteamento para 194.24.17.4
  - 11000010 00011000 00010001 00000100
- Endereço AND Máscara de Cambridge
  - 11000010 00011000 00010000 00000000 (não casa com endereço de Cambridge)
- Endereço AND Máscara de Edinburgh
  - 11000010 00011000 00010000 00000000 (não casa com endereço de Edinburgh)

Endereçamento sem Classes

- Endereço AND Máscara de Oxford
  - 11000010 00011000 00010000 00000000 (casa com endereço de Oxford)
- Se não for encontrada nenhuma outra correspondência que utilize uma máscara com mais bits, o pacote será enviado para Oxford

Agregação de Endereços IP

- Em roteadores distantes, endereços associados à mesma linha de saída são agregados

	Endereço	Máscara
Cam.	11000010 00011000 00000000 00000000	11111111 11111111 11111000 00000000
Edi.	11000010 00011000 00001000 00000000	11111111 11111111 11111100 00000000
Oxf.	11000010 00011000 00010000 00000000	11111111 11111111 11110000 00000000

  - Estas três entradas podem ser agrupadas em 194.24.0.0/19:
    - Endereço: 11000010 00011000 00000000 00000000
    - Máscara: 11111111 11111111 11100000 00000000
- Agregação é muito utilizada em toda a Internet para reduzir o tamanho das tabelas de roteamento

Exercício

6. Um grande número de endereços IP consecutivos está disponível a partir de 198.16.0.0. Suponha que quatro organizações, A, B, C e D, solicitem 4.000, 2.000, 4.000 e 8.000 endereços, respectivamente, e nessa ordem. Para cada uma delas, forneça o primeiro endereço IP atribuído, o último endereço IP atribuído e a máscara na notação *w.x.y.z/s*.

Escassez de Endereços IP

- O IP está ficando sem endereços
- Solução:
  - NAT: *Network Address Translation*
  - Atribuir a cada empresa um único endereço IP (ou, no máximo, um número pequeno deles) para tráfego na Internet
  - Dentro da empresa, todo computador obtém um endereço IP exclusivo, usado para roteamento interno
  - Quando um pacote sai da empresa e vai para o ISP, ocorre uma conversão de endereço

NAT

- Para tornar o NAT possível, três intervalos de endereços IP foram declarados como privativos:
  - 10.0.0.0 a 10.255.255.255/8 (16.777.216 hosts)
  - 172.16.0.0 a 172.31.255.255/12 (1.048.576 hosts)
  - 192.168.0.0 a 192.168.255.255/16 (65.536 hosts)
- As empresas podem utilizar esta faixa de endereços internamente

NAT

- Funcionamento do NAT:
  - Dentro das instalações da empresa, toda máquina tem um endereço exclusivo da forma *10.x.y.z*
  - Quando um pacote deixa as instalações da empresa, ele passa por uma caixa NAT que converte o endereço interno (exemplo 10.0.0.1) no endereço IP verdadeiro da empresa (exemplo 198.60.42.12)
  - Quando a resposta volta, ela é endereçada ao IP verdadeiro (198.60.42.12), como a caixa NAT sabe por qual endereço deve substituir o endereço da resposta?

NAT

- Portas do TCP e UDP:
  - Quando um processo deseja estabelecer uma conexão TCP com um processo remoto, ele se associa a uma porta TCP não utilizada
  - As portas são inteiros de 16 bits
  - Essa porta é chamada porta de origem e informa ao código do TCP para onde devem ser enviados os pacotes que chegarem pertencentes a essa conexão
  - O processo também fornece uma porta de destino para informar a quem devem ser entregues os pacotes no lado remoto

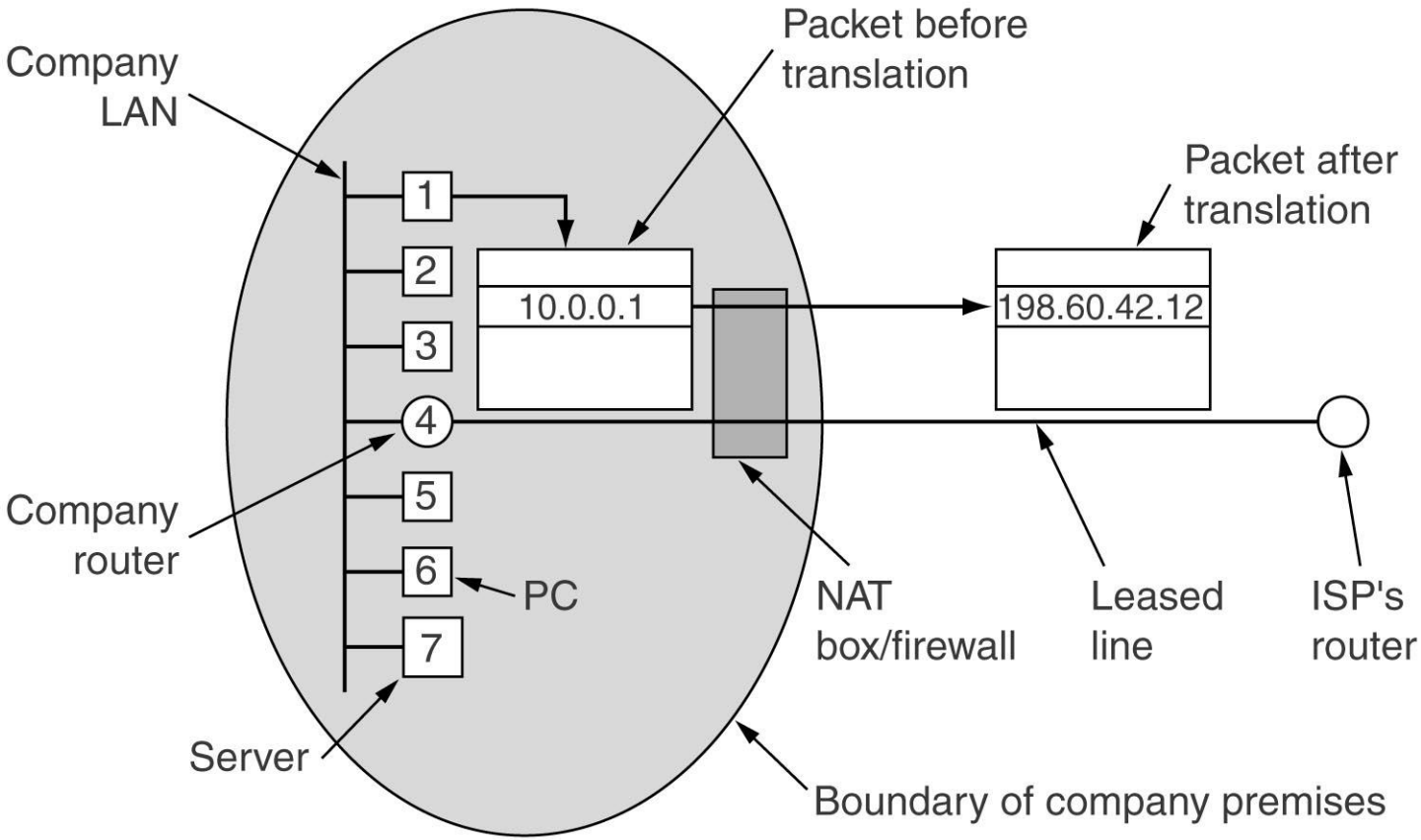
NAT

- Funcionamento do NAT:
  - Utilizam o campo de porta do UDP e do TCP
  - Sempre que um pacote de saída entra na caixa NAT, o endereço de origem *10.x.y.z* é substituído pelo endereço IP verdadeiro da empresa
  - O campo porta de origem do TCP é substituído por um índice para a tabela de conversão de 65.536 entradas da caixa NAT
  - Essa entrada de tabela contém a porta de origem e o endereço IP original

NAT

- Funcionamento:
  - Quando um pacote chega ao NAT vindo do ISP, o campo porta de destino do cabeçalho do TCP é extraído e usado como índice para a tabela de mapeamento da caixa NAT
  - A partir da entrada localizada, o endereço IP interno e o campo porta de origem do TCP original são extraídos e inseridos no pacote como endereço IP e porta de destino

NAT


Company LAN
Packet before translation
Packet after translation
1
2
3
4
5
6
7
10.0.0.1
198.60.42.12
Company router
PC
NAT box/firewall
Leased line
ISP's router
Server
Boundary of company premises

NAT

- Desvantagens:
  - A NAT viola o modelo arquitetônico do IP que estabelece que todo endereço IP identifica de forma exclusiva uma única máquina em todo o mundo
  - A NAT faz a Internet mudar suas características de rede sem conexões para uma espécie de rede orientada a conexões (NAT mantém informações sobre as conexões)
  - Os processos na Internet não são obrigados a usar o TCP ou o UDP

NAT

- Desvantagens
  - A NAT viola a regra mais fundamental da distribuição de protocolos em camadas: a camada *k* não pode fazer quaisquer suposições sobre o que a camada *k+1* inseriu no campo de carga útil
  - Algumas aplicações inserem endereços IP ou números de porta no corpo do texto (FTP)
  - No máximo 61.440 (65.536 – 4.096 portas reservadas) máquinas podem ser mapeadas em um endereço IP

Exercício

7. Tanto o NAT quanto o endereçamento sem classes da Internet (CIDR) foram projetados devido à escassez de endereços IP disponíveis Explique o funcionamento básico destas duas técnicas e a principal diferença entre elas.
8. Suponha que uma empresa recebeu o endereço 192.214.11.0/24 e ela necessita criar 3 subredes: 1 subrede com 126 hosts e 2 subredes com 62 hosts cada. Como você organizaria internamente a rede? Indique a máscara e os endereços de rede e de broadcast de cada subrede criada.

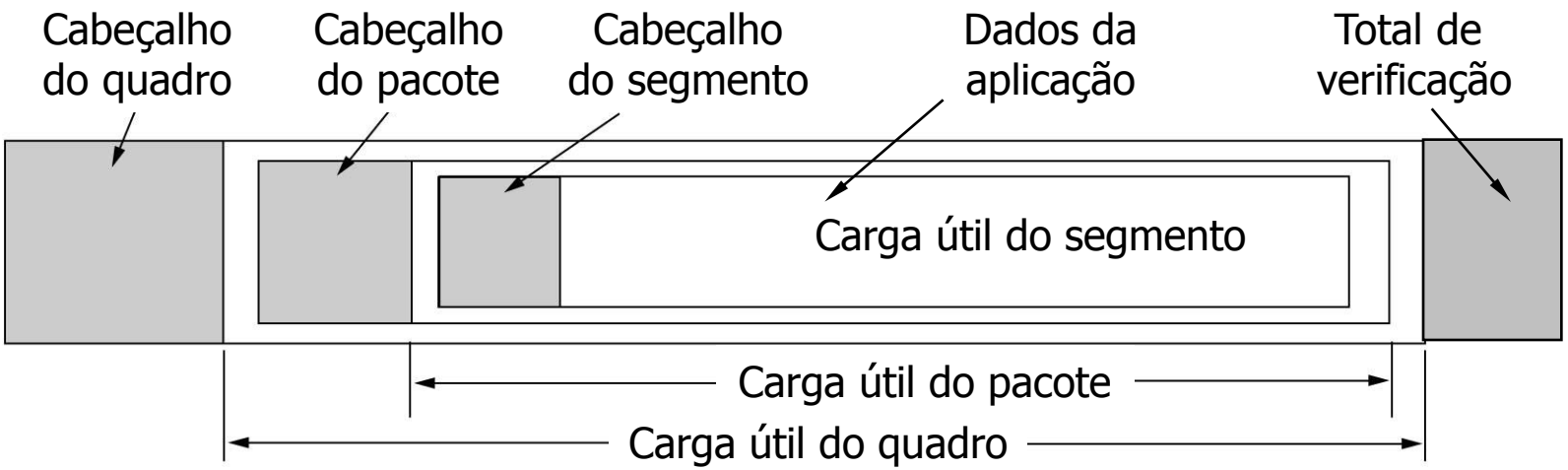
Exercício

9. Uma rede na Internet tem uma máscara de sub-rede 255.255.240.0 e possui um computador com IP 200.20.88.9 conectado a ela. Qual é o número máximo de computadores que podemos conectar nesta rede? Qual é o endereço de broadcast desta rede? Qual é o seu endereço de rede?
10. Ao subdividir uma rede de máscara 255.255.255.128, quais novas máscaras seriam possíveis criar para no mínimo 9 endereços em cada subrede? Considere que todas as subredes terão o mesmo número de hosts.

Camada de Transporte

A Camada de Transporte

- A unidade de dados trocada entre entidades da camada de transporte é chamada de segmento
- Aninhamento de segmentos, pacotes e quadros:

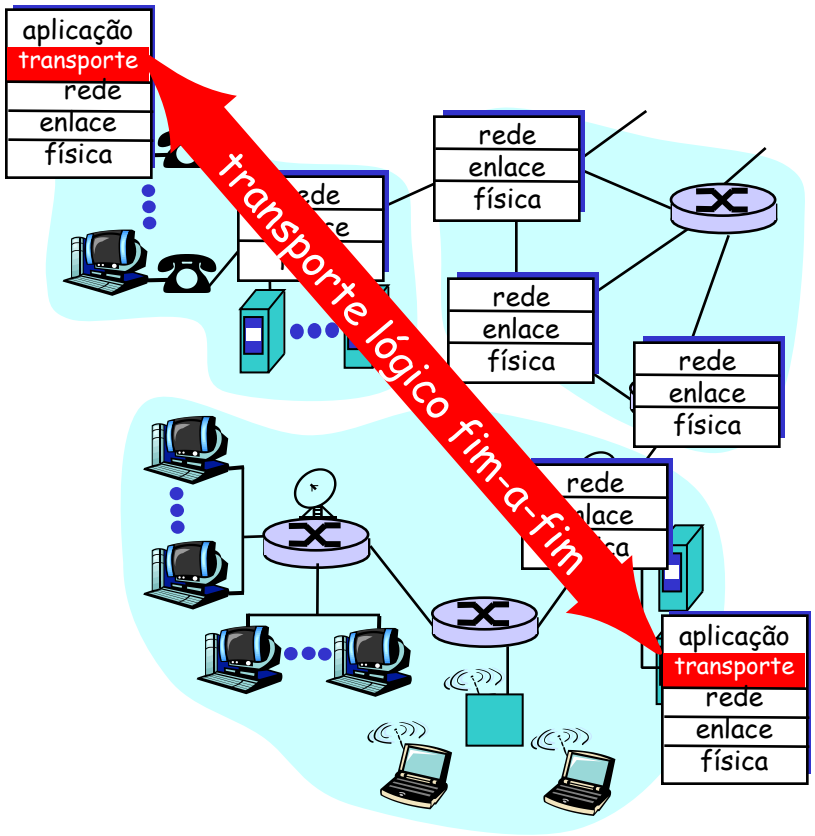


A Camada de Transporte

- Fornece comunicação lógica entre processos de aplicação em hospedeiros diferentes (multiplexação/demultiplexação de aplicação)
- Os protocolos de transporte são executados nos sistemas finais da rede

Camada de Rede	**Camada de Transporte**
Transferência de dados entre sistemas finais	Transferência de dados entre processos
Funciona principalmente nos roteadores	Funciona inteiramente nas máquinas dos usuários

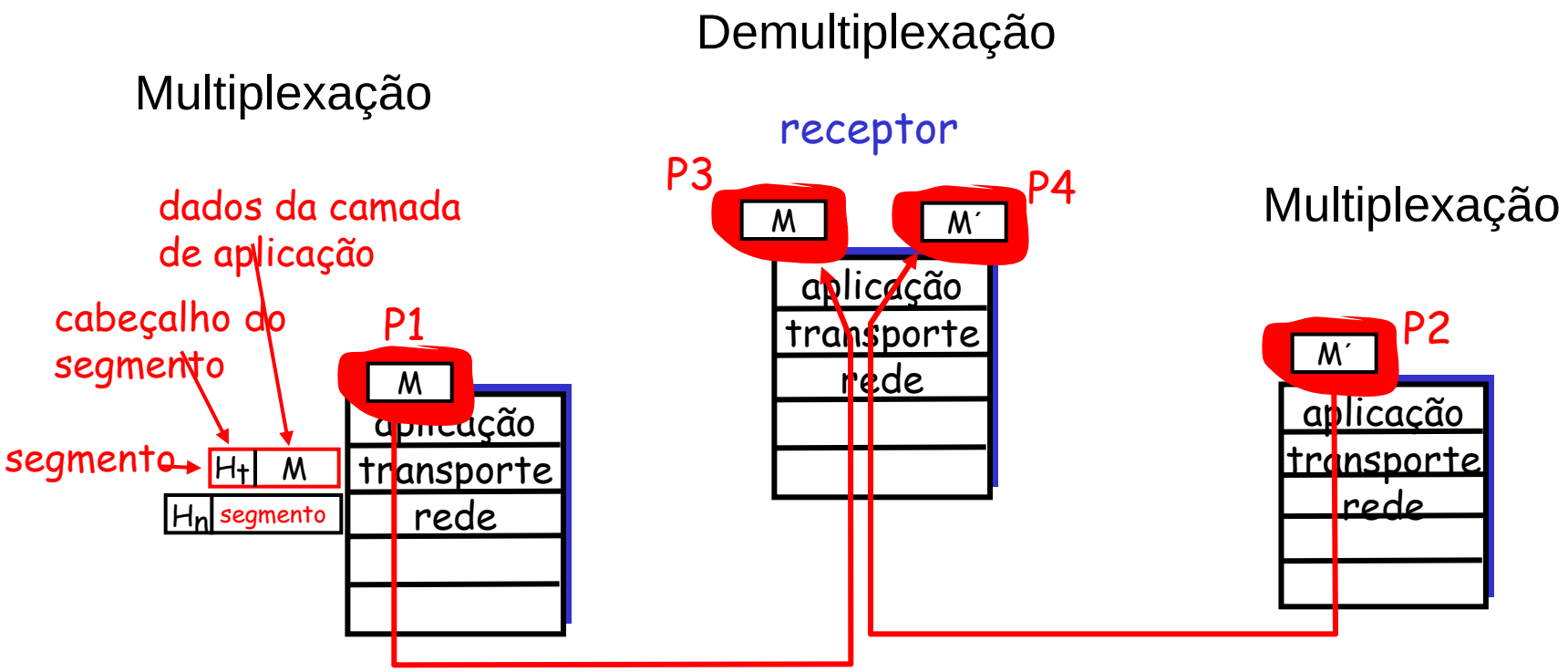
A Camada de Transporte



Multiplexação e Demultiplexação de Aplicações

- Objetivo:
  - Ampliar a entrega hospedeiro a hospedeiro para entrega aplicação a aplicação
- Multiplexação: reunir os dados provenientes de diferentes processos de aplicação (ocorre no hospedeiro de origem)
- Demultiplexação: entrega dos dados contidos em um segmento da camada de transporte ao processo de aplicação correto (ocorre no hospedeiro de destino)

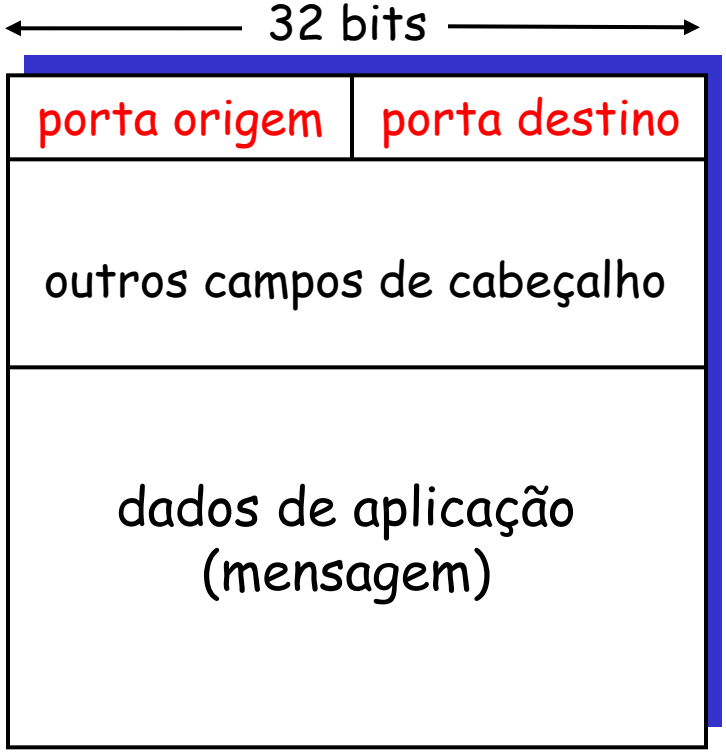
Multiplexação e Demultiplexação de Aplicações



Multiplexação e Demultiplexação de Aplicações

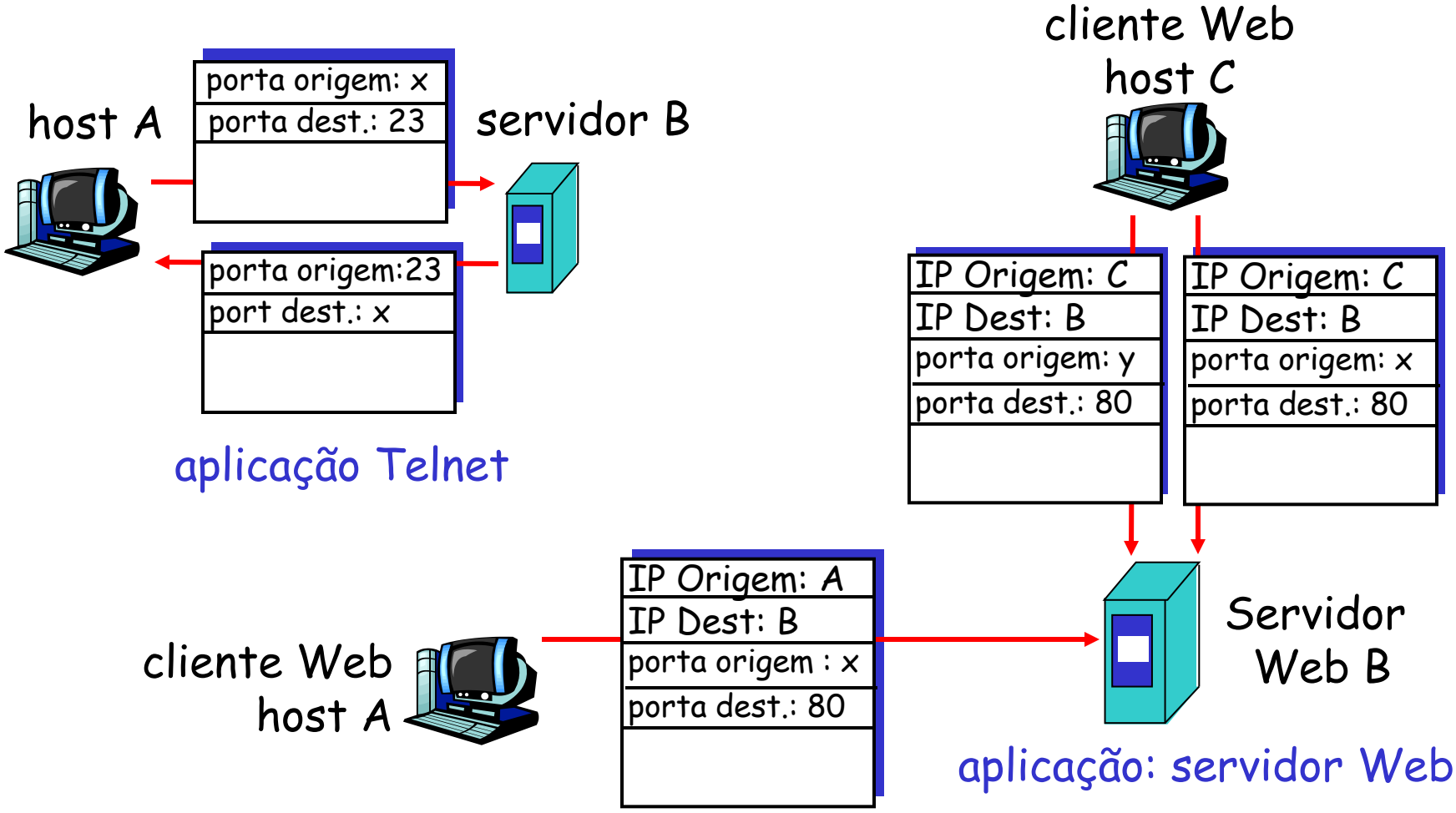
Multiplexação / Demultiplexação:

- Baseada nos número de porta da origem, número de porta do destino e endereços IP
- Os números de porta entre 0 e 1023 são chamados de números de porta bem conhecidos (reservados para uso por protocolos de aplicação bem conhecidos como HTTP e FTP)



formato do segmento TCP/UDP

Multiplexação e Demultiplexação de Aplicações: Exemplos



Protocolos da Camada de Transporte da Internet

- A Internet tem dois protocolos principais na camada de transporte
  - UDP
    - Sem conexões
    - Transferência não confiável de dados
  - TCP
    - Orientado à conexões
    - Transferência confiável de dados
    - Controle de congestionamento
    - Controle de fluxo

UDP

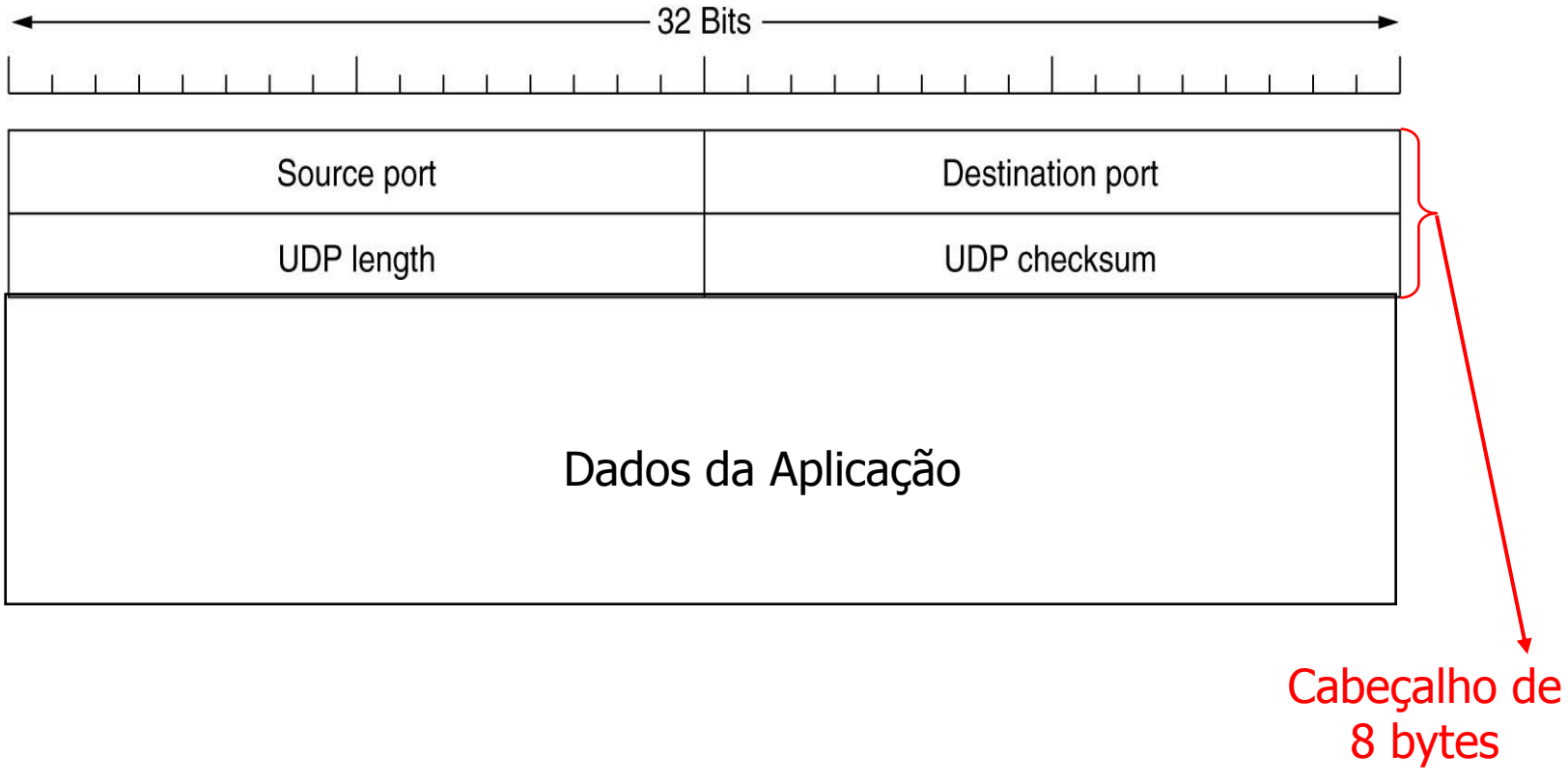
User Datagram Protocol

[RFC 768]

UDP

- Protocolo de transporte não confiável
- Não há estabelecimento de conexão
  - Não há apresentação entre o UDP transmissor e o receptor
  - Cada segmento UDP é tratado de forma independente dos outros
- Serviço *best-effort*, segmentos UDP podem ser:
  - Perdidos
  - Entregues fora de ordem para a aplicação

Estrutura do Segmento UDP



Estrutura do Segmento UDP

- *Source Port* e *Destination Port*:
  - Identificam os pontos extremos (processos) nas máquinas de origem e destino
  - Quando um pacote UDP chega, sua carga útil é entregue ao processo associado à porta destino
- UDP *length*:
  - Tamanho total do segmento (incluindo o cabeçalho) em bytes
- UDP *checksum*:
  - Campo opcional, armazenado como 0 se não for calculado

Estrutura do Segmento UDP

- UDP *checksum* :
  - Transmissor:
    - Calcula o complemento de 1 da soma de todas as palavras de 16 bits do segmento
  - Receptor:
    - Computa o total de verificação do segmento recebido
    - Verifica se o total de verificação é igual ao valor do campo no cabeçalho
      - Não: erro detectado
      - Sim: não há erros

UDP

- O UDP não faz:
  - Controle de fluxo
  - Controle de erros
  - Retransmissão após a recepção de um segmento incorreto
- O UDP apenas fornece uma interface para o protocolo IP com o recurso adicional de multiplexação/demultiplexação de vários processos que utilizam as portas

UDP

- Porque existe um UDP?
  - Não há estabelecimento de conexão e portanto não introduz nenhum atraso
  - Não há estado de conexão nem no transmissor e nem no receptor
  - Pequeno *overhead* no cabeçalho do pacote
  - Taxa de envio não regulada (não há controle de congestionamento)

UDP

- Onde é usado?
  - Aplicações cliente-servidor onde existem apenas uma requisição e uma resposta
  - O custo para estabelecer uma conexão é alto quando comparado com a transferência de dados
  - Aplicações de multimídia contínua (*streaming*)
    - Tolerantes à perda
    - Sensíveis à taxa

Exercícios

11. Por que o UDP existe? Não teria sido suficiente deixar que os processos dos usuários enviassem pacotes IP brutos?

12. Ao subdividir uma rede de máscara 255.255.255.128, quais novas máscaras seriam possíveis criar para no mínimo 14 máquinas em cada sub-rede?

TCP

Transport Control Protocol

RFCs: 793, 1122, 1323, 2018, 2581

TCP

- Fornece um fluxo de bytes fim-a-fim confiável em uma sub-rede não confiável
- Foi projetado para se adaptar dinamicamente às propriedades da sub-rede e ser robusto diante dos muitos tipos de falhas que podem ocorrer
- Principal protocolo de transporte da arquitetura TCP/IP
  - A maior parte das aplicações na Internet são baseadas no TCP

TCP

- Características
  1. Transferência confiável de dados
     - TCP garante que os dados serão entregues da forma que foram enviados
  2. Orientado à conexões
     - Conexões são gerenciadas nos sistemas finais
  3. Controle de fluxo
     - Transmissor não esgota a capacidade do receptor

TCP

- Características
  4. Controle de congestionamento
     - Transmissor não esgota os recursos da rede
  5. Gerenciamento de *timers*
     - O TCP necessita de vários *timers* para realizar seu trabalho

O Modelo de Serviço do TCP

- A camada TCP aceita fluxos de dados das aplicações, dividi-os em partes de no máximo 65.495 bytes (fluxo de bytes e não fluxo de mensagens)
  - Na prática, com frequência, temos 1.460 bytes de dados, para que ele possa caber em um único quadro Ethernet com os cabeçalhos IP e TCP
- Envia cada parte para a camada IP como um segmento distinto
- Ao chegar à camada IP da máquina destino, os dados são entregues à entidade TCP, que restaura os fluxos de bytes originais

O Modelo de Serviço do TCP

- A entidade TCP receptora retorna um segmento (com ou sem dados) com um número de confirmação igual ao próximo número de sequência que espera receber
- Se o *timer* do transmissor expirar antes de a confirmação ser recebida, o segmento será retransmitido

O Modelo de Serviço do TCP

- Usa um serviço não confiável para prover um serviço de entrega de dados confiável para as aplicações
- Deve ser capaz de compensar perdas e atrasos na sub-rede de comunicação de tal forma a prover o transporte de dados fim-a-fim de forma eficiente
- Deve ser capaz de executar essas tarefas sem sobrecarregar a sub-rede de comunicação e os roteadores

O Modelo de Serviço do TCP

- O serviço TCP é obtido quando tanto o transmissor quanto o receptor criam pontos extremos chamados **soquetes**
  - Identificação do soquete = endereço IP do host (32 bits) + número de porta (16 bits)
  - As portas abaixo de 1024 são denominadas portas bem conhecidas (*well-known ports*) e são reservadas para serviços padrão

O Modelo de Serviço do TCP

- O serviço TCP necessita que uma **conexão** seja explicitamente estabelecida entre um soquete da máquina transmissora e um soquete da máquina receptora
  - Estabelecimento da conexão
  - Transferência de dados
  - Término da conexão

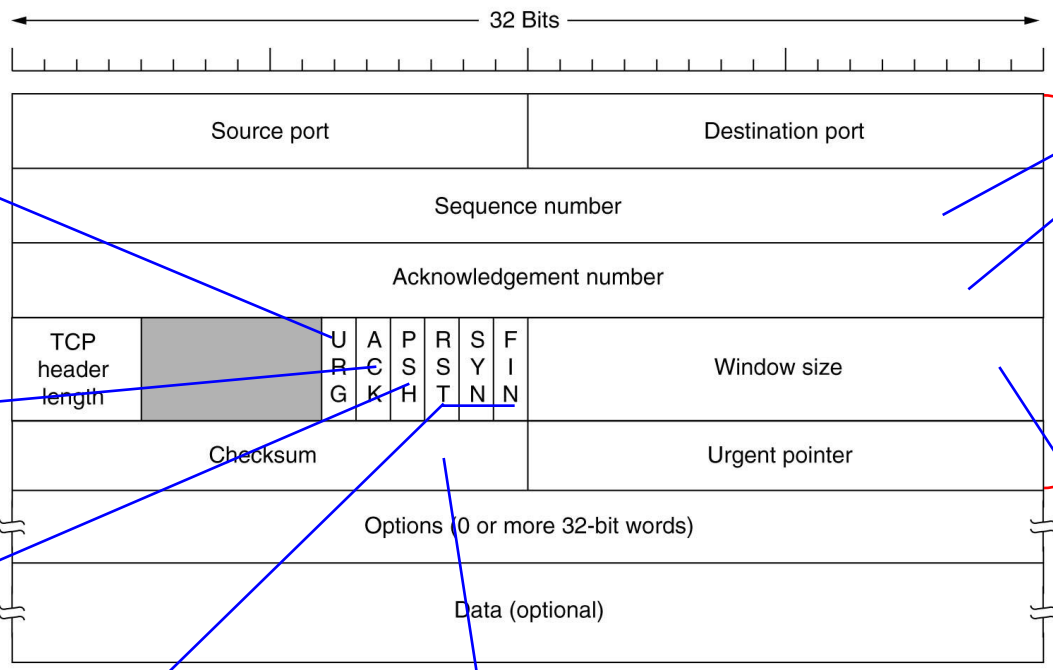
O Modelo de Serviço do TCP

- As conexões são identificadas pelos soquetes das máquinas transmissora e receptora, ou seja, cada conexão é identificada por:
  - Endereço IP origem
  - Porta origem
  - Endereço IP destino
  - Porta destino
- Um soquete pode ser utilizado por várias conexões ao mesmo tempo (duas ou mais conexões podem terminar no mesmo soquete)

O Modelo de Serviço do TCP

- Todas as conexões TCP são *full-duplex* e ponto a ponto
  - Tráfego pode ser feito em ambas as direções ao mesmo tempo
  - Cada conexão possui exatamente dois pontos terminais
  - TCP não admite multidifusão e difusão

O Segmento TCP



contagem por bytes de dados (não segmentos!)

20 bytes

URG: dados urgentes (pouco usado)

ACK: campo de ACK é válido

PSH: produz envio de dados (pouco usado)

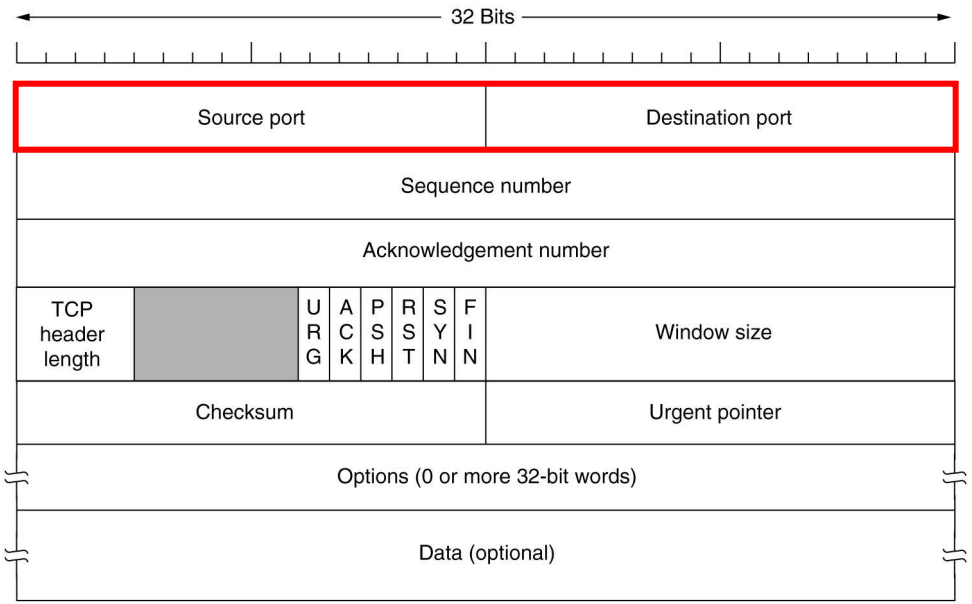
RST, SYN, FIN: estabelecimento de conexão (comandos de criação e término)

Total de verificação do segmento

número de bytes que o receptor está pronto para aceitar

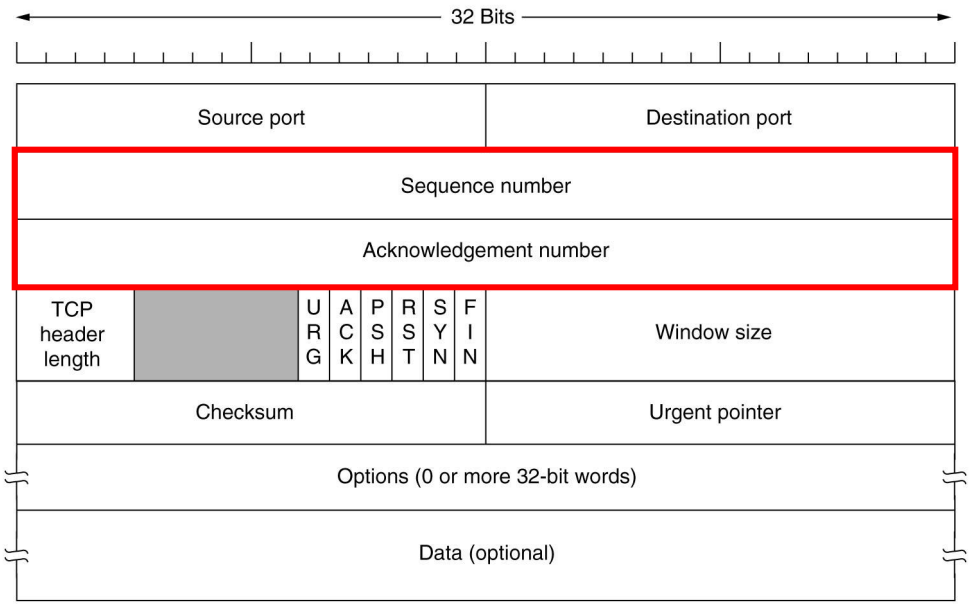
O Segmento TCP

- *Source port* (16 bits) e *Destination port* (16 bits):
  - Identificam os pontos terminais locais da conexão (portas)



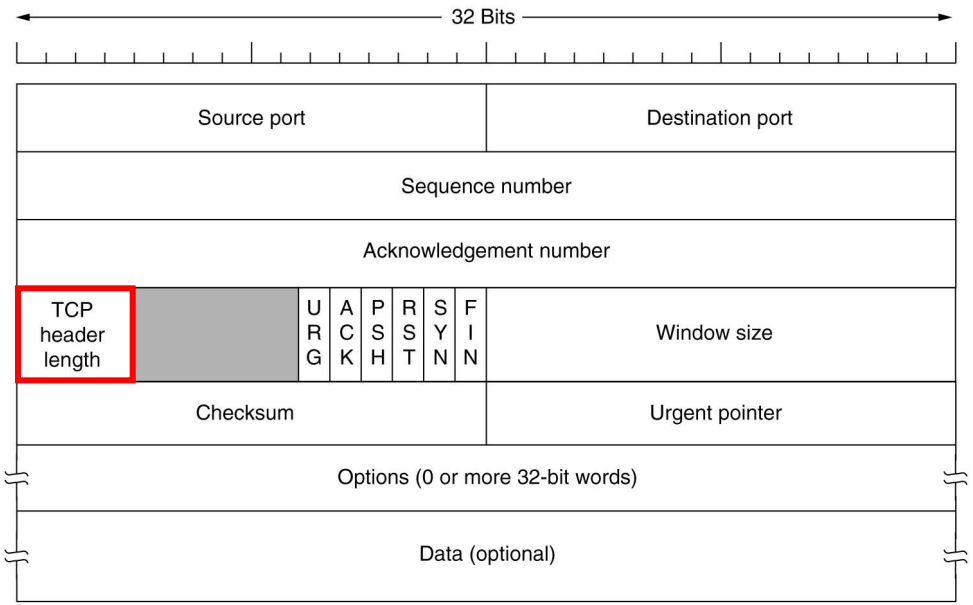
O Segmento TCP

- *Sequence number* (32 bits) e *Acknowledgement number* (32 bits):
  - Transferência confiável de dados



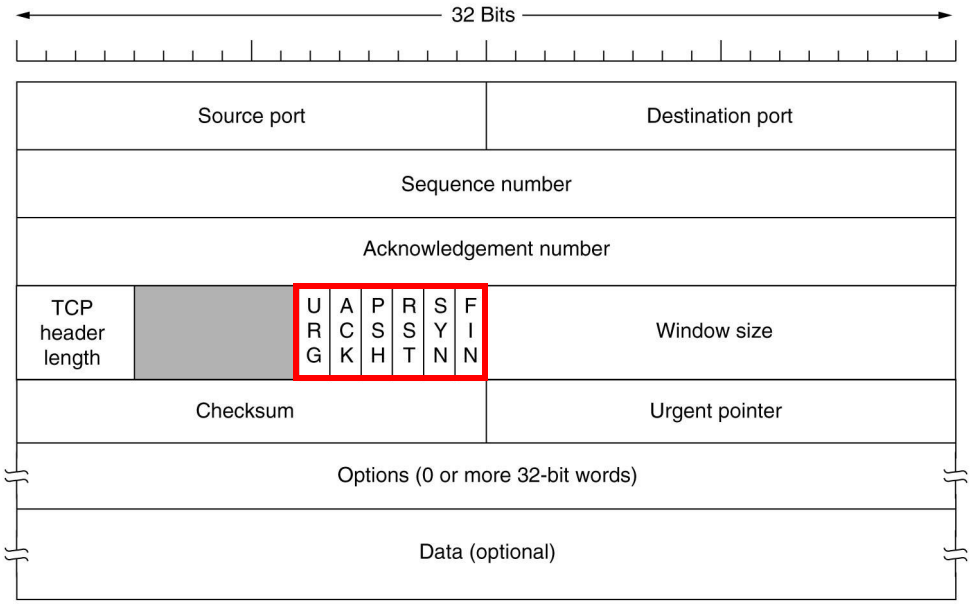
O Segmento TCP

- TCP *header length* (4 bits):
  - Informa quantas palavras de 32 bits existem no cabeçalho TCP



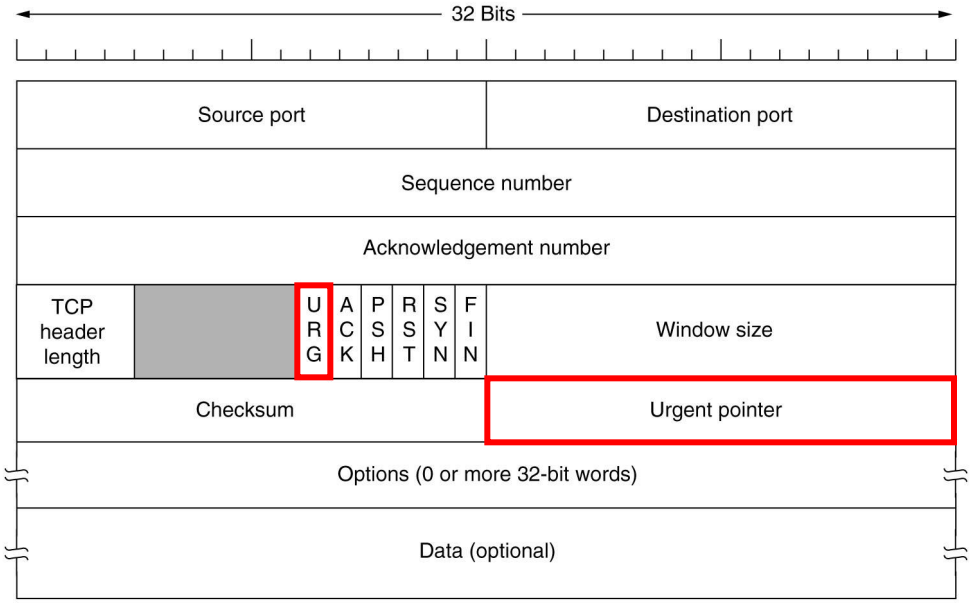
O Segmento TCP

- Seis Flags (URG, ACK, PSH, PST, SYN e FIN):



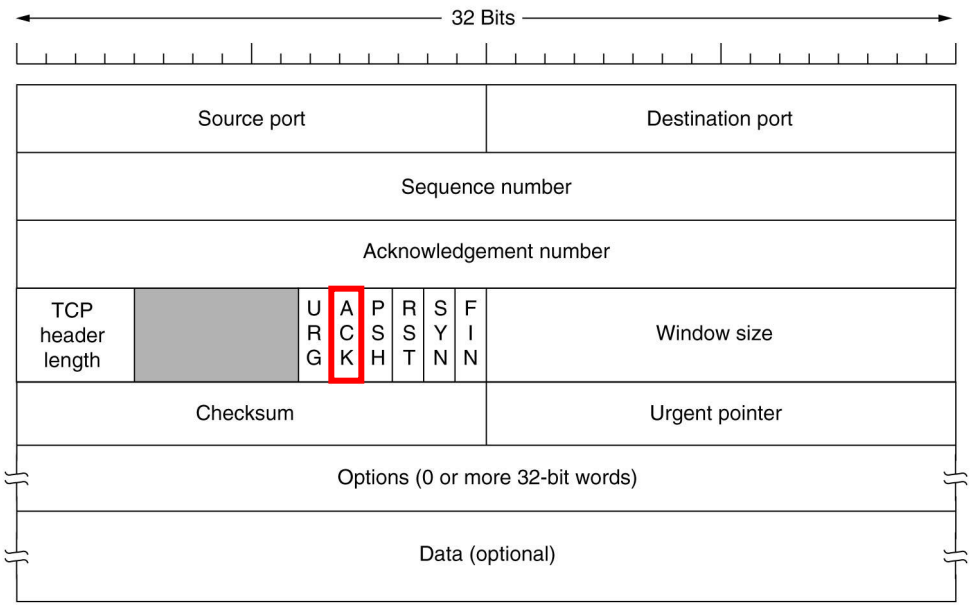
O Segmento TCP

- Bit URG e *Urgent pointer* (16 bits):
  - URG = 1, se *Urgent pointer* estiver sendo usado
  - *Urgent pointer* indica o deslocamento de bytes para os dados urgentes (mensagens de interrupção)



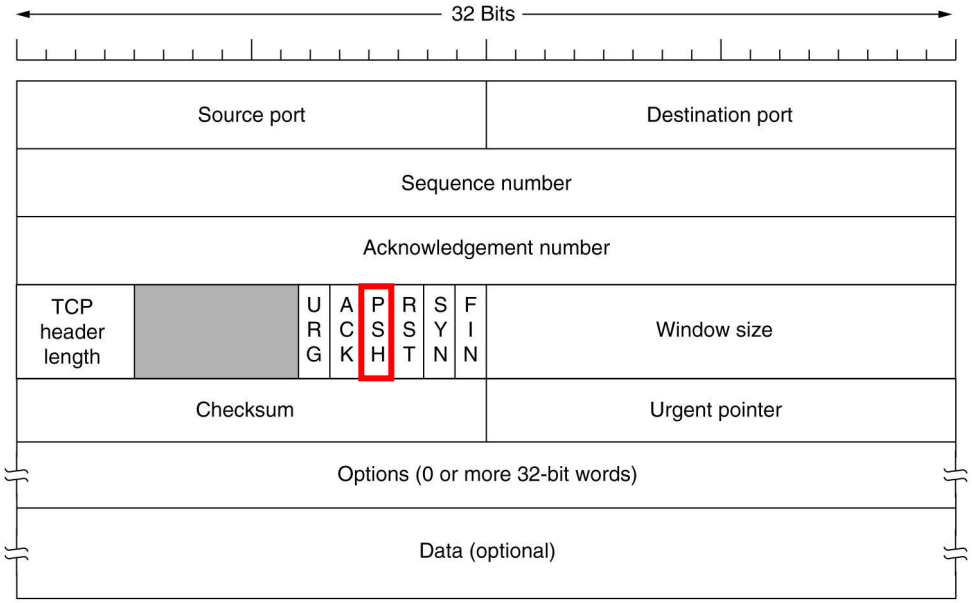
O Segmento TCP

- Bit ACK:
  - ACK = 1, se o campo *Acknowledgement number* é válido
  - Se ACK = 0, o segmento não contém uma confirmação



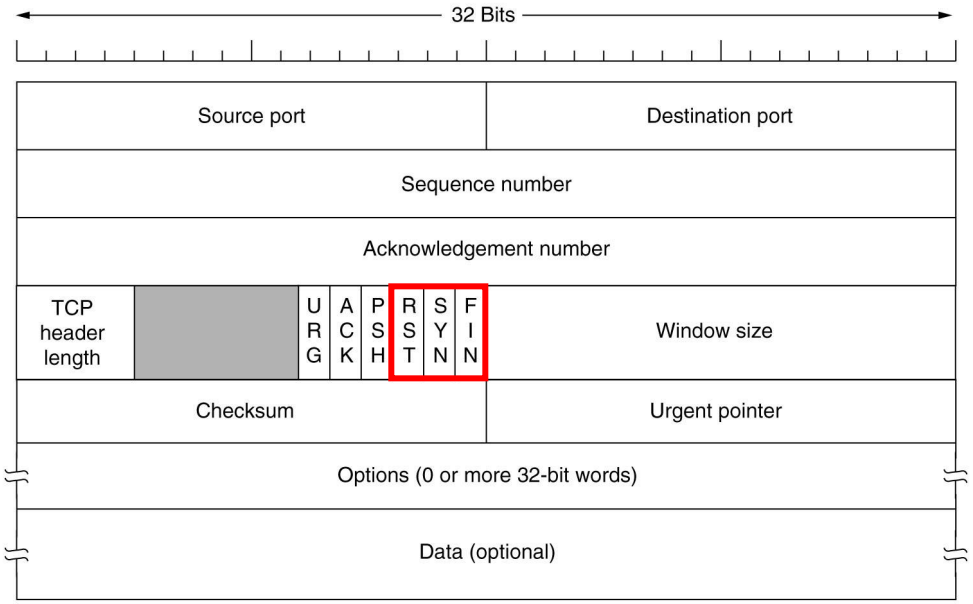
O Segmento TCP

- Bit PSH (pouco utilizado):
  - Informa ao TCP para não retardar a transmissão
  - O receptor é solicitado a entregar os dados à aplicação imediatamente



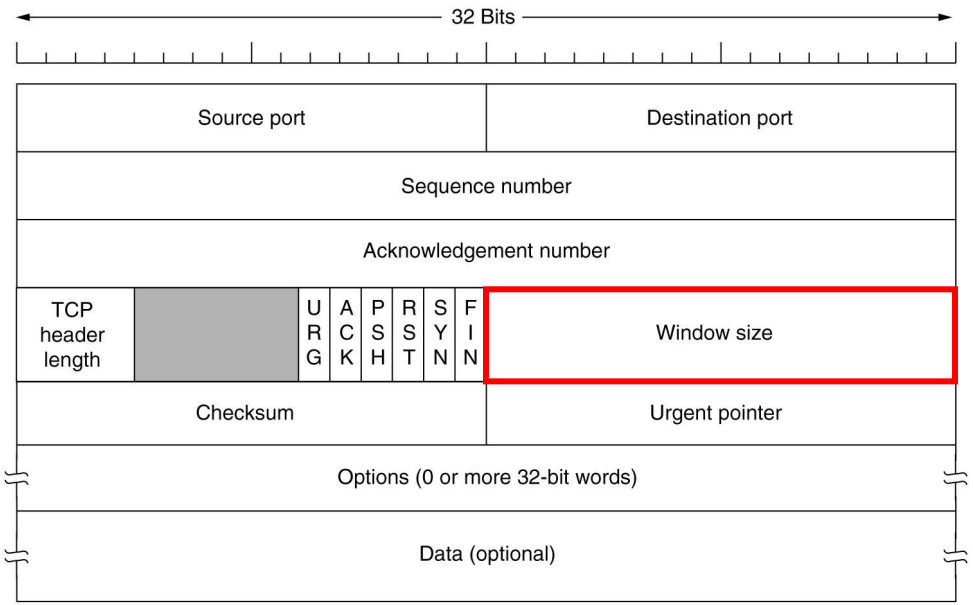
O Segmento TCP

- Bits RST, SYN e FIN:
  - Gerenciamento de conexões



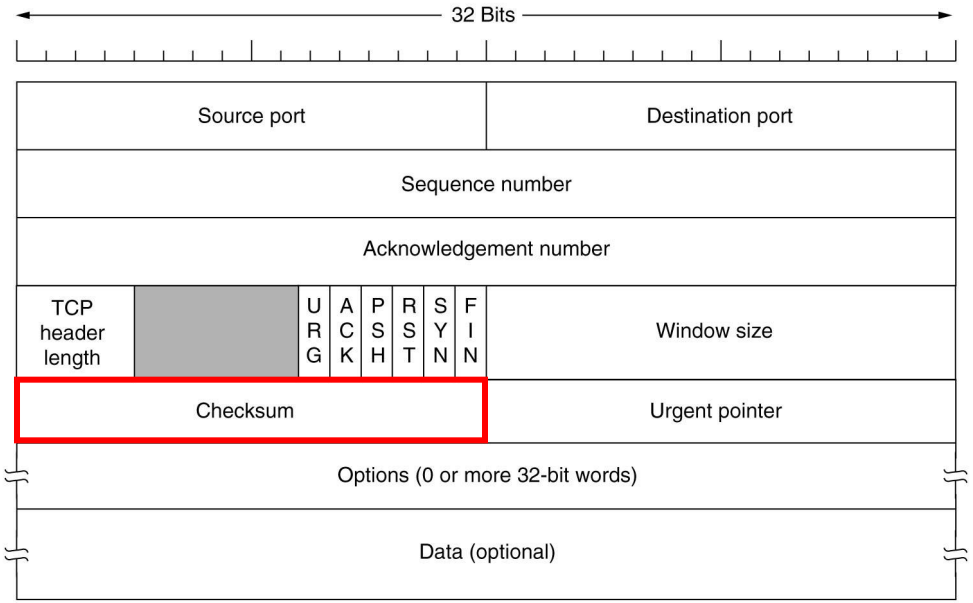
O Segmento TCP

- *Window size* (16 bits):
  - Controle de fluxo



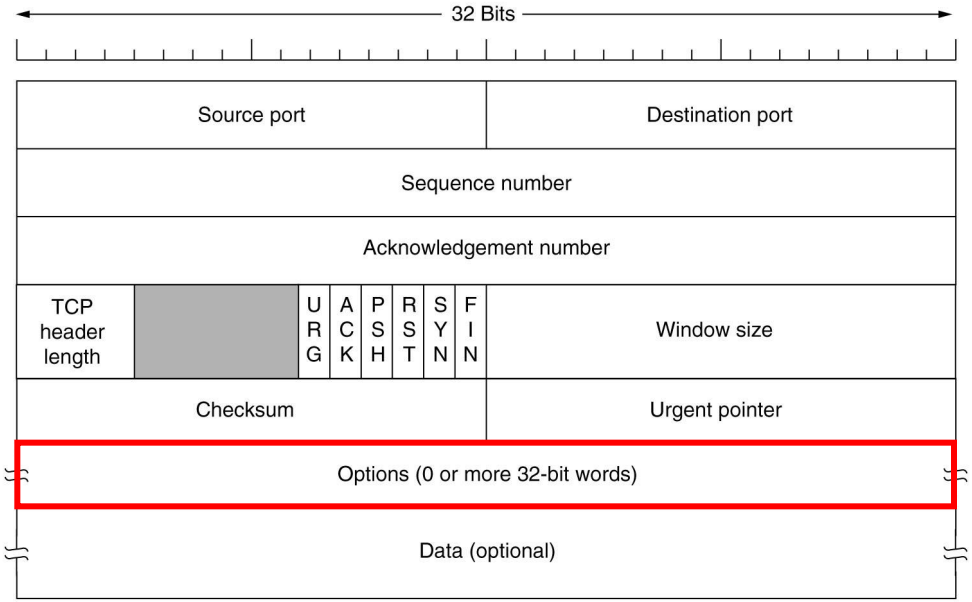
O Segmento TCP

- *Checksum* (16 bits):
  - Total de verificação do segmento



O Segmento TCP

- *Options* (no máximo 40 bytes):
  - Forma de oferecer recursos que não foram previstos pelo cabeçalho comum (exemplo: negociação entre os hosts para estipular o máximo de carga útil do TCP)



Exercício

13. Qual é a quantidade máxima de carga útil que o TCP e o UDP podem colocar em um segmento de forma que ele não seja fragmentado? Considere um MTU na camada de enlace de 576 bytes.

1. Transferência Confiável de Dados

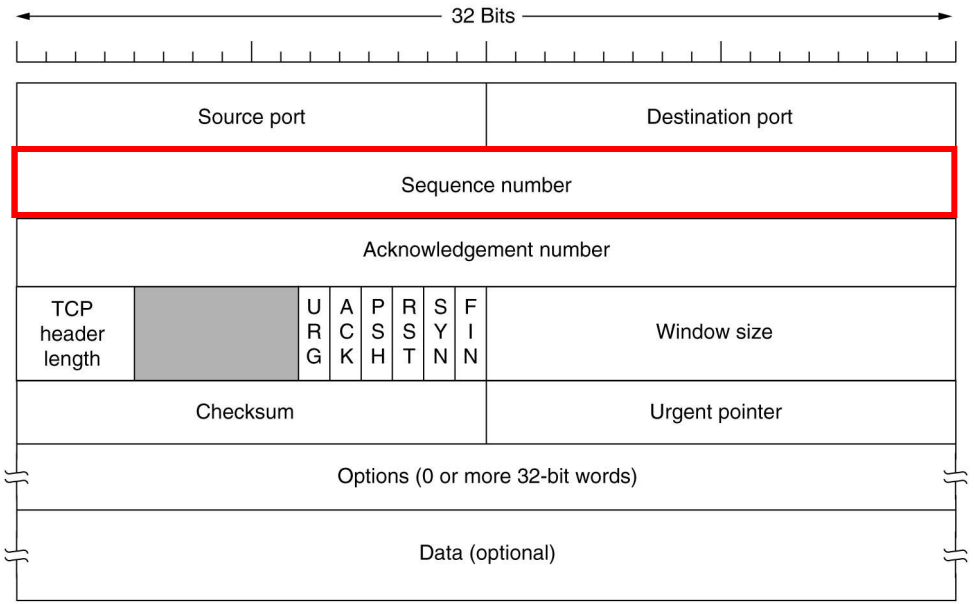
- Garante que a cadeia de dados que um processo lê a partir de seu buffer de recebimento TCP:
  - Não está corrompida
  - Não tem lacunas
  - Não possui duplicações
  - Está em sequência
- A cadeia de bytes é exatamente a mesma cadeia de dados enviada pelo sistema final que está do outro lado da conexão

1. Transferência Confiável de Dados

- O TCP vê os dados como uma cadeia de bytes desestruturada, mas ordenada
- Cada byte em uma conexão TCP tem seu próprio número de sequência de 32 bits

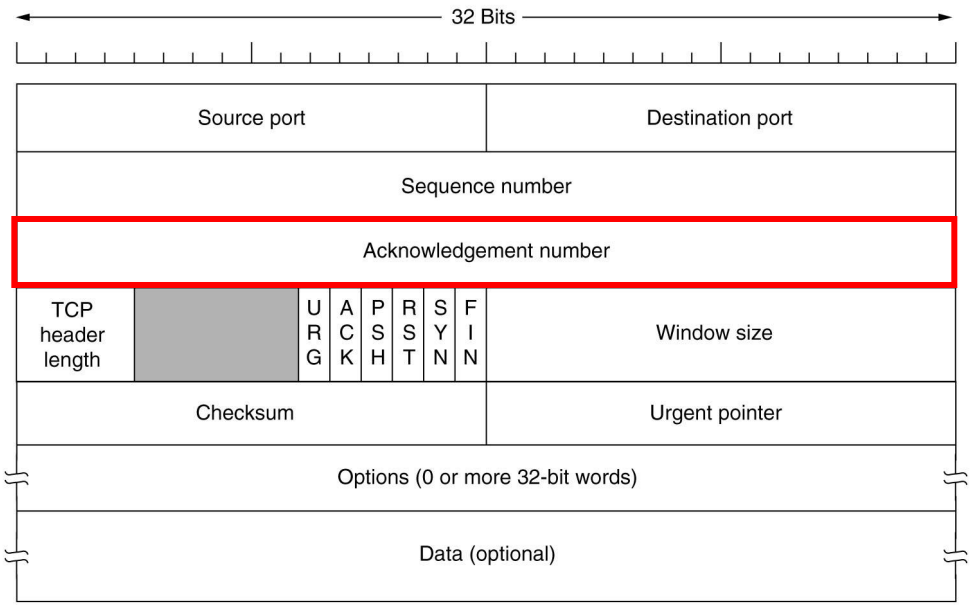
1. Transferência Confiável de Dados

- *Sequence number:*
  - Número do primeiro byte no segmento de dados
  - Aplicados sobre a cadeia de bytes transmitidos e não sobre a série de segmentos transmitidos

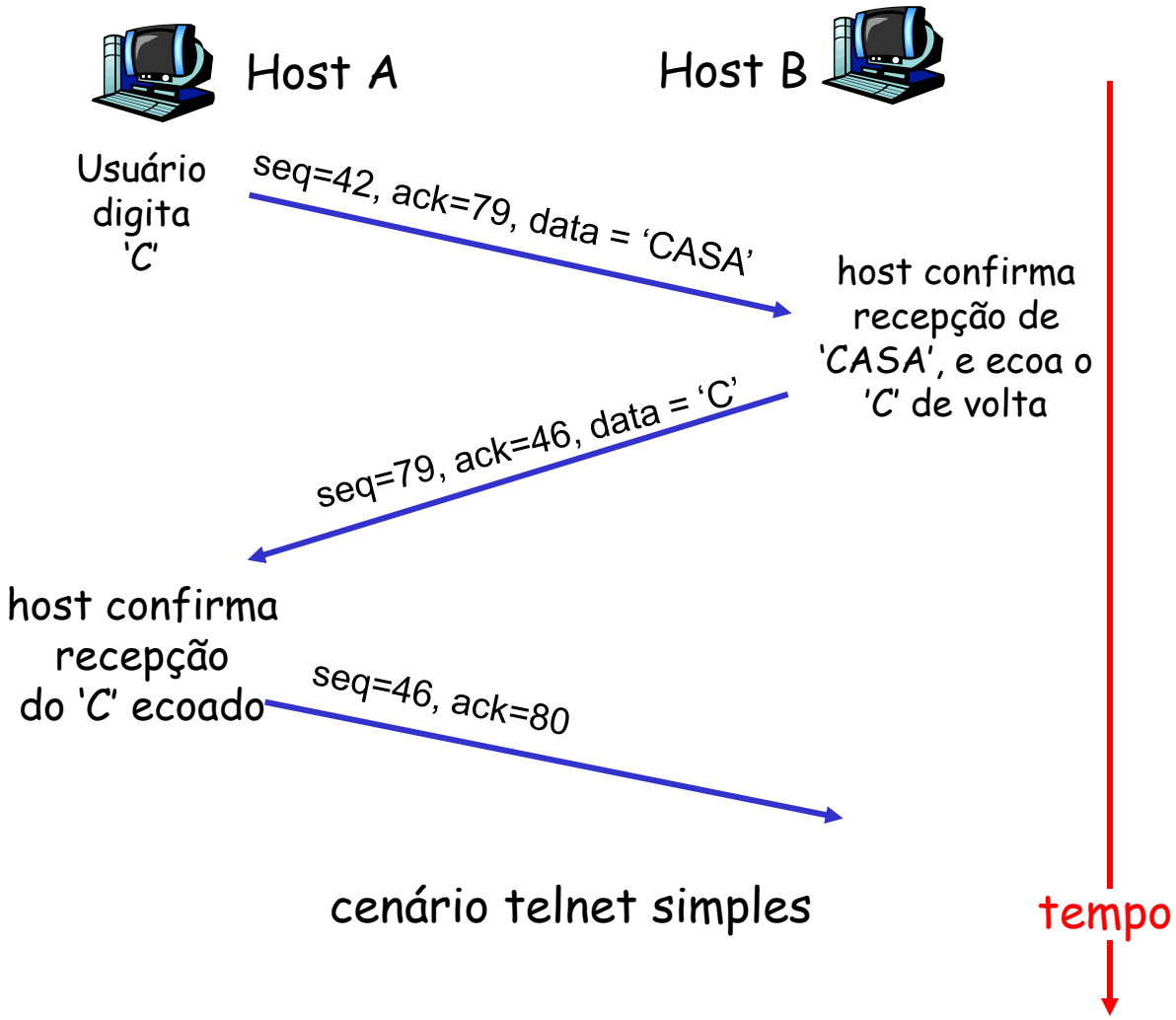


1. Transferência Confiável de Dados

- *Acknowledgement number* :
  - Especifica o número do próximo byte esperado e não o último byte recebido corretamente



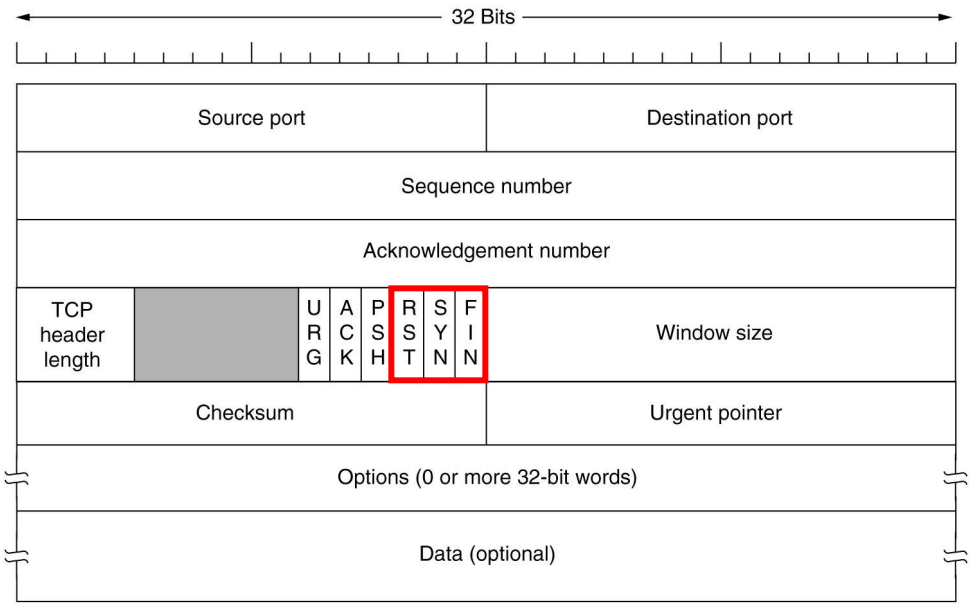
1. Transferência Confiável de Dados



cenário telnet simples

2. Gerenciamento da Conexão TCP

- Utilizam os bits RST, SYN e FIN do cabeçalho TCP



2. Gerenciamento da Conexão TCP

- Bit RST:
  - Recusar uma tentativa de conexão
- Bit SYN:
  - Estabelecer conexões
- Bit FIN:
  - Finalizar a conexão

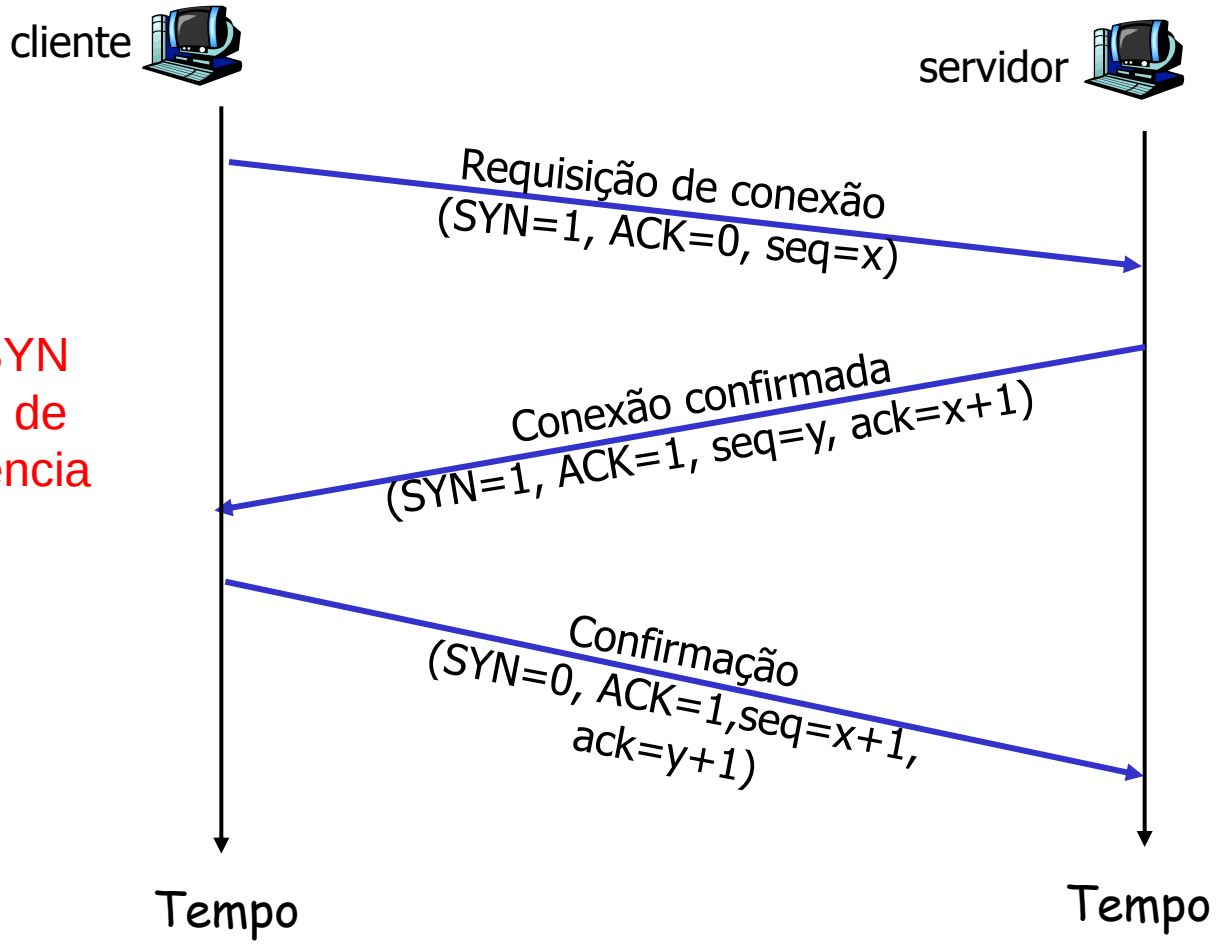
2. Gerenciamento da Conexão TCP

- TCP transmissor estabelece uma conexão com o receptor antes de trocar segmentos de dados
- As conexões são estabelecidas no TCP por meio do *handshake* de três vias (3-*way handshake*)
- Inicialização de variáveis:
  - Números de sequência
  - *Buffers*, controle de fluxo

2. Gerenciamento da Conexão TCP

- Estabelecimento da conexão TCP:
  - Passo 1: cliente envia um segmento com SYN=1 e ACK=0 ao servidor
    - Especifica o número de sequência inicial
  - Passo 2: servidor responde com o segmento SYNACK (SYN=1, ACK=1)
    - Reconhece o SYN recebido
    - Aloca *buffers*
    - Especifica o número de sequência inicial do servidor
  - Passo 3: cliente recebe o SYNACK

2. Gerenciamento da Conexão TCP



Um segmento SYN consome 1 byte de espaço de sequência

Estabelecimento da conexão TCP

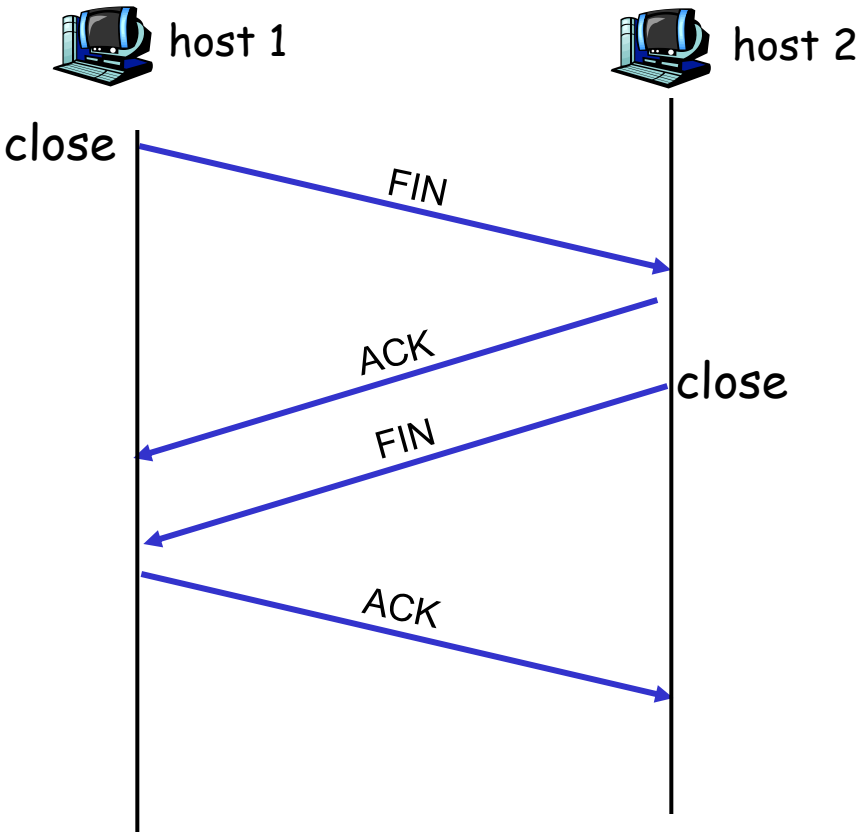
2. Gerenciamento da Conexão TCP

- Término da conexão TCP:
  - Apesar de as conexões TCP serem *full-duplex*, fica mais fácil compreender como as conexões são encerradas se as considerarmos um par de conexões simplex
  - Cada conexão simplex é encerrada de modo independente de sua parceira

2. Gerenciamento da Conexão TCP

- Término da conexão TCP:
  - Passo 1: host 1 envia o segmento TCP FIN ao host 2
  - Passo 2: host 2 recebe o FIN, responde com ACK, fecha a conexão no sentido host 1 – host 2 (dados podem continuar a fluir indefinidamente no sentido host 2 – host 1)
  - Passo 3: host 1 recebe o ACK e fecha a conexão no sentido host 1 – host 2
  - Passo 4: host 2 envia o segmento TCP FIN ao host 1
  - Passo 5: host 1 recebe o FIN, responde com ACK e fecha a conexão no sentido host 2 – host 1
  - Passo 6: host 2 recebe ACK e fecha a conexão no sentido host 2 – host 1

2. Gerenciamento da Conexão TCP



Término da conexão TCP

Exercício

14. Um processo no host 1 foi atribuído à porta *p*, e um processo no host 2 foi atribuído à porta *q*. É possível haver duas ou mais conexões TCP entre essas duas portas ao mesmo tempo?

15. A fragmentação e a remontagem de pacotes são tratadas pelo IP e são invisíveis para o TCP. Isso quer dizer que o TCP não tem de se preocupar com a chegada de dados na ordem errada?

3. Controle de Fluxo

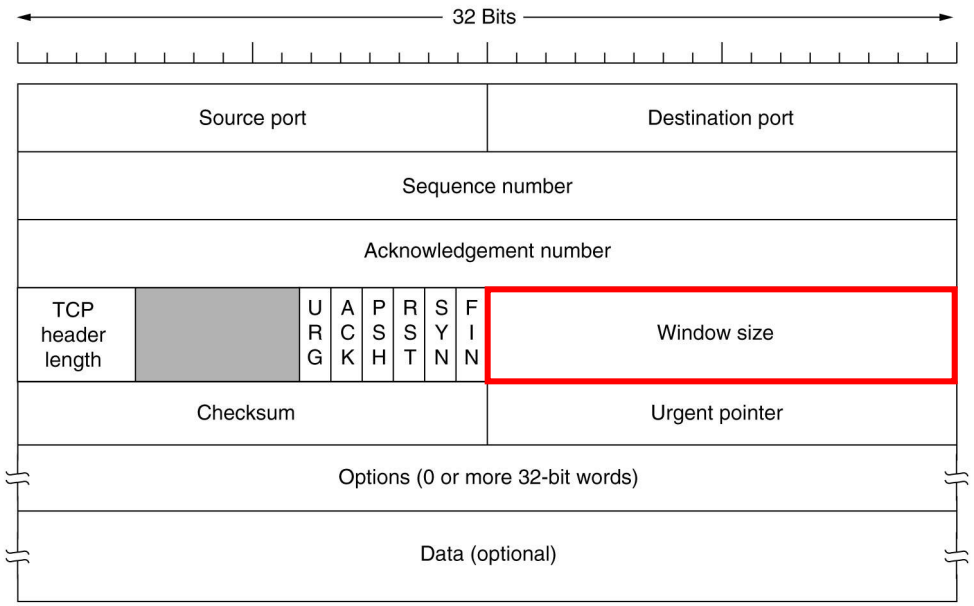
- Evitar que o transmissor esgote a capacidade do receptor
- No momento do estabelecimento da conexão, um *buffer* de recepção é alocado e seu tamanho é informado para a entidade par no campo *Window size* do segmento TCP
- Em toda confirmação é enviado o espaço disponível nesse *buffer*
  - Esse espaço é chamado de janela

3. Controle de Fluxo

- Transmissor não deve esgotar os *buffers* de recepção enviando dados rápido demais
  - Receptor: explicitamente informa ao transmissor a área livre no buffer no campo *Windows size* no segmento TCP
  - Transmissor: mantém a quantidade de dados transmitidos mas não reconhecidos menor que o último *Windows size* recebido

3. Controle de Fluxo

- *Window size:*
  - Indica quantos bytes podem ser enviados a partir do byte confirmado



4. Controle de Congestionamento

- Quando a carga oferecida a qualquer rede é maior que sua capacidade, acontece um congestionamento
- Diferente de controle de fluxo!
- Sintomas:
  - Perda de pacotes (saturação de buffer nos roteadores)
  - Atrasos grandes (filas nos buffers dos roteadores)
- Um dos 10 problemas mais importantes na Internet!

4. Controle de Congestionamento

- Congestionamento da rede pode ser piorado se a camada de transporte retransmite pacotes que não foram perdidos
  - Esse problema pode causar até um colapso da rede

4. Controle de Congestionamento

- Como detectar o congestionamento?
  - TCP usa a quantidade de pacotes perdidos (pacotes com *timeout*) como uma medida de congestionamento
    - Reduz a taxa de retransmissão a medida que esse valor aumenta

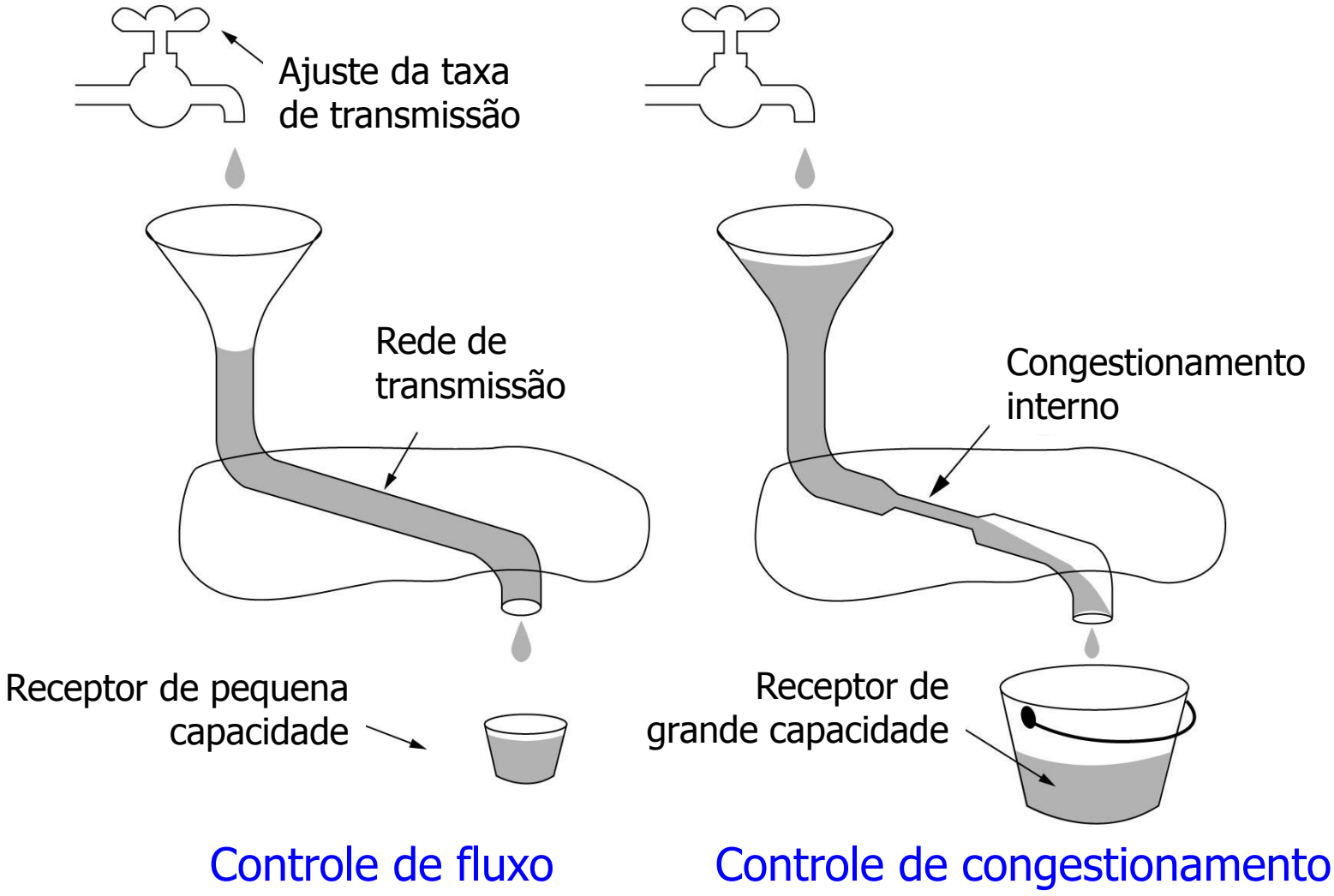
4. Controle de Congestionamento

- Antigamente, um *timeout* causado por um pacote perdido podia ter sido provocado por
  - Ruído em uma linha de transmissão, ou
  - O pacote foi descartado em um roteador congestionado
- Hoje em dia, a perda de pacotes devido a erros de transmissão é relativamente rara
- A maioria dos *timeouts* de transmissão na Internet se deve a congestionamentos

4. Controle de Congestionamento

- Na Internet, existem dois problemas potenciais:
  - A capacidade da rede $\Rightarrow$ Controle de Congestionamento
  - A capacidade do receptor $\Rightarrow$ Controle de Fluxo

4. Controle de Congestionamento



4. Controle de Congestionamento

- Cada transmissor mantém duas janelas:
  - A janela fornecida pelo receptor (controle de fluxo)
  - A janela de congestionamento (controle de congestionamento)
- O número de bytes que podem ser transmitidos é o valor mínimo entre as duas janelas

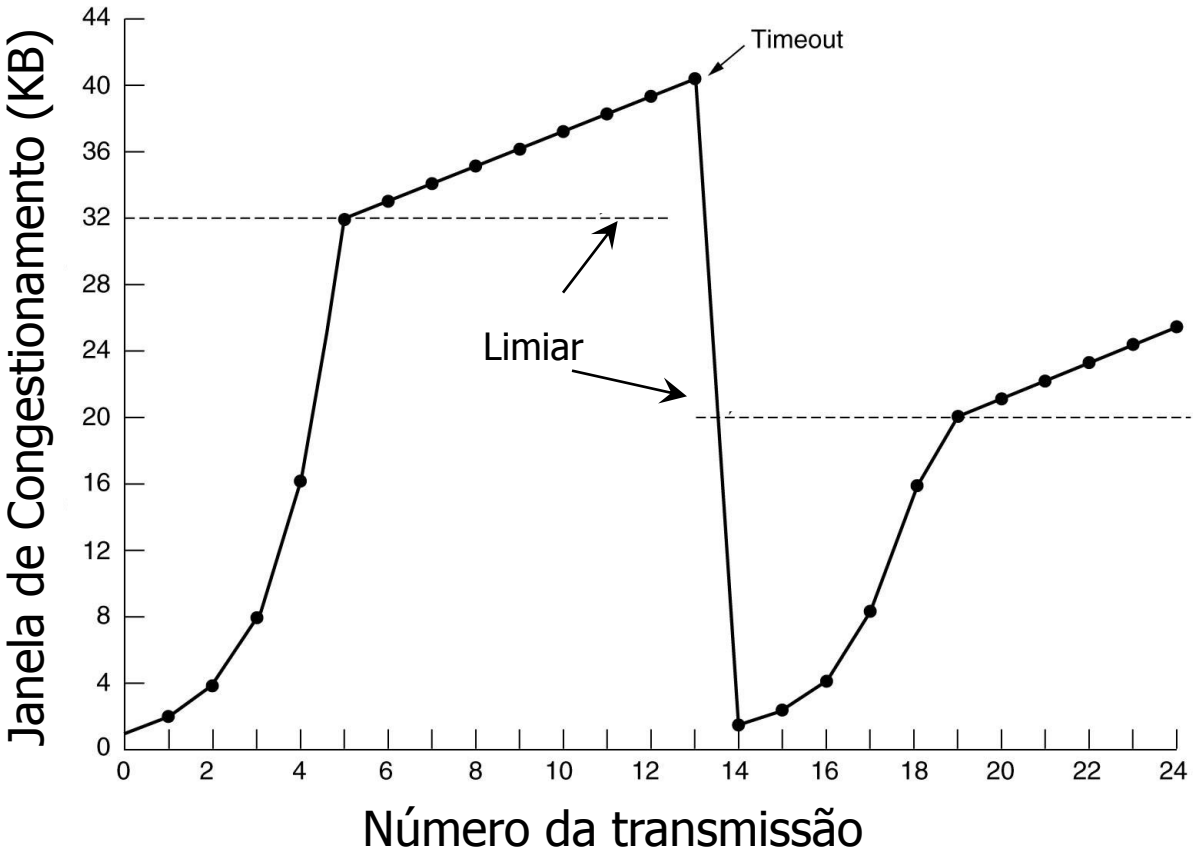
4. Controle de Congestionamento

- Algoritmo de inicialização lenta:
  - Quando uma conexão é estabelecida, a janela de congestionamento é igual ao tamanho do segmento máximo em uso na conexão
  - Cada rajada confirmada duplica a janela de congestionamento
- A janela de congestionamento mantém seu crescimento exponencial até que:
  - Um limiar seja atingido,
  - Aconteça um *timeout,* ou
  - A janela do receptor seja alcançada

4. Controle de Congestionamento

- Um limiar é atingido:
  - As transmissões bem-sucedidas proporcionam um crescimento linear à janela de congestionamento
- Acontece um *timeout*:
  - O limiar é definido como a metade da janela de congestionamento atual
  - Janela de congestionamento ← segmento máximo
  - Inicialização lenta é usada
- A janela de recepção é alcançada:
  - Janela de congestionamento pára de crescer e permanece constante

4. Controle de Congestionamento



A transmissão de mensagens é feita de forma exponencial até atingir um dado valor, quando passa a aumentar mais lentamente

Exercício

16. Considere o efeito de usar o início lento em uma linha com um tempo de percurso de ida e volta de 10 ms e sem congestionamento. A janela de recepção tem 24 KB e o tamanho máximo do segmento é 2 KB. Quanto tempo é necessário para que a primeira janela completa possa ser enviada?

5. Gerenciamento de *Timers*

- O TCP utiliza um *timer* de retransmissão:
  - Quando um segmento é enviado, um *timer* é ativado
  - Se o segmento for confirmado antes do *timer* expirar, ele será interrompido
  - Se o *timer* expirar antes da confirmação chegar, o segmento será retransmitido e o *timer* será disparado novamente

5. Gerenciamento de *Timers*

- Tempo de viagem de ida e volta (RTT): tempo transcorrido desde o instante em que um segmento é enviado até o instante em que ele é reconhecido
- Para cada conexão, o TCP mantém uma variável, RTT, que é a melhor estimativa no momento para o tempo de percurso de ida e volta até o destino em questão

5. Gerenciamento de *Timers*

- Como escolher o valor do *timer* do TCP?
  - Maior que o RTT (RTT varia)
  - Muito curto:
    - Retransmissões desnecessárias, sobrecarregando a Internet com pacotes inúteis
  - Muito longo:
    - O desempenho será prejudicado devido ao longo retardo de retransmissão sempre que um pacote se perder

5. Gerenciamento de *Timers*

- O TCP atualiza a variável RTT de acordo com a fórmula:

$$RTT = \alpha RTT + (1- \alpha)M$$

onde:

  - M último RTT medido
  - α é um fator de suavização que determina o peso dado ao antigo valor

5. Gerenciamento de *Timers*

- O TCP estima o *timer* de retransmissão como sendo β RTT, onde β é proporcional ao desvio-padrão da função de densidade de probabilidades de tempos de chegada de confirmações
  - Grande variância $\Rightarrow$ alto valor de β
  - Pequena variância $\Rightarrow$ pequeno valor de β