

## ATIVIDADE EM SALA - SISTEMAS DISTRIBUIDOS - PROF. VIEIRA

Nome: \_\_\_\_\_

Para a finalidade desta discussão, o contexto terá a atenção limitada às redes TCP-IP. O modelo OSI é muito bom, mas nunca foi completamente implementado, muito menos amplamente implantado em condições de elevado stress e de alto tráfego.

### SOQUETE

Um termo como "soquete" parece ser auto-explicativo: ele foi obviamente escolhido para evocar imagens de terminação na qual um cabo de rede é conectado, havendo fortes paralelos funcionais. No entanto, no jargão da rede, a palavra "soquete" carrega tanta bagagem que um re-exame cuidadoso é necessário.

Uma *conexão* TCP é definida por dois pontos finais também conhecidos por *soquetes*.

Na maioria dos linguagens derivadas de C, as conexões TCP são estabelecidas e manipulados usando métodos em uma instância de uma classe Soquete. Embora seja comum a operação em um nível mais alto de abstração, tipicamente uma instância de uma classe *NetworkStream*, geralmente expõe uma referência a um objeto do tipo "soquete". Para o programador, este objeto "soquete" parece representar uma conexão, porque a conexão é criada e manipulada usando métodos associados ao objeto soquete.

Em C #, para estabelecer uma conexão TCP (para um ouvinte existente), primeiro cria-se um *TcpClient*. Se não for especificado um parâmetro para o construtor *TcpClient*, ele utilizará os padrões - de uma forma ou de outra, o ponto da extremidade local está definido. Em seguida, o método *Connect* é chamado (considere como um exemplo sendo criado). Este método requer um parâmetro que descreve a outra extremidade (remota).

Tudo isto pode ser um pouco confuso e pode levar a acreditar que um soquete é uma conexão, o que não é verdade.

*(TCP-IP Illustrated Volume 1, W. Richard Stevens)* O soquete especifica os dois pontos finais que identificam cada conexão TCP na internet, composto de dois pares de elementos:

- o endereço do cliente IP + o número da porta do cliente, e
- o endereço IP do servidor + o número da porta do servidor.

Faz mais sentido ter uma classe *TcpConnection* com um construtor que recebe dois argumentos, *LocalEndpoint* e *RemoteEndpoint*. Um único argumento *RemoteEndpoint* pode ser suficiente quando os padrões para o parâmetro local são aceitos (*LocalEndpoint*). Esta situação é ambígua em computadores com diversas bases, mas a ambigüidade pode ser resolvida usando a tabela de roteamento, selecionando a interface com o caminho mais curto para o terminal remoto.

Uma *conexão* TCP é definido por dois pontos finais, também denominados *soquetes*. A combinação de um endereço IP e uma porta é rigorosamente conhecida como um ponto final e às vezes é chamado (erroneamente) de um soquete, mas um soquete *não* é identificado pela combinação de endereço IP e uma porta. Essa terminologia se originou com a especificação TCP inicial, a RFC793.

## PORTA (SW)

Em sentido mais amplo possível, uma porta é um ponto de entrada ou saída. Portas, então, são terminais de transporte se você enviar dados ou grandes recipientes de aço. A porta está sempre associado a um endereço IP de um hospedeiro (host) e o tipo de protocolo da comunicação. Ela completa o destino ou origem endereço de uma sessão de comunicações. A porta é identificada para cada endereço de protocolo por um número de 16 bits, conhecido como o **número da porta**

Um ponto de terminação em uma rede (soquete) é definido pela combinação de um endereço de rede e um identificador de *porta*. Note-se que o endereço/porta *não* identifica completamente um soquete.

O objetivo das portas é diferenciar vários pontos de extremidade em um determinado endereço de rede. Pode-se dizer que uma porta é um endpoint virtualizado. Esta virtualização faz várias ligações simultâneas em uma única interface de rede possível.

Entre as milhares de portas enumerados, 1024 números de porta bem conhecidos são reservados por convenção para identificar os tipos de serviços específicos em um host. Os protocolos que utilizam principalmente os portos são os de camada de transporte protocolos, como o Protocolo de Controle de Transmissão (TCP) e o User Datagram Protocol (UDP) do Internet Protocol Suite .

Um exemplo para a utilização de portas é o sistema de correio de Internet . Um servidor usado para enviar e receber e-mail, geralmente precisa de dois serviços: um para transportar o e-mail para e a partir de outros servidores, e outro para buscar mensagens de e-mail do servidor.

O primeiro serviço é executado com o Simple Mail Transfer Protocol (SMTP), aplicativo de serviço que normalmente monitora a porta TCP 25 para receber os pedidos. O segundo serviço é normalmente o Post Office Protocol (POP) ou o Internet Message Access Protocol (IMAP), que é usado por clientes de e-mail e aplicações em computadores pessoais dos usuários. O serviço POP monitora a porta TCP 110. Ambos os serviços podem estar em execução no mesmo computador host, seja ele um computador do usuário ou em outro servidor de correio.

Enquanto o número da porta de monitora um servidor é bem definida (o IANA define estas portas), o número da porta do cliente é muitas vezes escolhido a partir do intervalo de porta dinâmica. Em algumas aplicações, os clientes e o servidor usam números de porta específicas e atribuídas pelo IANA. Um bom exemplo disso é o DHCP em que o cliente sempre usa a porta UDP 68 e o servidor sempre usa a porta UDP 67

1 - Quais temas sobre Soquetes e Portas que você considera importante abordar, seja em Redes ou em outro tema do curso?

2 - De acordo como texto, a programação de soquetes permite alterar quais parâmetros?

3 - Em uma única linha de texto, estabeleça a relação entre soquetes e portas (SW).

4 - Qual a competência do IANA (Internet Assigned Numbers Authority)?