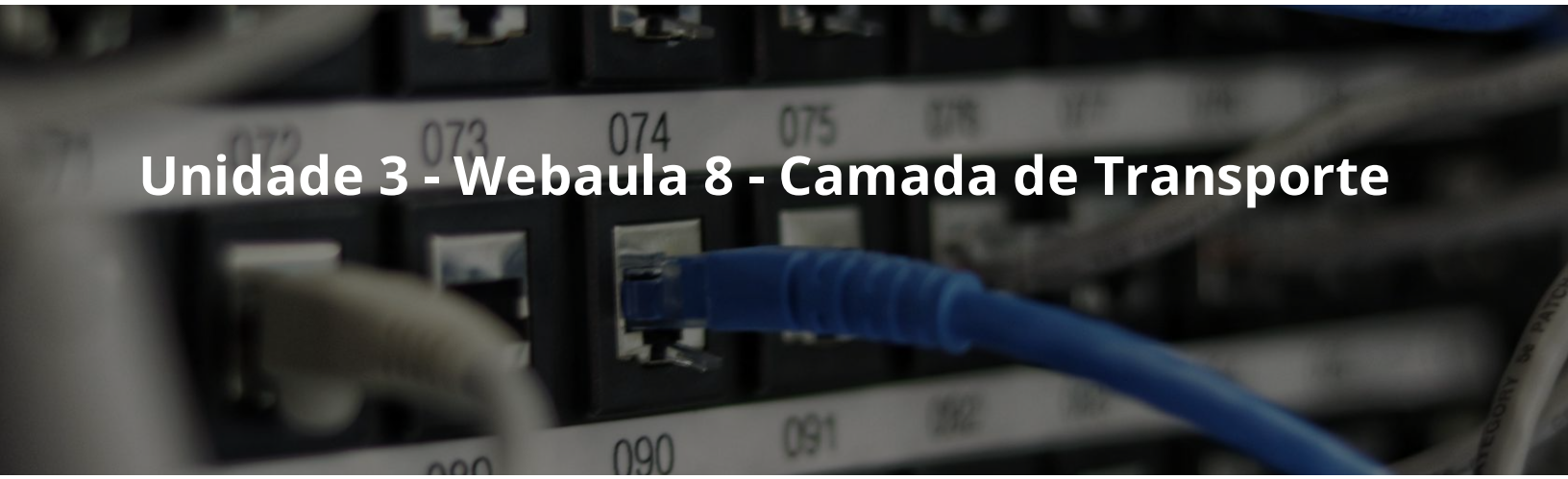


Unidade 3 - Webaula 8 - Camada de Transporte

Olá! Tudo bem?

Seja bem-vindo(a) à Webaula 8 de **Redes de Computadores.**

INTRODUÇÃO

- Introdução à unidade 3 - Arquitetura Internet - Alto Nível
- Introdução à Webaula 8

TÓPICO 1

- Multiplexação e Demultiplexação

TÓPICO 2

- Protocolo de Transporte TCP
- Atividade de Passagem

TÓPICO 3

- Protocolo de Transporte UDP
- Atividade de Passagem

TÓPICO 4

Segurança de Camada de Transporte

Atividade de Passagem

RESUMO

Resumo da Webaula 8

REFERÊNCIAS

Referências

Créditos

Introdução à unidade 3 - Arquitetura Internet - Alto Nível

Olá, estudante!

Na segunda Unidade foram apresentadas as principais características das três primeiras camadas do modelo TCP/IP, das camadas de mais baixo nível do modelo. Nesta terceira Unidade apresentarei as características e como funcionam as duas camadas de mais alto nível do modelo.

Os próximos três tópicos apresentam cada uma das últimas duas camadas do modelo TCP/IP.

- **Camada de Transporte**: permite que as entidades pares da origem e do destino mantenham um canal de comunicação.
- **Camada de Aplicação – básica**: contém os protocolos de suporte às aplicações padrão básicas de alto nível.
- **Camada de Aplicação – suplementar**: contém os protocolos de suporte às novas aplicações de alto nível.

Em cada uma destas duas camadas, será dado um enfoque primeiramente nos principais conceitos envolvidos na sua implementação e nos serviços necessários à sua integração ao modelo e, por fim, em como estes serviços são implementados na arquitetura TCP/IP.

Espera-se que ao final desta Unidade você seja capaz de entender as nuances envolvidas na transmissão de dados entre os equipamentos da origem e do destino e de compreender como é que as principais aplicações de rede funcionam na internet e como que elas trocam informações entre si.

E então, vamos começar?

CONTINUE

Introdução à Webaula 8

Olá!

Vimos que a principal função da camada de rede é encontrar o melhor caminho desde uma origem até um destino: roteamento. Agora veremos que a camada de transporte é complementar à camada de rede pois sua função é promover uma transferência de dados confiável e econômica entre um processo de um computador de origem e um processo de um computador de destino, fim a fim, transmitindo segmentos de dados independente das redes físicas utilizadas (KUROSE; ROSS, 2013).

A responsabilidade fundamental dos protocolos da camada de transporte é ampliar o serviço de entrega dos protocolos da camada de rede entre equipamentos de rede para um serviço de entrega entre processos que rodam nestes equipamentos.

As funcionalidades que os protocolos da camada de transporte podem executar ao enviar e ao receber segmentos entre equipamentos de rede são:

- **multiplexação e demultiplexação**: reunião de dados provenientes de diferentes processos no computador de origem, encapsulando-os com a informação de cabeçalho apropriada para identificação, e encaminhamento dos segmentos resultantes para a camada de rede, e vice-versa;
- **transferência confiável de dados**: garantia de que dados enviados do processo de origem ao processo de destino sejam recebidos de forma correta e ordenada (íntegra), o

que é obtido através da utilização de números sequenciais, confirmações de recebimento e temporizadores (*QoS – Quality of Service*);

- **segmentação**: divisão das mensagens em partes menores para transmissão, de acordo com a capacidade dos protocolos da camada de rede, e remontagem das mensagens no destino a partir destas partes;

- **controle de fluxo**: limita a taxa de transmissão à capacidade do receptor para prevenir que o equipamento de destino não seja inundado por um tráfego excessivo de dados;

- **controle de congestionamento**: adapta ao longo do tempo a taxa de transmissão de acordo com a capacidade da rede para prevenir que conexões da camada de transporte não sejam inundadas por um tráfego excessivo de dados nos enlaces constituintes da rede entre os equipamentos de origem e de destino.

O objetivo desta aula é que você conheça os principais protocolos da camada de transporte e os tipos de serviços que eles oferecem, orientado à conexão e sem conexão, para que você saiba escolher qual o mais apropriado a ser utilizado na implementação de um software de rede.

E então, vamos começar?

CONTINUE

Multiplexação e Demultiplexação

Você já pensou como é que um sistema operacional de um equipamento de destino, ao receber um pacote de dados da rede, sabe para qual aplicativo aquele pacote é dirigido? Pois esta é uma das funções da camada de transporte: os protocolos desta camada executam a multiplexação e a demultiplexação incluindo dois campos especiais no cabeçalho dos segmentos, um **número de porta** de origem e um número de **porta de destino**. Usados em conjunto, estes **números de porta** (também chamados de endereços de porta) identificam univocamente os processos dos equipamentos de rede envolvidos em uma comunicação fim a fim.

As portas, números de 16 bits no caso da internet, não são definidas por nenhum órgão internacional de normalização, mas as portas 0 até 1023 são convencionadas e reservadas para aplicações específicas (Tabela 1).

Tabela 1 - Exemplos de Portas de Aplicações Específicas

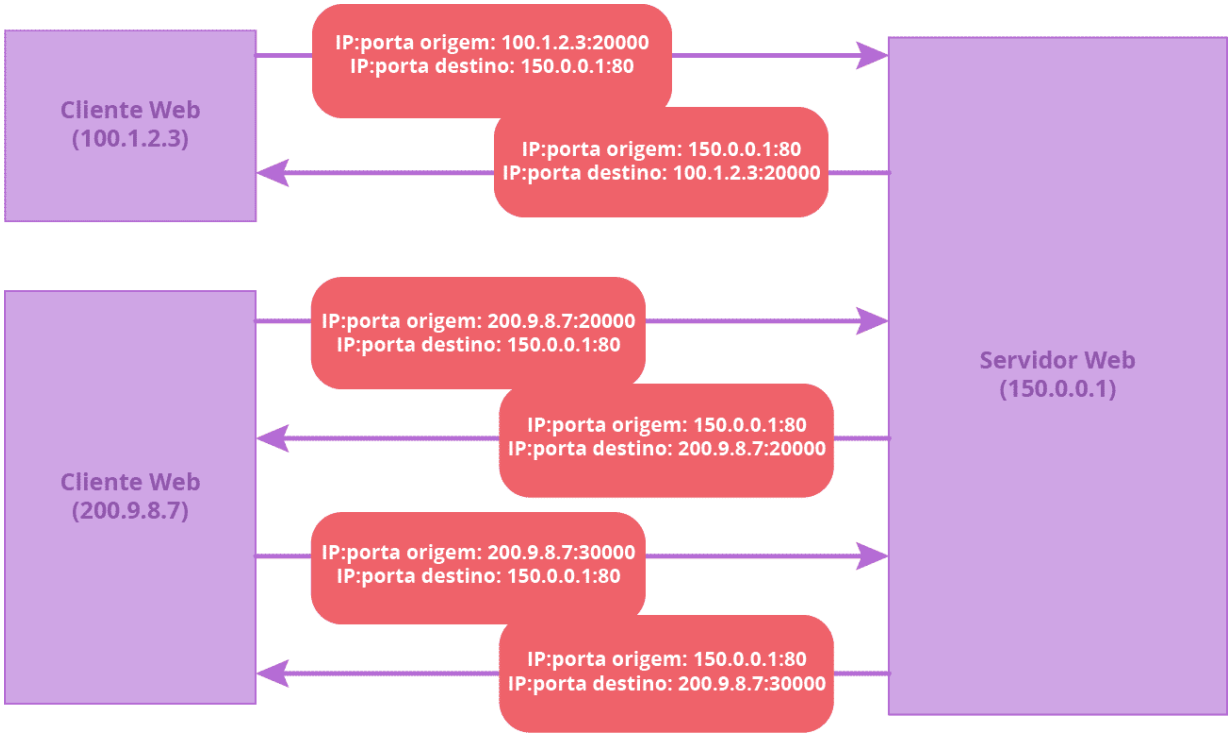
Porta	Serviço
21	FTP – protocolo de transferência de arquivos
23	Telnet – protocolo de emulação de terminal
25	SMTP – protocolo de encaminhamento de e-mails
53	DNS – serviço de resolução de nomes de domínio

Porta	Serviço
80	HTTP – protocolo de transferência de páginas web
110	POP3 – protocolo de transferência do conteúdo de e-mails
143	IMAP – protocolo de consulta do conteúdo de e-mails
161	SNMP – protocolo de gerenciamento de redes
443	HTTPS – protocolo seguro de transferência de páginas web
520	RIP – protocolo de publicação de tabelas de roteamento

Os segmentos de dados da camada de transporte referenciam ambas as portas, tanto de origem quanto de destino, porque um computador pode estar executando dois processos do mesmo tipo e no mesmo instante (conforme exemplo representado na Figura 1) e, portanto, somente a porta de destino de um processo não seria o suficiente para identificar univocamente os processos que estão se comunicando pela rede (MENDES, 2010).

Observe que em um equipamento cliente, o número da porta de destino é a porta da aplicação que está sendo invocada, enquanto o número da porta de origem é a porta do processo de origem, a qual não pode estar sendo utilizada por nenhum outro processo no equipamento (que é obtida automaticamente pelo protocolo da camada de transporte).

Figura 1 - Número de Porta para Multiplexação de Processos



Fonte: Elaboração própria (2022).
Arte/Diagramação: DME/FURB (2023).

CONTINUE

Protocolo de Transporte TCP

No conjunto de protocolos da internet há um protocolo da camada de transporte que fornece uma transferência confiável de dados chamado **TCP** (*Transmission Control Protocol*). Este protocolo implementa um serviço de transporte **orientado à conexão**.

As entidades TCP transmissoras e receptoras trocam dados na forma de segmentos. Um segmento TCP consiste em um cabeçalho fixo de 20 bytes (mais uma parte opcional raramente utilizada), seguido de zero ou mais bytes de dados (KUROSE; ROSS, 2013).

O protocolo TCP, definido nas RFC 793, RFC 1122, RFC 1323 e RFC 2581, decide qual deve ser o tamanho dos segmentos. Dois fatores restringem o tamanho do segmento: primeiro, cada segmento, incluindo o cabeçalho do TCP, deve caber na carga útil das camadas inferiores (os segmentos costumam ter 1460 bytes ou 512 bytes); segundo, cada enlace ao longo da rede tem uma **MTU** (*Maximum Transmission Unit*), e, portanto, cada segmento deve caber na menor MTU deste caminho.

Como o TCP é orientado à conexão, antes que um processo possa começar a enviar dados a outro, os dois processos precisam estabelecer a conexão alocando recursos em cada um deles e estabelecendo os parâmetros para a transferência dos dados. Como o protocolo TCP roda somente nos equipamentos transmissor e receptor, somente estes equipamentos finais mantêm o estado da conexão, e não todos os equipamentos intermediários da rede.

Perceba que uma conexão TCP provê um serviço bidirecional ponto-a-ponto, isto é, um fluxo simultâneo de dados nas duas direções entre um único remetente e um único destinatário.

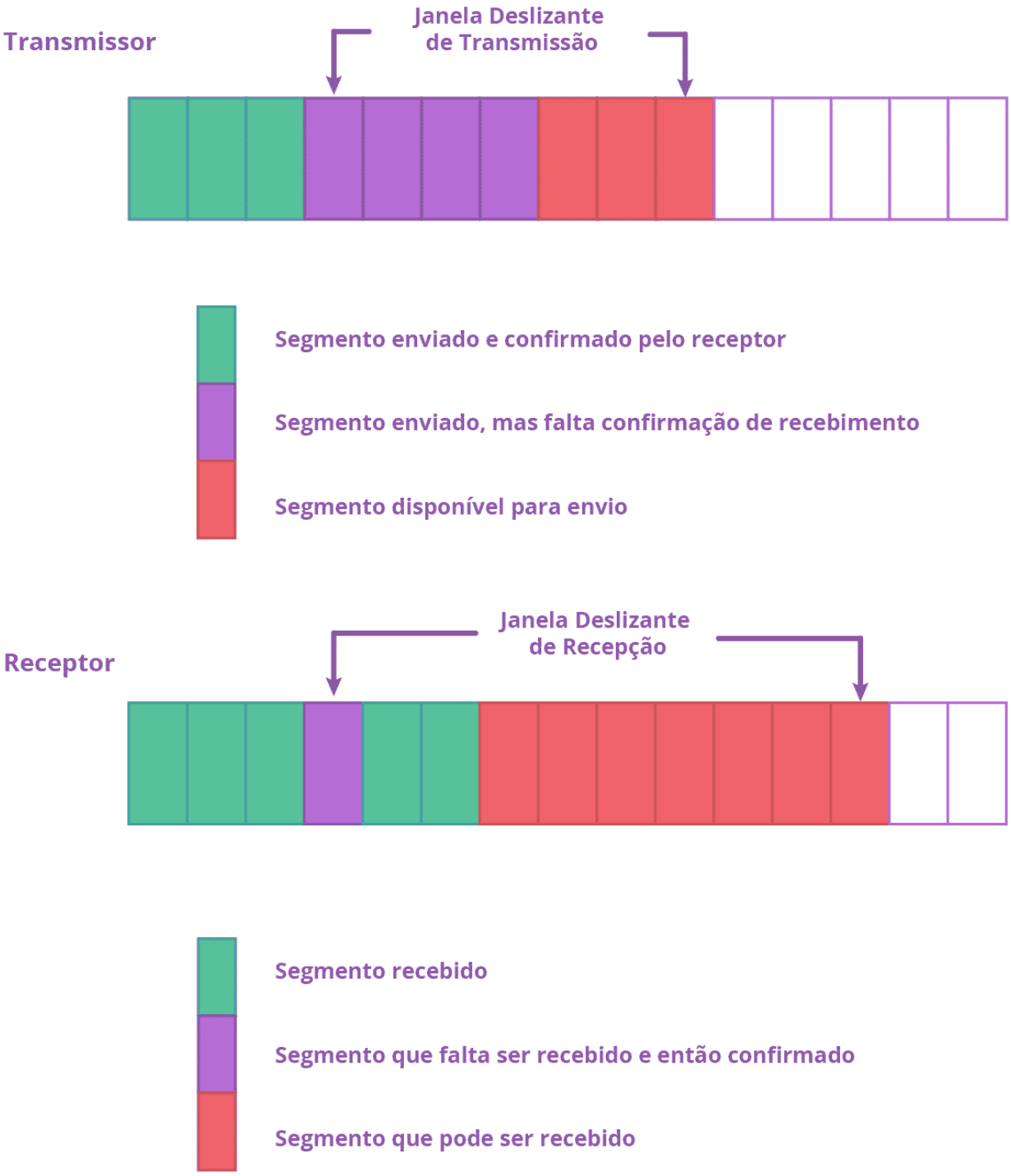
O mecanismo básico utilizado pelo TCP é a **janela deslizante**. Para cada segmento enviado de dentro da janela deslizante de transmissão, o transmissor dispara um temporizador. Quando o segmento

chega ao destino, a entidade TCP receptora retorna uma resposta com um número de confirmação igual ao próximo número de sequência que espera receber. Se o temporizador do transmissor expirar antes da confirmação ser recebida, o segmento é retransmitido. Você percebeu? Quando um segmento se perde na rede, o TCP retransmite o segmento em função do transmissor não ter recebido do receptor uma confirmação de recebimento e não em função de algum tipo de pedido de retransmissão do receptor.

Quando a carga de dados oferecida a qualquer rede é maior do que sua capacidade, acontece um congestionamento. Mesmo que a camada de rede também tente gerenciar o congestionamento, o trabalho mais pesado é feito pelo TCP.

O **controle de congestionamento** é um componente muito importante do protocolo TCP. Uma conexão TCP controla a taxa de transmissão limitando o número de segmentos transmitidos que ainda não foram confirmados, o que é feito através do controle do tamanho da janela de transmissão (Figura 2). Uma conexão TCP começa com um tamanho pequeno da janela e vai aumentando-o gradativamente até que ocorra uma perda de segmento, quando o valor da **janela de transmissão** é novamente reduzido e o processo é reiniciado.

Figura 2 - Controle de Congestionamento do TCP



Fonte: Elaboração própria (2022).
Arte/Diagramação: DME/FURB (2023).

Quase todas as implementações TCP atuais assumem que os temporizadores de retransmissão expiram devido a congestionamentos e não a pacotes perdidos (confiabilidade nos meios de

transmissão com fio). Consequentemente, quando um temporizador expira, o TCP diminui o ritmo (diminuindo a janela de transmissão) e começa a transmitir mais lentamente.

A ideia por trás desta abordagem é reduzir a carga de dados na rede e, portanto, diminuir o congestionamento. No entanto, os enlaces de dados das transmissões sem fio não são confiáveis, perdendo pacotes continuamente. A melhor estratégia para lidar com pacotes perdidos seria reenviá-los o mais rápido possível. Nesse caso específico, diminuir o ritmo de transmissão torna a situação ainda pior.

O protocolo TCP provê ainda um mecanismo denominado **controle de fluxo**, que elimina a possibilidade de a entidade TCP receptora não ter mais memória de recepção disponível. Este mecanismo é, portanto, um balanceamento de velocidades, tornando a taxa de transmissão no máximo igual à taxa possível de recepção. O controle de fluxo é obtido através de uma informação, mantida na entidade TCP transmissora, que corresponde ao tamanho da **janela de recepção** na entidade TCP receptora.

Na internet existem dois problemas potenciais: a capacidade da rede e a capacidade do receptor. Para lidar com eles, cada entidade de uma conexão TCP mantém duas janelas: a janela deslizante de recepção e a janela deslizante de transmissão. Cada uma das janelas mostra o número de bytes que a entidade pode enviar: portanto, a janela efetiva de transmissão deve ser o mínimo entre o que o transmissor e o que o receptor consideram viáveis.

O protocolo TCP utiliza temporizadores de controle. O principal deles é o **temporizador de retransmissão**. Quando um segmento é enviado, um temporizador de retransmissão é disparado. Se o segmento for confirmado antes dele expirar, ele será travado. Se, por outro lado, ele expirar antes de a confirmação chegar, o segmento será retransmitido e o temporizador disparado mais uma vez.

Além deste, há ainda o **temporizador de persistência** (para garantir que o pedido de pausa de transmissão continua válido) e o **temporizador de *keep alive*** (para verificar se uma conexão sem uso continua ativa).

Cabeçalho TCP

Um segmento TCP consiste em duas partes: cabeçalho e dados. O cabeçalho tem uma parte fixa de 20 bytes e uma parte opcional de tamanho variável. O formato do cabeçalho fixo está mostrado na Figura 3.

Figura 3 - Formato do Cabeçalho do TCP

Fonte: Elaboração própria (2022).
Arte/ Diagramação: DME/FURB (2023).

Os campos deste cabeçalho são:

- **Origem**: identifica o ponto original da conexão (porta). Um endereço de porta e o endereço IP do equipamento formam um TSAP (*Transport Service Access Point*) único de 48 bits;
- **Destino**: identifica o ponto terminal da conexão (porta). Um endereço de porta e o endereço IP do equipamento formam um TSAP único de 48 bits;
- **Sequência**: identifica qual segmento que está sendo enviado, ou seja, qual parte da mensagem;
- **Confirmação**: identifica qual o próximo segmento que pode ser enviado avisando que a parte anterior foi recebida com sucesso;
- **Comprimento**: informa quantas palavras de 32 bits existem no cabeçalho e, consequentemente, o comprimento da sua parte opcional;

- **Controle**: são bits que permitem alguns tipos de controle especiais para o pacote: se o número de confirmação é válido, se é para esperar o segmento inteiro antes de enviar uma confirmação, se é um pedido de estabelecimento ou de conclusão de uma conexão;

- **Janela**: indica quantos bytes da janela deslizante de transmissão podem ser enviados a partir do último byte confirmado, de acordo com o quanto o destinatário está apto a receber e tratar;

- **Soma**: é a soma de verificação dos demais campos do cabeçalho e dos dados;

- **Ponteiro**: permite ao transmissor enviar ao receptor um sinal sem envolver um serviço específico do TCP.

Programação de Aplicações de Rede com TCP

Se estiver desenvolvendo um aplicativo de rede, como é que faço para utilizar o TCP? Pois saiba que uma aplicação de rede baseada no protocolo TCP consiste em um par de programas: um **programa servidor** e um **programa cliente**. Quando estes programas são executados, é criado um processo servidor e um processo cliente e estes dois processos se comunicam através da escrita e da leitura de *sockets* (TANENBAUM; WETHERALL, 2012).

Do ponto de vista da aplicação, uma conexão TCP é simplesmente um canal virtual direto entre um ***socket* cliente** e um ***socket* servidor**. Este canal de comunicação se mantém até que um dos processos decida encerrá-lo.

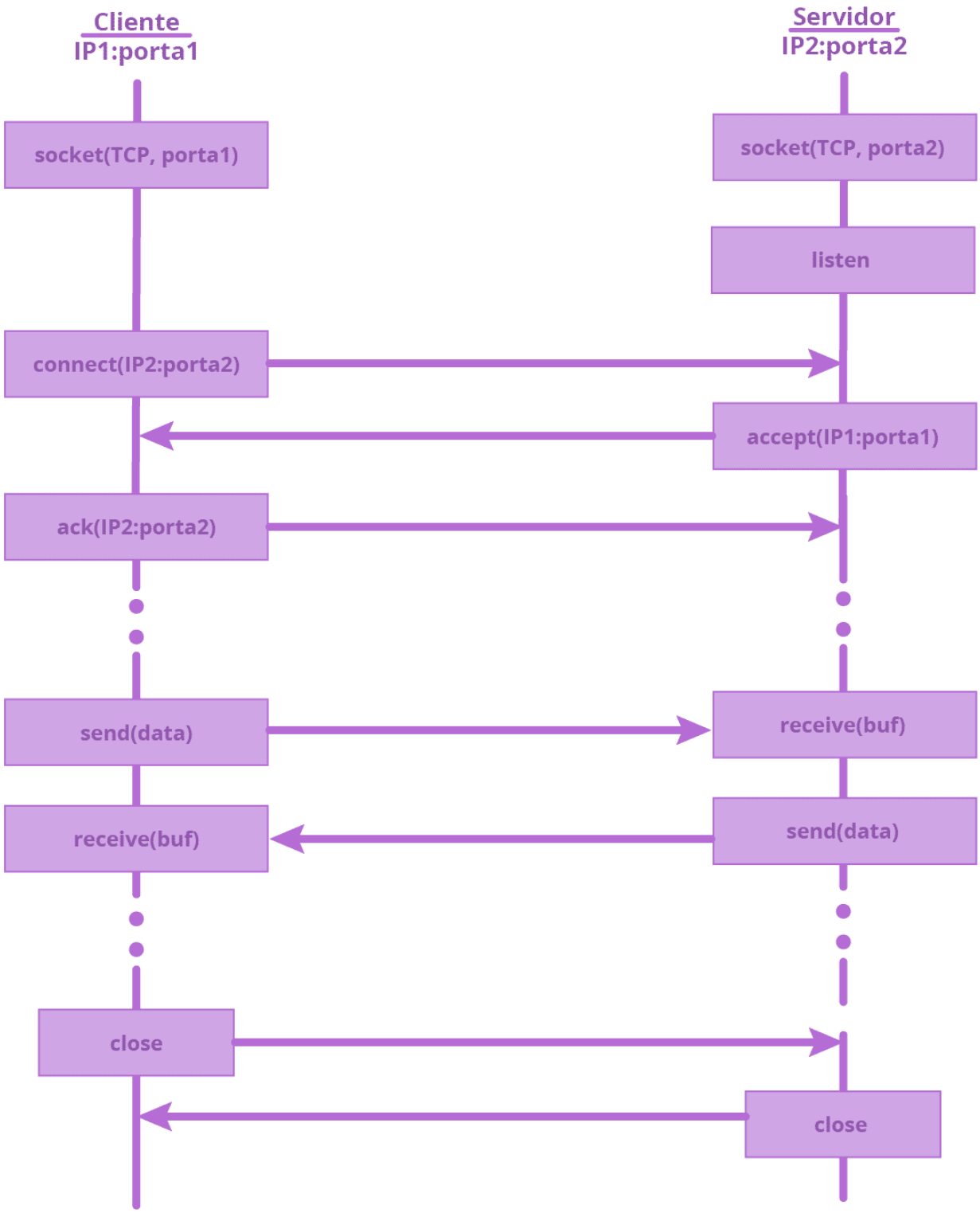
É importante ressaltar que em uma conexão TCP cabe à aplicação cliente contatar a aplicação servidora, a qual deve ter sido previamente inicializada e se encontrar em um estado de escuta por pedidos de conexão.

Para estabelecer uma conexão, o lado servidor aguarda passivamente por uma conexão executando a primitiva LISTEN.

O outro lado, o cliente, executa a primitiva CONNECT especificando o endereço IP e a porta (TSAP) com os quais deseja se conectar. Por fim, o lado servidor envia ao lado cliente uma primitiva ACCEPT indicando que o pedido de conexão foi aceito e recebe um ACK concluindo o processo de conexão.

A Figura 4 apresenta um diagrama que representa a sequência de criação de uma conexão TCP, o envio de dados entre o cliente e o servidor e o encerramento da conexão.

Figura 4 - Diagrama de Uma Conexão TCP



Fonte: Elaboração própria (2022).
Arte/ Diagramação: DME/FURB (2023).

CONTINUE

Atividade de Passagem

(ENADE) "No nível mais amplo, podem-se distinguir mecanismos de controle de congestionamento conforme a camada de rede ofereça ou não assistência explícita à camada de transporte com finalidade de controle de congestionamento." A respeito desse tema, avalie as asserções que se seguem e a relação proposta entre elas.
O protocolo de controle de transmissão (TCP) deve necessariamente adotar o método não assistido, no qual a camada de rede não fornece nenhum suporte explícito à camada de transporte com a finalidade de controle de congestionamento.

PORQUE

A camada de rede Internet Protocol (IP) não fornece realimentação de informações aos sistemas finais quanto ao congestionamento da rede.
Acerca dessas asserções, assinale a opção correta:

- ○ As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira

- As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira
- A primeira asserção é uma proposição verdadeira, e a segunda, uma proposição falsa
- A primeira asserção é uma proposição falsa, e a segunda, uma proposição verdadeira
- Tanto a primeira quanto a segunda asserções são proposições falsas

SUBMIT

(ENADE) O protocolo de controle de transmissão, ou TCP (Transmission Control Protocol), foi projetado especificamente para oferecer um fluxo de bytes fim a fim confiável em uma rede interligada não confiável. O TCP foi projetado para se adaptar dinamicamente às propriedades da rede interligada e ser robusto diante dos muitos tipos de falhas que podem ocorrer. A respeito das características do protocolo TCP, assinale a opção correta.

- Inicia e encerra conexões entre duas entidades por meio do handshake de três vias.
- Possui controle de congestionamento, evitando assim a ocorrência de tráfego excessivo na rede.
- Permite a comunicação entre entidades por meio de um serviço não orientado a conexões.
- Implementa o mecanismo de janela deslizante, permitindo o compartilhamento de uma mesma conexão.
- Adapta o tamanho do segmento TCP ao tamanho do MTU da camada de enlace, evitando assim a sua fragmentação ao longo do caminho.

SUBMIT

(ENADE) Em uma arquitetura cliente-servidor, para que uma aplicação ou serviço seja acessado, é necessário que uma comunicação seja estabelecida entre aquele que necessita (cliente) e aquele que disponibiliza o recurso

(servidor). Considerando o texto apresentado, avalie as asserções a seguir e a relação proposta entre elas.

I. Quando um programador desenvolve uma aplicação ou serviço de rede, é necessário utilizar um socket para unir o endereço IP da interface de rede com uma porta específica.

PORQUE

II. O socket possibilita ao sistema operacional identificar qual porta a aplicação está respondendo, permitindo que o fluxo de dados seja entregue corretamente.

A respeito dessas asserções, assinale a opção correta.

- As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.

As asserções I e II são proposições falsas.

SUBMIT

CONTINUE

Protocolo de Transporte UDP

Além do TCP, um protocolo orientado à conexão, a camada de transporte também abrange um protocolo **sem conexão**, o **UDP** (*User Datagram Protocol*), definido na RFC 768.

O UDP oferece uma forma das aplicações enviarem pacotes IP brutos encapsulados sem que seja necessário estabelecer e gerir uma conexão entre as entidades UDP de transmissão e de recepção: portanto o UDP oferece um serviço de transporte sem conexão.

O UDP pega as mensagens de um processo de um equipamento, anexa os campos de número de porta de origem e de destino para a multiplexação e demultiplexação, adiciona dois outros campos e passa o segmento resultante à camada de rede, que encapsula o segmento dentro de um pacote IP e, em seguida, faz uma tentativa de melhor esforço para entregar o segmento ao processo do equipamento receptor.

Reflita

Você consegue perceber quando é mais adequado para um desenvolvedor de aplicativos de rede usar o TCP ou usar o UDP?

Observe que apesar de o protocolo TCP prover um meio de comunicação de dados confiável, muitas aplicações são preferencialmente desenvolvidas utilizando o protocolo UDP pelas seguintes razões:

Não há estabelecimento de conexão

O UDP apenas envia segmentos sem qualquer formalismo preliminar;

Não há controle de estado de conexão

O UDP não tem parâmetros de controle de congestionamento, de sequência de segmentos e de confirmações;

Cabeçalho pequeno

O cabeçalho do UDP é composto por apenas 8 bytes, de fácil obtenção e processamento;

Taxa de transmissão sem controle

A taxa de transmissão de dados do UDP depende exclusivamente da capacidade de geração de dados da aplicação de origem, enquanto a taxa de recepção de dados pode ainda ser limitada pelo congestionamento da rede (pois parte dos dados pode ser perdida em função do congestionamento). Esta funcionalidade é importante para aplicação de multimídia em tempo real (KUROSE; ROSS, 2013).

1 Cabeçalho UDP

Um segmento UDP consiste em duas partes: cabeçalho e dados. O cabeçalho é composto por 8 bytes, como mostrado na Figura 5.

Figura 5 - Formato do Cabeçalho do UDP

Fonte: Elaboração própria (2022).
Arte/Diagramação: DME/FURB (2023).

Os campos deste cabeçalho são:

- **Origem**: identifica o ponto original do equipamento (porta). Um endereço de porta e o endereço IP do equipamento formam um TSAP (*Transport Service Access Point*) único de 48 bits;
- **Destino**: identifica o ponto terminal do equipamento (porta). Um endereço de porta e o endereço IP do equipamento formam um TSAP único de 48 bits;
- **Comprimento**: informa o comprimento do pacote, ou seja, do cabeçalho e dos dados;
- **Soma**: é a soma de verificação dos demais campos do cabeçalho e dos dados.

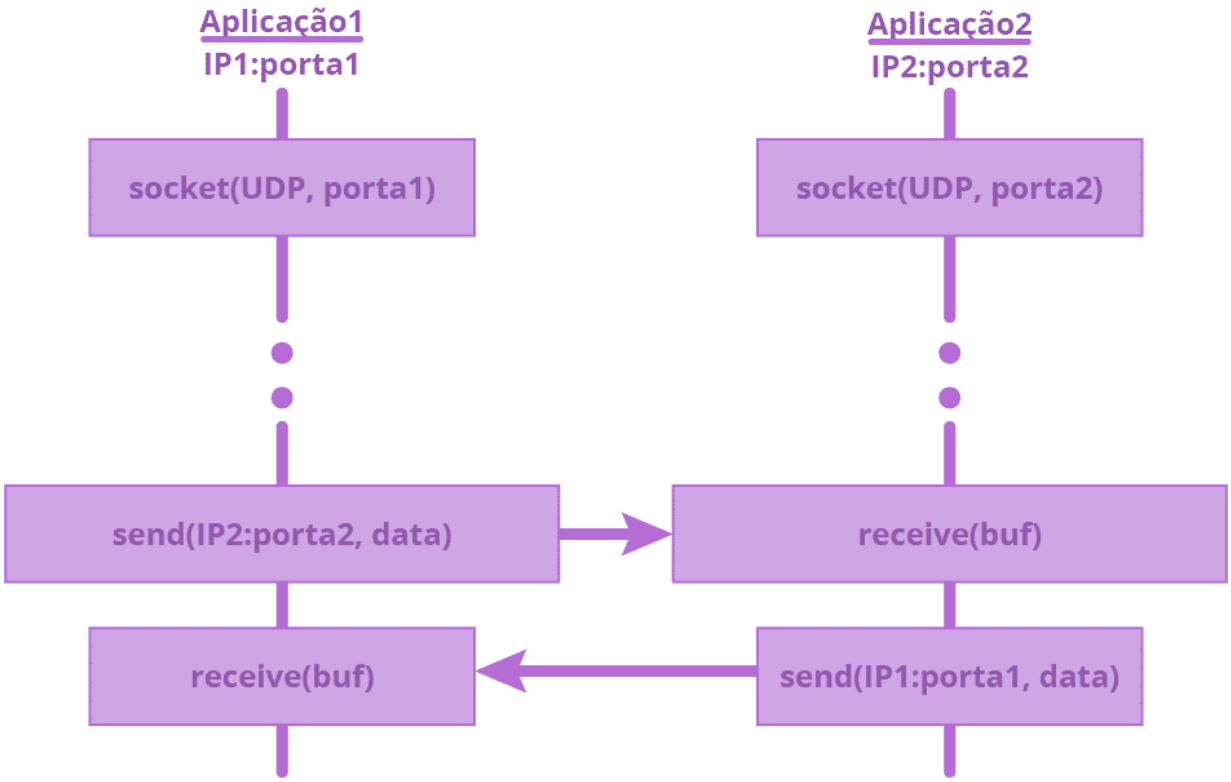
2 Programação de Aplicações de Rede com UDP

Como é que uso o UDP em um aplicativo que estiver desenvolvendo? Pois uma aplicação de rede baseada no protocolo UDP consiste em um par de programas para envio e recebimento de dados. Quando estes programas são executados, são criados dois processos que se comunicam através da escrita e da leitura de *sockets* (TANENBAUM; WETHERALL, 2012).

Do ponto de vista da aplicação, um *socket* UDP é simplesmente um meio de enviar dados a outro aplicativo sem que para isto seja previamente necessário estabelecer uma conexão com ele e, portanto, sem ativar mecanismos de QoS para a comunicação.

O mecanismo de funcionamento do protocolo de transporte UDP está representado no diagrama da Figura 6, que contém a sequência de envio de dados através da comunicação sem conexão.

Figura 6 - Diagrama de Uma Comunicação UDP



Fonte: Elaboração própria (2022).
Arte/Diagramação: DME/FURB (2023).

---

3 Protocolo de Transporte Multimídia RTP

Com o objetivo de se disponibilizar um protocolo de transporte de tempo real genérico para aplicações multimídia, criou-se uma especialização dos protocolos da camada de transporte da arquitetura internet chamada **RTP** (*Real-time Transport Protocol)*. O RTP, definido na RFC 1889, é utilizado para transportar segmentos com formatos de dados para multimídia. Os segmentos RTP são preferencialmente enviados através do protocolo UDP da camada de transporte.

Mas o que o RTP inclui de importante nos protocolos da camada de transporte? Pois saiba que com o RTP, cada segmento enviado em um fluxo de dados de áudio ou vídeo (chamado de fluxo contínuo de

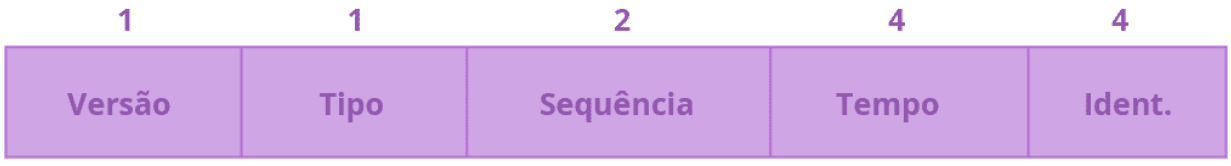
mídia) recebe um número sequencial que permite ao destino descobrir se algum segmento estiver faltando: neste caso, a melhor ação que o destino deve executar é fazer a aproximação do valor que falta por interpolação. A retransmissão não é aconselhável, pois o segmento retransmitido provavelmente chegaria tarde demais para ser útil. Como consequência, o RTP não tem nenhum controle de fluxo, nenhum controle de erros, nenhuma confirmação e nenhum mecanismo de solicitar retransmissões.

Muitas aplicações multimídia necessitam de um indicador de tempo para permitir que o destino realize algum tipo de armazenamento temporário do fluxo contínuo e reproduza cada segmento depois de um intervalo correto de tempo (contado desde o início do fluxo), independente de quando chegar o segmento. O uso do indicador de tempo elimina os efeitos da variação do tempo de atraso causado pela rede (*jitter*), assim como permite a sincronização de vários fluxos contínuos simultâneos de mídia.

Uma aplicação servidora de fluxo contínuo de mídia deve encapsular a mensagem em segmentos RTP (gerando as informações de temporização e de numeração sequencial para o cabeçalho) e enviá-los através de um *socket* UDP ou TCP à aplicação cliente no destino. A aplicação cliente recebe pelo *socket* o segmento RTP, extrai seu conteúdo e reconstitui o fluxo contínuo a partir das informações disponíveis no seu cabeçalho.

Um segmento RTP consiste em duas partes: cabeçalho e dados. O cabeçalho tem 12 bytes, cujo formato está mostrado na Figura 7.

Figura 7 - Formato do Cabeçalho do RTP



Fonte: Elaboração própria (2022).
Arte/Diagramação: DME/FURB (2023).

Os campos deste cabeçalho são:

- **Versão**: controla a versão do protocolo RTP e contém alguns bits de controle;
- **Tipo**: permite que a aplicação especifique o tipo de codificação do fluxo contínuo que está sendo transferido no campo de dados do pacote RTP;
- **Sequência**: é um contador dos pacotes gerados de forma que a aplicação cliente possa identificar a ordem dos pacotes e detectar eventuais pacotes perdidos durante a transmissão;
- **Tempo**: é um número que permite sincronizar a aplicação servidora e a cliente e eliminar o *jitter* de transmissão através da indicação temporal do momento em que foi gerado o sinal do fluxo contínuo;
- **Identificação**: identifica univocamente geradores de fluxo contínuo de forma que a aplicação cliente possa tratar mais de uma mídia independentemente em uma mesma sessão.

Há um protocolo que o RTP pode utilizar conjuntamente para controle de congestionamento e monitoração da qualidade de serviço (*QoS - Quality of Service*) da transmissão do fluxo contínuo de mídia denominado **RTCP** (*Real-time Transport Control Protocol*).

Pacotes RTCP são transmitidos por cada um dos participantes de uma sessão RTP para todos os demais participantes, normalmente através de um endereço IP de multidifusão. Em uma sessão RTP, normalmente há um único endereço de multidifusão para transmitir os pacotes RTP e RTCP pertencentes à sessão. Eles são enviados independentemente porque usam diferentes números de porta do protocolo de transporte.

CONTINUE

Atividade de Passagem

(ENADE) No desenvolvimento e na programação de aplicações em redes TCP/IP, qual tipo de protocolo de transporte libera o programador da responsabilidade de detectar e corrigir erros durante a transmissão, objetivando tornar a programação da aplicação mais simples?

- ◯ sem conexão
- ◯ orientado a conexão
- ◯ orientado a bit
- ◯ orientado a byte
- ◯ datagrama confirmado

SUBMIT

(ENADE) Em redes de computadores, a camada de transporte é responsável pela transferência de dados entre máquinas de origem e destino. Dois protocolos tradicionais para essa camada são o Transmission Control Protocol (TCP) e o User Datagram Protocol (UDP). Diferentemente do UDP, o TCP é orientado à conexão. Com relação a esses protocolos, avalie as afirmações a seguir.

I. O UDP é mais eficiente que o TCP quando o tempo de envio de pacotes é fundamental.

II. O TCP é o mais utilizado em jogos on-line de ação para a apresentação gráfica.

III. O TCP é mais eficiente que o UDP quando a confiabilidade de entrega de dados é fundamental.

É correto o que se afirma em:

- ◯ II, apenas.
- ◯ III, apenas.

- ( ) I e II, apenas.
- ( ) I e III, apenas.
- ( ) I, II e III.

SUBMIT

(ENADE) O protocolo RTP (Real Time Protocol, ou Protocolo de Tempo Real) é um dos protocolos usados nas transmissões em tempo real na Internet. Cada pacote RTP contendo um pedaço de uma mensagem multimídia codificada é encapsulado em um datagrama IP na origem e, ao chegar no destino, o conteúdo é reproduzido. Nesse cenário, um pacote íntegro será descartado no receptor se chegar atrasado, caracterizando as aplicações multimídia em tempo real como sensíveis ao atraso de pacotes. Acerca do funcionamento do protocolo RTP, avalie as asserções a seguir e a relação proposta entre elas.

I. Para lidar com possíveis atrasos de pacotes em trânsito na Internet, o protocolo RTP fornece um mecanismo para garantir a entrega de dados a tempo no destino.

PORQUE

II. Os roteadores conseguem distinguir e tratar datagramas IP que carregam pacotes RTP e assim estabelecer um esquema de QoS (Quality of Services, ou Qualidade de Serviços).

A respeito dessas asserções, assinale a opção correta.

- ○ As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- ○ As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- ○ A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- ○ A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- ○ As asserções I e II são proposições falsas.

SUBMIT

CONTINUE

Segurança de Camada de Transporte

Quando a segurança é disponibilizada pela camada de transporte, então todas as aplicações que utilizarem o protocolo estarão se beneficiando dos serviços de segurança desta camada. Há basicamente dois esquemas de segurança, operando sobre a camada de transporte, sendo utilizados: **SSL** (*Secure Sockets Layer*) e **TLS** (*Transport Layer Security*).

O SSL é uma interface de segurança desenvolvido para suprir sigilo e autenticação para a camada de transporte. Este nome deriva do fato de implementar uma interface de programação similar à do *socket* padrão (FORRISTAL; TRAXLER, 2002).

O SSL começa negociando o algoritmo e as chaves de criptografia simétrica e então autentica o cliente e o servidor através da troca de informações de um certificado digital presente no servidor. A partir de então, a transmissão dos dados entre cliente e servidor é criptografada com chaves simétricas válidas para a sessão segura. O SSL é utilizado no comércio eletrônico e no acesso bancário pela internet, sendo implementado em todos os navegadores e servidores WWW.

O TLS, definido inicialmente na RFC 2246, é uma evolução do SSL para aumentar ainda mais a sua segurança, pois alterou a forma com que a chave simétrica da sessão segura é gerada tornando-a ainda mais difícil de ser descoberta. O esquema de segurança TLS substituiu o esquema de segurança SSL.

Um exemplo é para o caso de se incluir a interface de segurança no protocolo HTTP *(Hypertext Transfer Protocol*) de navegação da internet: a implementação deste protocolo sobre o TLS é o protocolo **HTTPS** (*Hypertext Transfer Protocol Secure*).

Outro exemplo é para o caso de se incluir a segurança no protocolo SMTP (*Simple Mail Transfer Protocol*) de encaminhamento de mensagens de correio eletrônico pela internet: a implementação deste

protocolo sobre o TLS é o protocolo **SMTPS** (*Simple Mail Transfer Protocol Secure*).

CONTINUE

Atividade de Passagem

(ENADE) Os protocolos TLS (Transport Layer Security) e SSL (Secure Sockets Layer) utilizam algoritmos criptográficos para, entre outros objetivos, fornecer recursos de segurança aos protocolos comumente utilizados na Internet, originalmente concebidos sem a preocupação com a segurança nos processos de autenticação e/ou transferência de dados. Observada a pilha de protocolos TCP/IP, esses protocolos atuam:

- ◯ na camada de rede
- ◯ na camada de aplicação
- ◯ na camada de transporte
- ◯ entre a camada de transporte e a camada de rede
- ◯ entre a camada de aplicação e a camada de transporte

SUBMIT

CONTINUE

Resumo da Webaula 8

Como você viu nessa aula, a camada de transporte é uma complementação da camada de rede para prover formas de acompanhar o tráfego dos dados entre as aplicações de rede fim-a-fim, identificadas por um endereço de porta.

Você viu que existem dois tipos de protocolos nessa camada: o TCP, orientado a conexão, e o UDP, sem conexão. O primeiro tem a vantagem de garantir a integridade da transmissão implementando a segmentação, o controle de fluxo e o controle de congestionamento. Já o segundo deixa a cargo do programador implementar alguma dessas funcionalidades quando for preciso, mas por outro lado não tem limite de taxa de transmissão (portanto pode haver perda de alguns pacotes de dados conforme a capacidade da rede) e por isso é ideal de ser utilizado em aplicativos que usam conteúdo multimídia.

CONTINUE

Referências

O estudo das camadas do modelo de referência TCP/IP abordadas nesta parte do livro pode ser encontrado em:

KUROSE, James F; ROSS, Keith W. **Redes de computadores e a internet**: uma abordagem top-down. 6 ed. São Paulo: Pearson Education do Brasil, 2013.

MENDES, Douglas R. **Redes de Computadores:** teoria e prática. São Paulo: Novatec, 2010.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de computadores**. 5 ed. São Paulo: Pearson Prentice Hall, 2011.

Tópicos específicos sobre protocolos de comunicação entre aplicativos de rede: FORRISTAL, Jeff; TRAXLER, Julie. **Site seguro**: aplicações web. Rio de Janeiro: Alta Books, 2002.

As normas referenciadas nessa unidade podem ser obtidas diretamente da página da internet dos respectivos organismos de padronização:

- **IANA**: www.iana.org
- **ICANN**: www.icann.org
- **IETF**: www.rfc-editor.org/rfc-index.html
- **ISO**: www.iso.org/standards-catalogue/browse-by-ics.html

- **ITU-T**: www.itu.int/pub/T-REC
- **W3C**: www.w3.org/standards

CONTINUE

Créditos

Reitora

Profª. Ma. Marcia Cristina Sardá Espindola

Vice-Reitor

Prof. Dr. Marcus Vinicius Marques de Moraes

Pró-Reitor de Ensino de Graduação, Ensino Médio e Profissionalizante

Prof. Dr. Romeu Hausmann

Pró-Reitor de Administração

Prof. Me. Jamis Antônio Piazza

Pró-Reitora de Pesquisa, Pós-Graduação, Extensão e Cultura

Profª. Drª. Michele Debiasi Alberton

Divisão de Modalidades de Ensino Chefia da Divisão

Profª. Drª. Clarissa Josgrilberg Pereira

Professores Autores

Prof. Me. Francisco Adell Péricas

Design Instrucional

Profª. Drª. Clarissa Josgrilberg Pereira

Prof. Dr. Maiko Rafael Spiess

Prof. Me. Francisco Adell Péricas

Marcia Luci da Costa

Me. Wilson Guilherme Lobe Junior

Revisão Textual

Me. Wilson Guilherme Lobe Junior

Laura Cristina Zorzo

Roteirização

Laura Cristina Zorzo

Produção de Mídia

Gerson Luís de Souza

Gustavo Bruch Féo

Equipe de Design Gráfico

Amanda Kannenberg

Camylle Sophia Teske

Laura Cristina Zorzo

Nicolle Sassella

Renan Diogo Depiné Fiamoncini

Diagramado por Amanda Kannenberg em 09 de Fevereiro de 2023

CONTINUE