

Camada de Transporte

Redes de Computadores

Profa. Kalinka Castelo Branco

Universidade de São Paulo

Maio de 2020

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

1 Introdução

Protocolos e Serviços

Multiplexação

Endereçamento

Sockets

2 UDP

3 TCP

Recuperação de erros

Estabelecimento de conexões

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de erros
Estabelecimento de conexões



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

Função

A função básica da Camada de Transporte é aceitar dados da camada de aplicação (camadas superiores), dividi-los em unidades menores em caso de necessidade, passá-los para a Camada de Rede e garantir que todas essas unidades cheguem corretamente à outra extremidade.

Camada de Transporte

Profa.
Kalinka
Branco

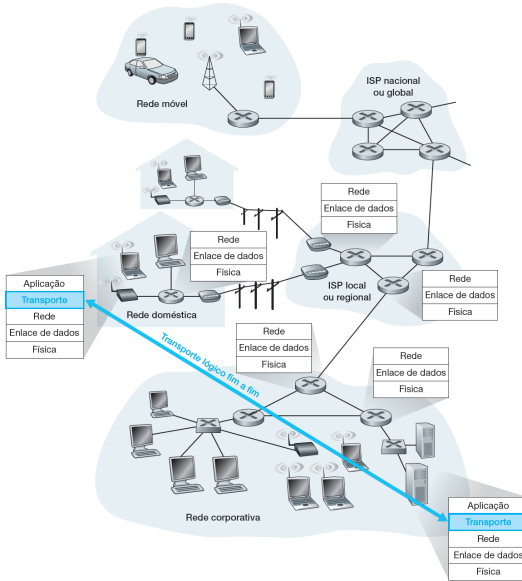
Introdução

Protocolos e Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de erros
Estabelecimento de conexões



- A Camada de Transporte fornece comunicação lógica entre processos de aplicações.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- A Camada de Transporte é uma camada fim a fim, que liga a origem ao destino;
- Um programa da máquina de origem mantém uma conversa com um programa semelhante instalado na máquina de destino, utilizando cabeçalhos de mensagem e mensagens de controle;
 - As PDUs são chamadas de **segmentos**. Um segmento é composto pelo cabeçalho da Camada de Transporte e os dados da Camada de Aplicação.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Fornecem comunicação lógica entre processos de aplicação em diferentes *hosts*;
- Os protocolos de transporte são executados nos sistemas finais da rede;
- Serviço de Transporte vs Serviços de Rede:
 - **Camada de Rede:** transferência de dados entre computadores (*end systems*);
 - **Camada de Transporte:** transferência de dados entre processos.
- Utiliza e aprimora os serviços oferecidos pela Camada de Rede.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Serviços de Transporte da Internet:
 - Não confiável (“best-effort”): UDP
 - Confiável: TCP
 - Controle de congestionamento;
 - Controle de fluxo;
 - Orientado à conexão.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento

Sockets

UDP

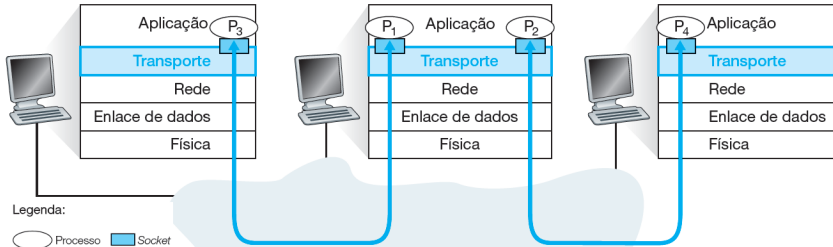
TCP

Recuperação de
erros

Estabelecimento
de conexões

- O hardware e/ou software da Camada de Transporte que executa o trabalho é chamado de **entidade de transporte** (*transport entity*);
- Esta entidade de transporte pode estar:
 - No *kernel* do sistema operacional;
 - Em um outro processo de usuário;
 - Em uma biblioteca vinculada a aplicações de rede;
 - Ou em uma placa de interface de rede.

- **Demultiplexação:** entrega de segmentos recebidos aos processos de aplicação corretos;
- **Multiplexação:** reunir dados de múltiplos processos de aplicação ou de diferentes conexões de um processo, juntar cabeçalhos com dados para cada segmento e repassar para a Camada de Rede.



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

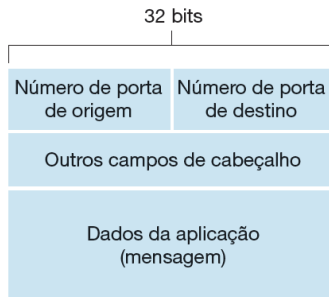
UDP

TCP

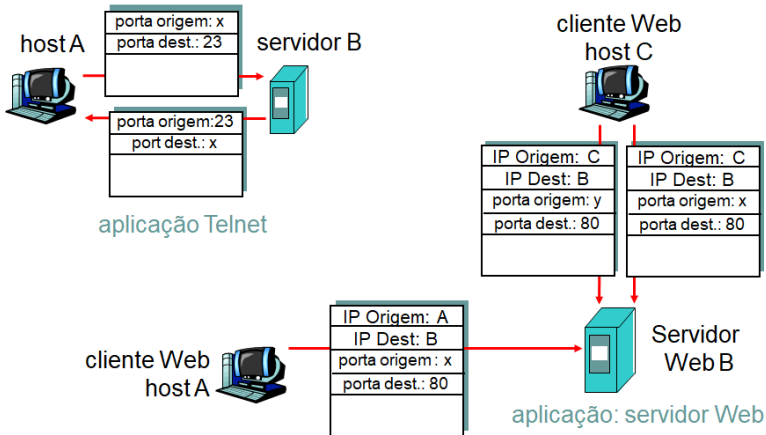
Recuperação de
erros

Estabelecimento
de conexões

- **Multiplexação/demultiplexação:** baseada em número de porta do transmissor, número de porta do receptor e endereços IP:
 - Números de porta origem e destino em cada segmento;
 - Lembre: portas com números bem conhecidos são usadas para aplicações específicas.



Exemplo:



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Muitos *hosts* são multiprogramados;
 - Isso significa que muitas conexões estarão entrando e saindo de cada *host* e processos executam de forma concorrente.
- É responsabilidade da Camada de Transporte multiplexar todas as comunicações para a Camada de Rede e posteriormente determinar a qual processo uma mensagem pertence;

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento

Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Da mesma forma que em outras camadas, a camada de transporte também possui um endereçamento;
- Quando um processo de aplicação deseja estabelecer uma conexão com um processo de aplicação remoto, é necessário especificar a aplicação com a qual ele irá se conectar.
 - O método utilizado é definir os endereços de transporte que os processos podem ouvir para receber solicitações de conexão.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Os processos utilizam os TSAP (*Transport Service Access Point* – Ponto de Acesso de Serviços de Transporte) para se intercomunicarem;
- Em redes IP, o TSAP é um número de 16 bits chamado de porta. O endereço da Camada de Transporte é um número de 48 bits, composto pela agregação do endereço IP do *host* e o número da porta;
- Os serviços são obtidos por meio da comunicação entre os *sockets* do transmissor e do receptor.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

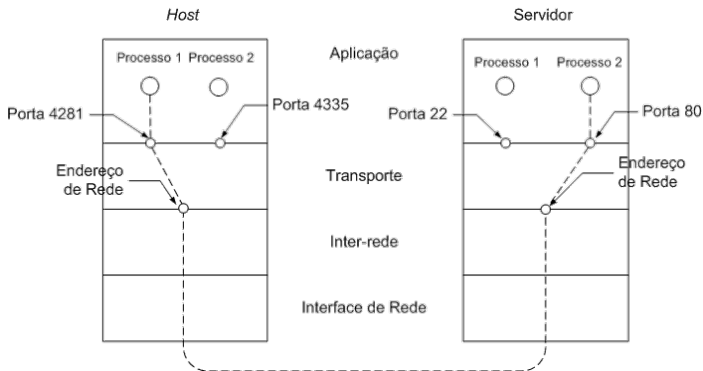
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- A numeração de portas utiliza 16 bits. Logo, existem $2^{16} = 65536$ portas a serem utilizadas no total;
- Isso para cada protocolo da camada de transporte. Portanto, 65536 portas para o TCP e 65536 portas para o UDP;
- Conhecer essas portas é fundamental para operar um *Firewall* de forma satisfatória.

Firewall

Dispositivo de uma rede de computadores que tem por objetivo aplicar políticas de segurança a um determinado ponto da rede. Pode ser do tipo filtro de pacotes, *proxy* de aplicações, etc.

- **Com tantas portas, como saber todas elas?**
- Não precisa conhecer todas, uma vez que a maior parte delas não é especificada;
- Apenas as primeiras 1024 são especificadas;
- Para um administrador de rede é imprescindível saber pelo menos as portas dos serviços básicos de Rede: Telnet, SSH, FTP, SMTP, POP, HTTP, HTTPS...;
- O uso das portas de 1 a 1024 é padronizada pela IANA (*Internet Assigned Numbers Authority*);
 - Essa entidade é responsável por alocar portas para determinados serviços;
 - Essas portas são chamadas de *well-known ports*.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento

Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

Serviço	Porta	Protocolo
daytime	13	TCP e UDP
ftp-data	20	TCP
ftp	21	TCP
ssh	22	TCP
telnet	23	TCP
smtp	25	TCP e UDP
name	42	TCP e UDP
nameserver	42	TCP e UDP
tftp	69	TCP e UDP
www	80	TCP
pop3	110	TCP e UDP
netbios-ns	137	TCP e UDP
netbios-dgm	138	TCP e UDP
netbios-ssn	139	TCP e UDP

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento

Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Os *sockets* são diferentes para cada protocolo de transporte. Desta forma, mesmo que um *socket* TCP possua o mesmo número que um *socket* UDP, ambos são responsáveis por aplicações diferentes;
- Os *sockets* de origem e destino são responsáveis pela identificação única da comunicação. Desta forma é possível a implementação da função conhecida como multiplexação;
- A multiplexação possibilita que haja várias conexões partindo de um único *host* ou terminando em um mesmo servidor.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- A formação do *socket* se dá da seguinte forma:
 - Ao iniciar uma comunicação é especificado para a aplicação o endereço IP de destino e a porta de destino;
 - A porta de origem é atribuída dinamicamente pela Camada de Transporte. Ele geralmente é um número aleatório acima de 1024;
 - O endereço IP de origem é atribuído pela Camada de Rede.

- `int sockfd = socket(domain, type, protocol):` criação do *socket*.
 - `sockfd` é o descritor do *socket*;
 - `domain` é o domínio de comunicação (`AF_INET` para IPv4 e `AF_INET6` para IPv6);
 - `type` é o tipo da comunicação (`SOCK_STREAM` para TCP e `SOCK_DGRAM` para UDP);
 - `protocol` é o protocolo usado na comunicação, dependente da implementação (`IPPROTO_TCP` para TCP e `IPPROTO_UDP` para UDP).
- `int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen):` anexa um endereço local ao *socket*.
 - `*addr` é o endereço do servidor;
 - `addrlen` é o tamanho do endereço do servidor;

- `int listen(int sockfd, int queue_size)`: anuncia o aceite de conexões.
 - `queue_size` é o número máximo de conexões;
- `int new_socket = accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen)`: bloqueia até uma tentativa de conexão ser recebida.
- `int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen)`: conexão feita por um cliente a um servidor.
- `int close(int sockfd)`: encerra a conexão.

- `ssize_t send(int sockfd, const void *buf, size_t len, int flags)`: envia dados através do *socket*.
 - `*buf` são os dados;
 - `len` é o tamanho dos dados;
 - `flags` usado para configuração do envio;
 - Sem `flags` é equivalente ao `write` e é usado para envio de dados no protocolo TCP.
- `ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *dest_addr, socklen_t addrlen)`: envia dados através do *socket*, geralmente usado com UDP.
 - `*dest_addr` é o endereço de destino;
 - `addrlen` é o tamanho do endereço de destino;

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento

Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- `ssize_t recv(int sockfd, void *buf, size_t len, int flags)`: recebe dados através do *socket*.
 - Sem `flags` é equivalente ao `read` e é usado para envio de dados no protocolo TCP.
- `ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen)`: recebe dados através do *socket*, geralmente usado com UDP.
 - `*src_addr` é o endereço de origem;
 - `*addrlen` é o endereço de uma variável que guarda o tamanho do endereço de origem;

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- Definido pela RFC 768;
- Protocolo de transporte da Internet “sem gorduras”;
- Serviço “*best effort*”;
- Segmentos UDP são conhecidos como Datagramas;
- Datagramas podem ser perdidos e entregues fora de ordem para a aplicação;
- Sem conexão:
 - Não há confirmação entre o UDP transmissor e o receptor;
 - Cada segmento UDP é tratado de forma independente dos outros.

- **Por quê existe um protocolo como o UDP?**
 - Não há estabelecimento de conexão (que pode gerar em atrasos);
 - Simples: não há estado de conexão nem no transmissor, nem no receptor;
 - Cabeçalho de segmento reduzido;
 - Não há controle de congestionamento: UDP pode enviar segmentos tão rápido quanto desejado (e possível).

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

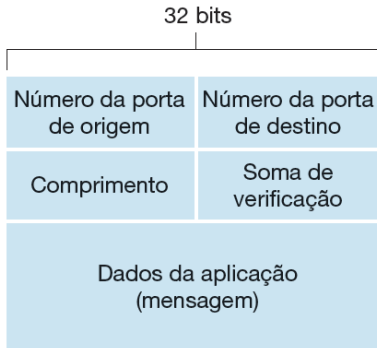
UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- Muito usado por aplicações de multimídia contínua (voz e vídeo);
 - Tolerantes à perda;
 - Sensíveis à taxa.
- Outros usos do UDP:
 - DNS (Sistema de Nomes de Domínios);
 - SNMP (Protocolo Simples de Gerenciamento de Rede);
 - RIP (*Routing Information Protocol*).
- Transferência confiável sobre UDP:
 - Acrescentar confiabilidade na Camada de Aplicação;
 - Recuperação de erro específica de cada aplicação.

- **Porta de Origem e Porta de Destino:** indicam os pares de porta que estão executando a comunicação;
- **Comprimento:** indica o comprimento de todo o datagrama isto é, cabeçalho e dados;
- **Soma de verificação:** verificação de integridade do datagrama.



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- O TCP foi formalmente definido na RFC 793. Posteriormente, alguns erros foram corrigidos e o TCP foi definido na RFC 1122;
- O TCP (Protocolo de Controle de Transmissão) foi projetado para oferecer um fluxo de bytes fim a fim confiável em uma rede não confiável;
- É um protocolo orientado a conexão que permite a entrega sem erros de um fluxo de bytes para qualquer *host*;
- O TCP foi projetado para se adaptar dinamicamente às propriedades da Camada de Rede e ser robusto diante dos muitos tipos de falhas que podem ocorrer.

TCP (*Transport Control Protocol*)

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e Serviços

Multiplexação

Endereçamento Sockets

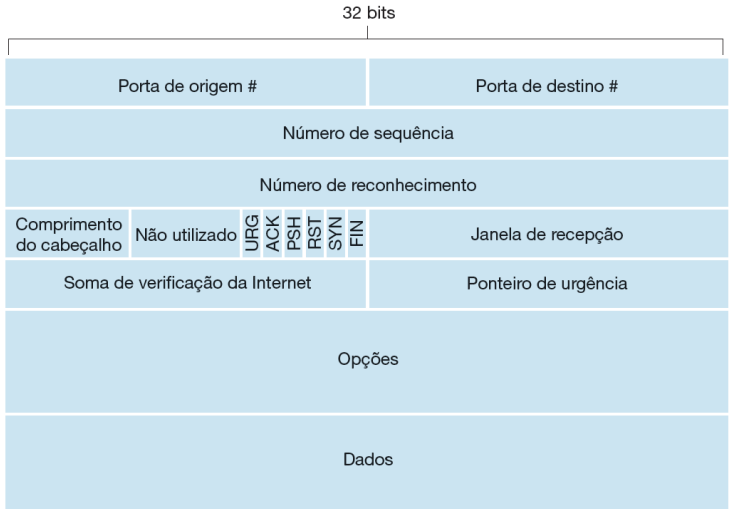
UDP

TCP

Recuperação de erros

Estabelecimento de conexões

- Segmento TCP:



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

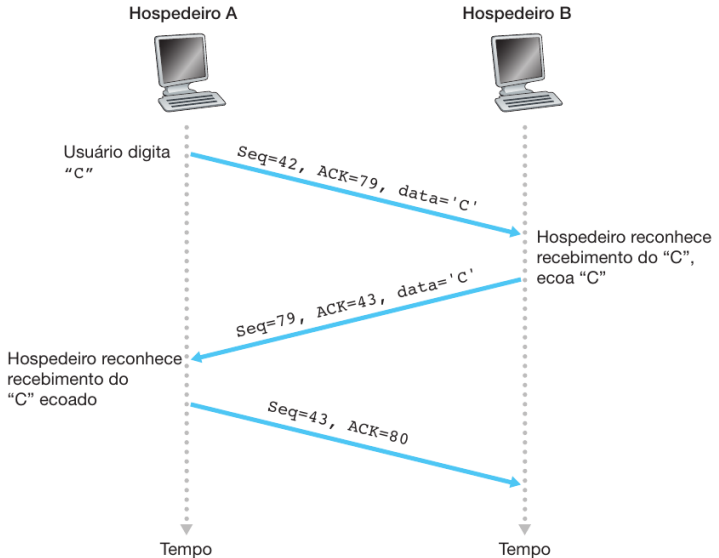
Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- O TCP proporciona uma transferência confiável de dados, o que também é chamado de confiabilidade ou recuperação de erros;
- Para conseguir a confiabilidade, o TCP enumera os bytes de dados usando os campos referentes à sequência e aos ACKs no cabeçalho TCP;
- O TCP alcança a confiabilidade em ambas as direções, usando um campo referente ao número de sequência de uma direção, combinado com o campo referente ao ACK na direção oposta.



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

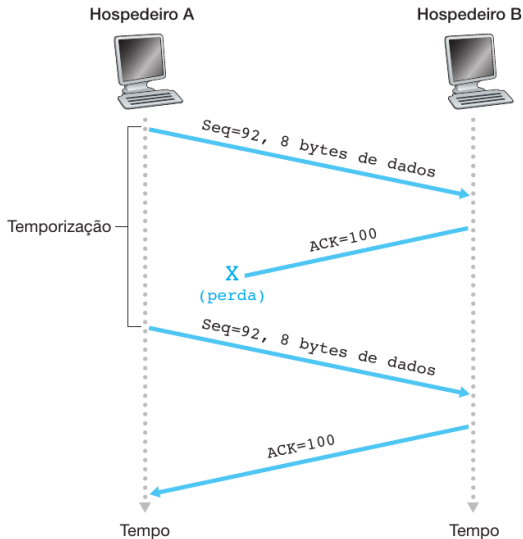
Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- O campo ACK indica o próximo byte a ser recebido. O número de sequência indica o número do primeiro byte do segmento correspondente à sua posição no fluxo de dados.
- Caso a temporização se esgote, os segmentos são reenviados, garantindo que sempre sejam entregues.



Camada de Transporte

Profa.
Kalinka
Branco

Introdução

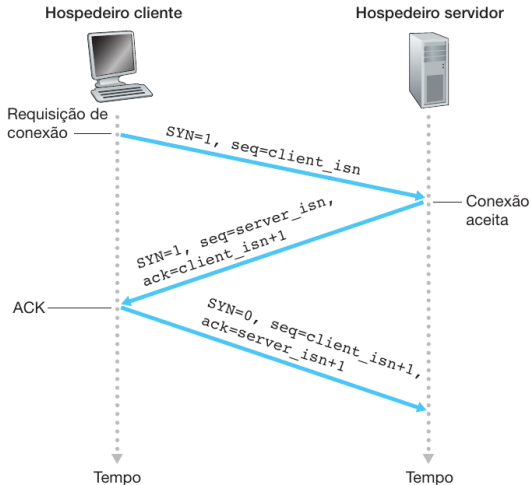
Protocolos e Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de erros
Estabelecimento de conexões

- O estabelecimento de uma conexão TCP ocorre antes que qualquer outro recurso TCP possa começar seu trabalho;
- O estabelecimento da conexão se fundamenta no processo de inicialização dos campos referentes à sequência, aos ACKs e na troca dos números de *sockets* usados;
- As conexões são estabelecidas no TCP por meio do *three-way handshake* (acordo de três vias).



- **SYN:** indica pedido de conexão e confirmação da conexão;
- **seq:** número sequencial com início diferente para cada *host*;
- **ACK:** segmento de confirmação (*acknowledgment*).

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- O estabelecimento da conexão é feito usando dois bits no cabeçalho TCP: SYN e ACK;
- Um segmento que possua a flag SYN ativa, sinaliza uma requisição de sincronia do número de sequência (seq). Essa sincronização é necessária em ambos os sentidos, pois origem e destino utilizam números de sequência distintos;
- Cada pedido de conexão é seguido de uma confirmação utilizando o bit ACK;
- O segundo segmento do *three way handshake* exerce as duas funções ao mesmo tempo: confirma a sincronização do servidor com o cliente e requisita a sincronização do cliente com o servidor.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

- Basicamente o *three way handshake* “simula” um acordo:
 - ① O cliente pergunta ao servidor: “Você está aí?”;
 - ② O servidor responde: “Sim estou...”;
 - ③ Depois o servidor pergunta: “Você está aí?”;
 - ④ O cliente responde: “Sim estou...”
- Mas espere! Como temos 4 sentenças em apenas 3 trocas de mensagens? Simples, a segunda mensagem contém uma resposta e uma pergunta.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- 1 **Cliente:** Servidor, você está aí? (SYN)
 - 2 **Servidor:** Sim, estou... (ACK). E você, está aí? (SYN);
 - 3 **Cliente:** Sim, estou... (ACK).
- O servidor precisa perguntar se o cliente está lá pela simples necessidade de sincronização do número de sequência. O número de sequência é utilizado para garantir a entrega de todas as mensagens.

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de erros
Estabelecimento de conexões

- ① **Cliente:** Câmbio servidor, mensagem 2000 (Número de sequência do cliente). O senhor está disponível (SYN)?
- ② **Servidor:** Positivo, cliente! Mensagem 1450 (Número de sequência do servidor). Prossiga com a mensagem 2001 (ACK=2001). Câmbio!
- ③ **Cliente:** Positivo, servidor! Mensagem 2001 (Número de sequência). Confirmando número da próxima mensagem: 1451. Câmbio!

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

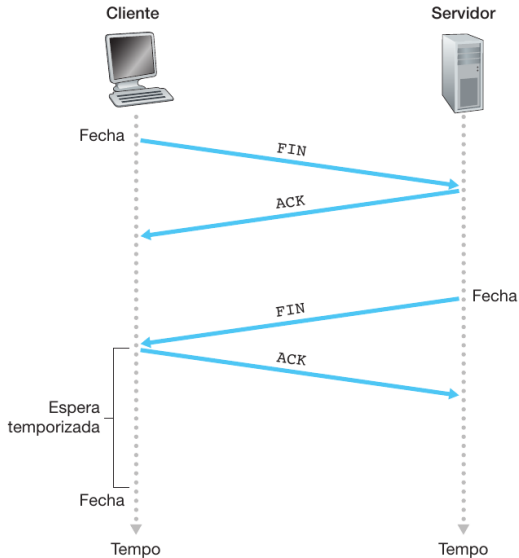
Protocolos e
Serviços
Multiplexação
Endereçamento
Sockets

UDP

TCP

Recuperação de
erros
Estabelecimento
de conexões

- Dessa forma eles trocam o número de sequência, que tem como função “enumerar” as mensagens de cada um;
- Por exemplo, se a última mensagem foi a 2001 e a mensagem que chegou pro servidor foi a 2003, ele tem completa certeza que uma mensagem (2002) se perdeu no caminho! Então, basta solicitar uma retransmissão;
- O número de sequência nem sempre é incrementado por 1, ele pode ser incrementado com base no número de bytes enviados pela origem;
- O ACK tem como objetivo solicitar a continuidade das mensagens. Pode-se interpretar um ACK=210 como sendo: “Pronto, recebi até a 209, pode mandar a 210.”



- **FIN**: indica pedido para fechar a conexão (emitido pelos dois *hosts*);
- **ACK**: segmento de confirmação (*acknowledgment*).

Camada de Transporte

Profa.
Kalinka
Branco

Introdução

Protocolos e
Serviços

Multiplexação

Endereçamento
Sockets

UDP

TCP

Recuperação de
erros

Estabelecimento
de conexões

Exercício

Os hospedeiros A e B estão se comunicando por meio de uma conexão TCP, e o hospedeiro B já recebeu de A todos os bytes até o byte 126. Suponha que A envie, então, dois segmentos para B sucessivamente. O primeiro e o segundo segmentos contêm 80 e 40 bytes de dados. No primeiro segmento, o número de sequência é 127, o número de porta de partida é 302, e o número de porta de destino é 80. O hospedeiro B envia um reconhecimento ao receber um segmento do hospedeiro A:

- (a) No segundo segmento enviado do hospedeiro A para B, quais são o número de sequência, da porta de origem e da porta de destino?
- (b) Se o primeiro segmento chegar antes do segundo, no reconhecimento do primeiro segmento que chegar, qual é o número do reconhecimento, da porta de origem e da porta de destino?