TEMA 07
Protocolo TCP e UDP

Habilidades

- Identificar as principais diferenças entre TCP e UDP
- Configurar e gerenciar sockets TCP e UDP em aplicações
- Compreender o modelo de três vias do TCP para estabelecimento de conexões
- Diagnóstico de problemas de rede relacionados ao TCP e UDP
- Avaliar o desempenho de aplicativos usando TCP ou UDP
- Implementar medidas de segurança para proteger a comunicação TCP e UDP

Para obter uma comunicação é necessário que ambas as pessoas falem a mesma língua, não é diferente com os computadores, quando um computador necessita envia uma mensagem a outro, ambos precisam estar "conversando" na mesma linguagem, ou seja, é necessário ter um protocolo para obter a comunicação, como no mundo real existem várias línguas, para os computadores existem vários protocolos.
Nos tempos antigos para enviar uma mensagem para outros países, era necessário recorrer ao telegrama ou um telefone.
Atualmente os computadores necessitam ter protocolos para possibilitar a conexão entre si, entre diversos protocolos que existam hoje em dia, dois se destacam pela sua importância e usabilidade que são o **TCP (Transmission Control Protocol)** e o **UDP (User Datagram Protocol)**.

O protocolo TCP

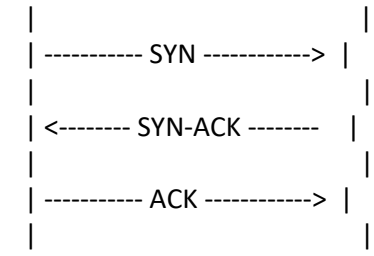
O protocolo TCP tem como característica ter uma transmissão confiável ponto-a-ponto, ou seja, se o {Computador (A) transmitir dados para o computador (B), será necessário **enviar um ACK (Bit de Reconhecimento)** para o computador (B), possibilitando a recuperação dos pacotes, rejeitando os pacotes duplicados e organizando os pacotes que forem chegando.
O TCP tem algumas características especificas como **ordenar os pacotes**, não possibilita o acumulo de pacotes na rede, e consegue transmitir ao mesmo momento a vários destinos diferentes, quando inicia uma conexão ou também a fecha.
Quando estabelecemos a comunicação com o protocolo TCP com duas máquinas, quem envia os pacotes é denominado Cliente, e quem recebe os pacotes é chamado de Servidor, sendo assim a comunicação é estabelecida nos dois sentidos.

A comunicação do Protocolo TCP é realizada em três etapas:

- **1° Etapa:** O Cliente envia o SYN (Uma solicitação de conexão, no qual possui um sequencial de bytes para Cliente-Servidor);
- **2° Etapa:** O Servidor aceita o pedido de conexão (SYN), e envia um pacote aceitando o pedido (ACK), no qual possui um sequencial de bytes para Servidor-Cliente;
- **3° Etapa:** O Destino (servidor) transmite o ACK permitindo o SYN, então a troca de informações é efetuada entre esses hosts, quando se encerra a conexão, tanto o cliente ou servidor podem finalizar a conexão, um dos dois pode enviar o pacote FIN (Fim da conexão), quando se recebe o FIN a máquina envia um ACK aceitando o fim da conexão e algum tempo depois (milésimos de segundos) ele envia o FIN, finalizando de fato a conexão.

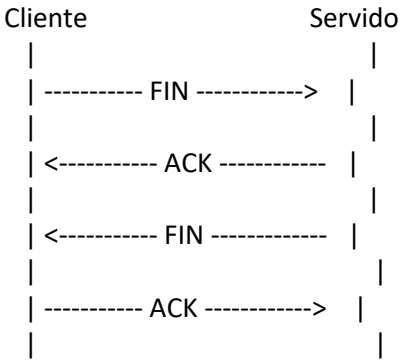
Obs.: Para cada pacote transmitido é inserido um checksum, que analisa se os pacotes estão danificados, ou perdidos durante a conexão, caso isso ocorra o protocolo retransmite os dados.

Cliente Servidor



Conexão estabelecida

Estabelecimento de Conexão (Three-Way Handshake) "aperto de mãos"
1) SYN: O cliente envia um segmento com o bit SYN ativado para iniciar a conexão.
2) SYN-ACK: O servidor responde com um segmento contendo tanto o bit SYN quanto o bit ACK ativados, confirmando o recebimento do SYN do cliente e sincronizando seu próprio número de sequência.
3) ACK: O cliente envia um segmento com o bit ACK ativado, confirmando o recebimento do SYN-ACK do servidor. A conexão está estabelecida e pronta para transferência de dados.



Conexão finalizada

Finalização de Conexão (Four-Way Handshake) "aperto de mãos"
1) FIN: O cliente envia um segmento com o bit FIN ativado, indicando que deseja terminar a conexão.
2) ACK: O servidor responde com um segmento contendo o bit ACK ativado, confirmando o recebimento do FIN do cliente.
3) FIN: O servidor envia um segmento com o bit FIN ativado, indicando que também deseja terminar a conexão.
4) ACK: O cliente responde com um segmento contendo o bit ACK ativado, confirmando o recebimento do FIN do servidor. A conexão está finalizada.

Protocolo UDP

O UDP é um protocolo que não é confiável, pois ele só envia os dados para o host de destino, por exemplo ele embala os datagramas e simplesmente envia, sem pedir confirmação de pacote ou verificar erros nos pacotes.
Este protocolo não dá garantias que o pacote será enviado até o seu destino, e muito menos que elas cheguem à ordem, pois não possui métodos para gerenciar os fluxos de dados.
Muitos pensam que por causa da falta de confiabilidade do protocolo UDP, que é melhor utilizar o TCP para obter a comunicação, entretanto vários serviços usufruem do protocolo UDP, pois **é mais rápido** o seu desempenho na rede.
O protocolo UDP é muito utilizado em serviço de multimídia (streaming e VOIP), pois se houver perda de dados não ocasiona delay na transmissão, se por acaso algum pacote for perdido, o mesmo não será perceptível para o usuário.

Portas TCP e Portas UDP

Existem várias portas TCP e UDP, e cada porta especifica um serviço na rede, de acordo com seu protocolo em questão.
Existem **65.536 probabilidades de portas**, isso porque são codificadas em 16 bits, por isso existe um órgão chamado IANA (Internet Assigned Numbers Authority) que parametriza essas portas e seus usos.
Porta de 0 a 1023 são as mais conhecidas, pois os administradores de rede ou usuário com privilégio acessam os seus serviços. Já as portas de 1024 até 49151 são denominadas de portas registradas, e as portas de 49153 a 65535 são as portas dinâmicas ou privadas.

Abaixo uma tabela com os serviços/portas mais utilizadas hoje em dia:

1. Porta 20: FTP (File Transfer Protocol) - TCP
2. Porta 21: FTP (File Transfer Protocol) - TCP
3. Porta 22: SSH (Secure Shell) - TCP
4. Porta 23: Telnet - TCP
5. Porta 25: SMTP (Simple Mail Transfer Protocol) - TCP
6. Porta 53: DNS (Domain Name System) - TCP/UDP
7. Porta 67: DHCP (Dynamic Host Configuration Protocol) - UDP
8. Porta 68: DHCP (Dynamic Host Configuration Protocol) - UDP
9. Porta 80: HTTP (Hypertext Transfer Protocol) - TCP
10. Porta 110: POP3 (Post Office Protocol) - TCP
11. Porta 123: NTP (Network Time Protocol) - UDP
12. Porta 143: IMAP (Internet Message Access Protocol) - TCP
13. Porta 161: SNMP (Simple Network Management Protocol) - UDP
14. Porta 194: IRC (Internet Relay Chat) - TCP/UDP
15. Porta 389: LDAP (Lightweight Directory Access Protocol) - TCP/UDP
16. Porta 443: HTTPS (HTTP Secure) - TCP
17. Porta 445: SMB (Server Message Block) - TCP
18. Porta 465: SMTP sobre SSL (Secure Mail Transfer Protocol) - TCP
19. Porta 514: Syslog - UDP
20. Porta 587: SMTP (Submission) - TCP
21. Porta 636: LDAPS (LDAP sobre SSL/TLS) - TCP
22. Porta 993: IMAP sobre SSL (IMAPS) - TCP
23. Porta 995: POP3 sobre SSL (POP3S) - TCP
24. Porta 1433: Microsoft SQL Server - TCP
25. Porta 1812: RADIUS (Remote Authentication Dial-In User Service) - UDP
26. Porta 1813: RADIUS Accounting - UDP
27. Porta 3306: MySQL - TCP
28. Porta 3389: RDP (Remote Desktop Protocol) - TCP/UDP
29. Porta 5060: SIP (Session Initiation Protocol) - TCP/UDP
30. Porta 5061: SIP sobre TLS - TCP

Esta lista cobre algumas das portas e serviços mais utilizados na rede, juntamente com os protocolos correspondentes.

Identificar as principais diferenças entre TCP e UDP:

TCP (Transmission Control Protocol) e UDP (User Datagram Protocol) são dois dos principais protocolos de transporte na pilha de protocolos da Internet. A principal diferença entre eles está na

forma como lidam com a entrega de dados e na confiabilidade da comunicação.
O TCP é um protocolo orientado à conexão e fornece uma comunicação confiável. Isso significa que ele estabelece uma conexão antes de transmitir dados, verifica se os pacotes são entregues em ordem e garante que não haja perda de dados durante a transmissão. Ele é usado em situações em que a **integridade dos dados é crucial**, como em transferências de arquivos, navegação na web e emails.

Por outro lado, o UDP é um protocolo orientado a datagramas, o que significa que ele não estabelece uma conexão antes de enviar dados. Ele simplesmente envia os pacotes de dados para o destino, sem garantir a entrega ou a ordem dos pacotes. O UDP é mais rápido e eficiente em termos de latência, tornando-o **ideal para aplicativos em tempo real, como videoconferência, streaming de áudio e jogos online**.

Latência, no contexto de redes de computadores e telecomunicações, refere-se ao tempo que um dado ou uma solicitação leva para viajar de um ponto a outro na rede. Em outras palavras, é o atraso que ocorre desde o momento em que uma solicitação é enviada até o momento em que a resposta é recebida. A latência é uma medida importante da qualidade e desempenho de uma rede, especialmente em aplicações que exigem resposta rápida, como jogos online, videoconferências e transações financeiras.

Configurar e gerenciar sockets TCP e UDP em aplicações:

Configurar e gerenciar sockets TCP e UDP é fundamental para desenvolvedores de aplicativos que precisam de comunicação em rede. Sockets são interfaces de programação que permitem que aplicativos se comuniquem por meio de redes.
Para configurar e gerenciar sockets TCP, os desenvolvedores precisam estabelecer uma conexão utilizando funções como `socket()`, `bind()`, `listen()` e `accept()`. Eles também devem implementar a lógica de envio e recebimento de dados usando `send()` e `recv()`.
No caso de sockets UDP, os desenvolvedores criam sockets usando `socket()` e usam `sendto()` e `recvfrom()` para enviar e receber datagramas UDP. É importante lembrar que, ao contrário do TCP, não é necessário estabelecer uma conexão prévia.
O gerenciamento de sockets envolve a liberação adequada de recursos, tratamento de exceções e, em muitos casos, a implementação de mecanismos de timeout para lidar com a perda de pacotes.

Compreender o modelo de três vias do TCP para estabelecimento de conexões:
O modelo de três vias (three-way handshake) é fundamental para o estabelecimento de conexões TCP. Ele consiste em três passos sequenciais:

1. O cliente inicia a conexão enviando um segmento TCP com a flag SYN (synchronize) definida para o servidor. Isso indica o desejo de iniciar uma conexão.
2. O servidor recebe o segmento SYN, confirma a solicitação do cliente e responde com um segmento que possui as flags SYN e ACK (acknowledge) definidas. Isso significa que o servidor está disposto a estabelecer uma conexão e reconhece a solicitação do cliente.
3. O cliente recebe o segmento de resposta do servidor e confirma a conexão enviando um segmento com a flag ACK definida. Agora, a conexão está estabelecida e ambos os lados podem começar a trocar dados.

Esse processo garante que ambos os lados estejam cientes da intenção de estabelecer uma conexão e sincronizados para a comunicação subsequente. É um elemento fundamental para a confiabilidade do TCP.

Diagnóstico de problemas de rede relacionados ao TCP e UDP:

Diagnosticar problemas de rede relacionados a TCP e UDP é uma habilidade crítica para administradores de rede e desenvolvedores de aplicativos. Alguns dos problemas comuns incluem **latência excessiva, perda de pacotes, congestionamento e configurações inadequadas**.
Para diagnosticar esses problemas, é importante utilizar ferramentas como o `ping` e o `traceroute` para testar a conectividade e identificar pontos de falha na rede. Para problemas específicos de aplicativos, a análise de logs e o uso de ferramentas de monitoramento de rede são essenciais.
Além disso, entender os diferentes comportamentos do TCP e UDP é fundamental. O TCP lida automaticamente com retransmissões de pacotes perdidos, **enquanto o UDP não o faz**. Portanto, ao diagnosticar problemas com UDP, é necessário implementar lógica de recuperação de pacotes no aplicativo.

Avaliar o desempenho de aplicativos usando TCP ou UDP:
Avaliar o desempenho de aplicativos que usam TCP ou UDP envolve medir diversos aspectos, como **latência, taxa de transferência e impacto na qualidade do serviço (QoS)**.
Para aplicativos que utilizam TCP, é importante avaliar a latência da conexão, pois o protocolo introduz alguma sobrecarga devido ao estabelecimento de conexão e ao controle de fluxo. A taxa de transferência também deve ser monitorada para garantir que atenda às necessidades da aplicação.
No caso de aplicativos UDP, a latência é crítica, especialmente para aplicativos em tempo real, como videoconferência e jogos online. Também é importante garantir que a taxa de transferência seja suficiente para a transmissão de dados em tempo real, sem atrasos perceptíveis.
Além disso, a QoS deve ser avaliada para garantir que a comunicação seja estável e livre de interrupções, independentemente de ser TCP ou UDP. Isso envolve o gerenciamento adequado da largura de banda e a priorização de pacotes.

Implementar medidas de segurança para proteger a comunicação TCP e UDP:
Implementar medidas de segurança para proteger a comunicação TCP e UDP é essencial para garantir a integridade e a confidencialidade dos dados transmitidos. Algumas das medidas comuns incluem:
- **Criptografia**: Usar protocolos como TLS/SSL para criptografar os dados transmitidos, garantindo que apenas os destinatários autorizados possam decifrá-los.
- **Autenticação:** Implementar mecanismos de autenticação, como senhas, tokens ou certificados, para garantir que apenas usuários autorizados tenham acesso à comunicação.
- **Firewalls:** Configurar firewalls para controlar o tráfego de entrada e saída, permitindo apenas o tráfego autorizado e bloqueando ameaças potenciais.
- **VPN (Rede Privada Virtual)**: Usar VPNs para criar túneis de comunicação segura por meio de redes públicas, protegendo os dados de interceptação.
- **Controle de acesso**: Implementar políticas de controle de acesso para determinar quem pode iniciar ou participar de conexões TCP ou UDP.
- **Monitoramento de segurança**: Utilizar ferramentas de monitoramento de segurança para detectar e responder a ameaças em tempo real.

A implementação adequada dessas medidas de segurança ajuda a proteger a comunicação TCP e UDP contra ameaças externas e garante que os dados sejam transmitidos com segurança.

RESUMO:

O Protocolo de Controle de Transmissão (TCP) e o Protocolo de Datagrama de Usuário (UDP) são **duas das principais tecnologias de comunicação utilizadas na internet**. Identificar as principais diferenças entre TCP e UDP é fundamental para escolher o protocolo adequado para uma aplicação

específica. Enquanto o TCP oferece uma conexão confiável e orientada a fluxo, garantindo que os dados sejam entregues na ordem correta e sem erros, o UDP é mais leve e rápido, porém menos confiável, já que não possui mecanismos de confirmação de entrega.
Configurar e gerenciar sockets TCP e UDP em aplicações é uma habilidade essencial para desenvolvedores de software e administradores de rede. Os sockets são os pontos de extremidade da comunicação e permitem que os dados sejam transmitidos entre dispositivos. Compreender o modelo de três vias do TCP para estabelecimento de conexões é crucial para garantir uma comunicação estável. Esse modelo envolve uma troca de mensagens entre cliente e servidor para estabelecer a conexão de forma segura.
O diagnóstico de problemas de rede relacionados ao TCP e UDP é uma habilidade crucial para manter a integridade da comunicação. Isso envolve a identificação de possíveis problemas, como perda de pacotes, congestionamento de rede ou falhas de roteamento, e a aplicação de soluções adequadas. Além disso, avaliar o desempenho de aplicativos usando TCP ou UDP é importante para garantir que a escolha do protocolo esteja alinhada com os requisitos de latência e confiabilidade da aplicação.
Implementar medidas de segurança para proteger a comunicação TCP e UDP é fundamental, especialmente em um cenário de ameaças cibernéticas crescentes. Isso inclui a criptografia dos dados transmitidos, autenticação de dispositivos e controle de acesso. Em resumo, dominar o TCP e o UDP envolve não apenas compreender suas diferenças e aplicações, mas também saber configurá-los, solucionar problemas de rede, otimizar o desempenho e garantir a segurança da comunicação. Essas habilidades são essenciais para profissionais de TI e desenvolvedores que trabalham com redes e aplicações online.

ATIVIDADES:

1. Qual é a principal diferença entre os protocolos TCP e UDP em termos de confiabilidade na entrega de dados?
2. Como o TCP garante a entrega ordenada de dados entre emissor e receptor?
3. Em que contexto ou cenários o TCP é preferível ao UDP na comunicação em rede?
4. Quais são as principais características do protocolo UDP e em que tipos de aplicativos ele é frequentemente utilizado?
5. Qual é o propósito da sequência de três vias (three-way handshake) no protocolo TCP e como ela é realizada?
6. Quais são as limitações do protocolo UDP em comparação com o TCP, e como essas limitações podem afetar aplicativos de tempo real?
7. Como o TCP lida com a retransmissão de pacotes perdidos e o controle de congestionamento?
8. Quando o uso de TCP pode levar a problemas de latência em aplicativos de alta sensibilidade ao tempo?
9. Qual é a diferença entre o modelo de comunicação orientada a conexão do TCP e o modelo de comunicação orientada a datagramas do UDP?
10. Quais medidas de segurança podem ser implementadas para proteger a comunicação via TCP e UDP e garantir a integridade e confidencialidade dos dados transmitidos?

Vamos às respostas!

1. A principal diferença entre os protocolos TCP e UDP em termos de confiabilidade na entrega de dados é que o TCP fornece uma entrega de dados confiável, garantindo que os dados sejam entregues na ordem correta e sem erros, enquanto o UDP não oferece garantias de entrega ou ordenação de pacotes.

2. O TCP garante a entrega ordenada de dados entre emissor e receptor através do seu mecanismo

de controle de fluxo e confirmação de recebimento. O emissor aguarda uma confirmação do receptor para cada pacote enviado antes de enviar o próximo, garantindo que os pacotes sejam entregues na ordem correta.

3. O TCP é preferível ao UDP em situações onde a integridade e a ordem dos dados são críticas, como em transferências de arquivos, navegação na web e emails. Ele é adequado para cenários em que é essencial garantir que os dados sejam entregues corretamente e sem erros.

4. O protocolo UDP é caracterizado pela sua falta de confiabilidade, pois ele simplesmente envia os dados para o destino sem garantias de entrega ou ordenação dos pacotes. Ele é frequentemente utilizado em aplicativos de tempo real, como streaming de áudio e vídeo, jogos online e VoIP, onde a latência é mais importante do que a integridade dos dados.

5. O propósito da sequência de três vias (three-way handshake) no protocolo TCP é estabelecer uma conexão confiável entre o cliente e o servidor antes de iniciar a troca de dados. Esse handshake consiste em três etapas: o cliente envia um segmento SYN (synchronize) para iniciar a conexão, o servidor responde com um segmento SYN-ACK (synchronize-acknowledgement) confirmando a solicitação do cliente e sincronizando seus próprios números de sequência, e finalmente o cliente envia um segmento ACK (acknowledgement) confirmando a resposta do servidor. Com essa troca de mensagens, ambas as partes estão cientes da intenção de estabelecer uma conexão e sincronizadas para a comunicação subsequente.

6. As limitações do protocolo UDP em comparação com o TCP incluem a falta de confiabilidade na entrega de dados, a ausência de controle de fluxo e retransmissão de pacotes perdidos, e a falta de garantias de ordenação dos pacotes. Essas limitações podem afetar aplicativos de tempo real, como videoconferência e jogos online, onde a perda ou atraso de pacotes podem impactar negativamente a experiência do usuário.

7. O TCP lida com a retransmissão de pacotes perdidos e o controle de congestionamento através de seus mecanismos de confirmação de recebimento, retransmissão seletiva e janela deslizante. Quando um pacote é perdido ou corrompido, o receptor solicita sua retransmissão ao emissor. Além disso, o TCP ajusta dinamicamente sua taxa de transmissão com base na disponibilidade de largura de banda e na ocorrência de congestionamento na rede.

8. O uso de TCP pode levar a problemas de latência em aplicativos de alta sensibilidade ao tempo quando há congestionamento na rede ou quando ocorrem retransmissões de pacotes perdidos. Isso ocorre porque o TCP espera pela confirmação de recebimento de cada pacote antes de enviar o próximo, o que pode introduzir atrasos significativos em situações de congestionamento.

9. A diferença fundamental entre o modelo de comunicação orientada a conexão do TCP e o modelo de comunicação orientada a datagramas do UDP está na confiabilidade da entrega de dados e no estabelecimento de conexões. O TCP estabelece uma conexão antes de enviar dados e garante a entrega ordenada e confiável dos dados, enquanto o UDP não estabelece uma conexão prévia e não oferece garantias de entrega ou ordenação dos pacotes.

10. Algumas medidas de segurança que podem ser implementadas para proteger a comunicação via TCP e UDP incluem criptografia dos dados transmitidos usando protocolos como TLS/SSL, autenticação de dispositivos e usuários, configuração de firewalls para controlar o tráfego de rede, uso de VPNs para criar túneis de comunicação segura, controle de acesso para determinar quem pode iniciar ou participar de conexões, e monitoramento de segurança para detectar e responder a ameaças em tempo real. Essas medidas ajudam a garantir a integridade e confidencialidade dos

dados transmitidos e protegem contra ameaças cibernéticas.