Aula 4 – Protocolos de redes de computadores

Objetivos

Entender o funcionamento dos principais protocolos utilizados nas redes de computadores.

Compreender quais protocolos são utilizados em cada camada.

Conhecer o funcionamento dos protocolos mais usuais no dia-a-dia dos usuários.

Ter o entendimento do endereçamento IP em suas versões 4 e 6.

4.1 Considerações iniciais

Protocolos em sua essência são regras e procedimentos de comunicação. Na comunicação em redes de computadores os protocolos definem as regras que os sistemas precisam seguir para comunicar-se entre si. Já, os pacotes são conjuntos de *bits* ou sinais que são agrupados de forma que possam trafegar pelo meio de transmissão (MORAES, et al., 2003).

Os protocolos não dependem da implementação, o que significa que sistemas e equipamentos de fabricantes diferentes podem comunicar-se, desde que sigam as regras do protocolo. Dessa forma, os protocolos da arquitetura TCP/IP estão organizados em uma pilha de protocolos, a exemplo da organização em camadas da arquitetura. No Quadro 4.1 são apresentados os principais protocolos de rede e as camadas de operação onde os mesmos atuam.

Quadro 4.1: Principais protocolos de rede e camadas de operação

Camada	Principais protocolos
Aplicação	HTTP, DNS, SSH
Transporte	TCP, UDP
Internet	IP
Interface de rede	Ethernet

Fonte: Moraes, et al., 2003

4.2 Protocolos da camada de aplicação

Nessa seção serão abordados os principais protocolos da camada de aplicação, bem como, suas características e aplicabilidade. Os protocolos pertencentes a esta camada são responsáveis pela funcionalidade das aplicações utilizadas pelo usuário.

4.2.1 HTTP

O protocolo de transferência de hipertexto (HTTP – *HiperText Transfer Protocol*) é o principal protocolo da *World Wide Web* (WWW) ou simplesmente *web*. O HTTP é usado na *web* para a comunicação e transferência de documentos HTML (*HiperText Markup Language*) entre um servidor *web* e um cliente. O HTTP é um protocolo da camada de aplicação e usa o protocolo TCP para o transporte dos documentos e das mensagens (pedidos e respostas).

Baseado no modelo de arquitetura cliente/servidor e no paradigma de requisição e resposta, o HTTP é responsável pelo tratamento de pedidos e respostas entre um cliente e um servidor. Além disso, utiliza como padrão a porta 80.

O protocolo HTTP é a base da funcionalidade da internet. Construído sob o modelo de referência TCP/IP é caracterizado como um protocolo veloz, leve e orientado à conexão.

As portas utilizadas na comunicação de dados, como por exemplo, a porta 80 para internet (http), a porta 443 para (https), são endereçadas na camada de transporte do modelo de referência OSI.

4.2.2 SMTP

Protocolo responsável pelo envio de *e-mails*, o SMTP (*Simple Mail Transfer Protocol*) realiza a comunicação entre o servidor de *e-mails* e o computador requisitante. Este protocolo utiliza por padrão a porta 25.

O protocolo SMTP tem a função de somente enviar *e-mails* (a um destinatário ou mais) fazendo a transmissão do mesmo. Para recebimento das mensagens de um servidor utiliza-se outro protocolo, o POP3 que tem a função de receber mensagens do servidor para o programa cliente de *e-mail* do usuário (Outlook, entre outros).

Para que seja efetivado o envio de *e-mails* através deste protocolo, uma conexão é estabelecida entre o computador cliente e o servidor responsável pelo envio de *e-mails* (servidor SMTP, devidamente configurado).

4.2.3 POP3

Responsável pelo recebimento de *e-mails*, o protocolo POP3 (*Post Office Protocol*) controla a conexão entre um servidor de *e-mail* e o cliente de *e-mail*. De modo geral, sua função é permitir "baixar" todos os *e-mails* que se encontram no servidor para sua caixa de entrada.

O protocolo POP3 realiza três procedimentos básicos durante sua operação de recebimento de *e-mails* que são: **autenticação** (realizada geralmente pelo nome de usuário e uma senha), **transação** (estabelecimento de conexão cliente/servidor) e **atualização** (finalização da conexão cliente/servidor).

Existem duas formas básicas de enviar e receber *e-mails*. A primeira delas é utilizar um cliente de *e-mail* como o Outlook Express, Apple Mail, Kmail, entre outros. Para isso é necessário configurar manualmente os servidores de envio (SMTP) e recebimento de mensagens (POP3). As vantagens deste tipo de serviço são: leitura e escrita de *e-mails* em modo *off-line*, armazenamento de *e-mails* no próprio computador do usuário, entre outros. Em contrapartida existem os chamados *webmails* que utilizam a própria estrutura da internet para acessar os *e-mails* através de um endereço da *web* específico, como: **http://webmail.exemplo.com.br**. As vantagens deste tipo de serviço são a centralização dos recursos de *e-mail* (contatos, *e-mails* enviados, recebidos) bem como a utilização de múltiplas contas e personalizações mais simplificadas que podem ser aplicadas (SILVA, 2010).

4.2.4 FTP

O protocolo FTP (*File Transfer Protocol*) é utilizado na transferência de arquivos cliente/servidor, tanto para *download* quanto *upload* de arquivos. Para tal procedimento este protocolo utiliza as portas 20 e 21. A porta 20 é utilizada para transmissão de dados, enquanto que a porta 21 é utilizada para controle das informações.

Os serviços de FTP subdividem-se em: servidores e clientes de FTP.

Os servidores de FTP permitem criar uma estrutura (serviço) onde é possível acessar via navegador, por exemplo, um endereço específico ao serviço (Ex.: ftp.exemplo.com.br) e fazer *upload* e/ou *download* de arquivos de forma *on-line*. Este tipo de servidor de FTP pode ser privado (na qual exige uma autenticação do usuário, mediante nome de usuário e senha) ou público, onde o acesso não necessita autenticação para acesso aos serviços.

Já os clientes de FTP, são programas instalados no computador do usuário, utilizados para acessar os servidores de FTP de forma personalizada. São exemplos destes programas aplicativos: Filezilla, Cute FTP, WS FTP, entre outros.

Vale ressaltar que todos os *browsers* (navegadores) possuem suporte para acesso FTP, dessa forma, a utilização de programas externos de FTP não é obrigatória e sim uma opção caso o usuário achar mais conveniente.

4.2.5 DNS

O Sistema de Nomes de Domínio (DNS – *Domain Name System*) é um esquema hierárquico e distribuído de gerenciamento de nomes. O DNS é usado na internet para manter, organizar e traduzir nomes e endereços de computadores. Na internet toda a comunicação entre dois computadores de usuários ou servidores é feita conhecendo-se o endereço IP da máquina de origem e o endereço IP da máquina de destino. Porém, os usuários preferem usar nomes ao se referir a máquinas e recursos.

Os computadores dispostos em uma rede de computadores são identificados por seu número IP (endereço lógico) e seu endereço MAC (identificação física, designada na fabricação do dispositivo de rede). Os endereços IP na versão 4 (IPv4), compostos de 32 *bits*, geralmente são difíceis de serem memorizados, conforme aumenta a quantidade de computadores na rede, servidores, entre outros. Como forma de facilitar a memorização de computadores, *sites*, servidores e demais dispositivos que trabalham com a numeração IP, foi criado o sistema DNS, que torna possível relacionar nomes aos endereços IP, realizando a troca (endereço por nome). Dessa forma, torna-se mais simples lembrar um determinado endereço (www.exemplo.com.br) do que um número IP relacionado ao domínio (como por exemplo: 200.143.56.76).

O funcionamento do DNS baseia-se em um mapeamento de IPs em nomes. Estes ficam armazenados em tabelas dispostas em banco de dados nos servidores DNS. Nestes servidores são realizadas as trocas de endereços IP em nomes e vice-versa.

A estrutura de nomes na internet tem o formato de uma árvore invertida onde a raiz não possui nome. Os ramos imediatamente inferiores à raiz são chamados de TLDs (*Top-Level Domain Names*) e são por exemplo ".com", ".edu", ".org", ".gov", ".net", ".mil", ".br", ".fr", ".us", ".uk", etc. Os TLDs que não designam países são utilizados nos EUA. Os diversos países utilizam a sua própria designação para as classificações internas. No Brasil, por exemplo, temos os nomes ".com.br", ".gov.br", ".net.br", ".org.br" entre outros.

Cada ramo completo até a raiz como, por exemplo, "puc-rio.br", "acme.com.br", "nasa.gov", e outros, são chamados de domínios. Um domínio é uma área administrativa englobando ele próprio e os subdomínios abaixo dele. Por exemplo, o domínio ".br" engloba todos os subdomínios do Brasil.

- **Hierarquia de nomes**

Uma hierarquia de nomes é utilizada para caracterizar o uso de cada extensão do domínio. No Quadro 4.2, são caracterizados alguns dos principais domínios utilizados e seu respectivo significado.

Quadro 4.2: Tipos de domínios

Nome do domínio	Significado
com	Organizações comerciais
edu	Instituições educacionais
gov	Instituições governamentais
mil	Agências militares
net	Organizações da rede
org	Organizações não comerciais
int	Organizações internacionais
Código de países	Identificador de 2 letras para domínios de países específicos

Fonte: Tanenbaum, 2003

O registro.br (www.registro.br) é a entidade nacional que trata do registro de domínios para a internet no Brasil, ou seja, que estão sob a faixa ".br". Dessa forma, ao registrar um novo domínio na internet com a extensão final ".br" é necessário consultar se o domínio em questão não está registrado e se o mesmo é possível de ser registrado (**www.registro.br**). Para registrar um novo domínio, além de cadastrar-se no portal é necessário informar onde este domínio ficará hospedado (servidor de hospedagem), bem como pagar uma taxa anual para exercer a utilização deste domínio.

4.2.6 DHCP

O protocolo DHCP (*Dynamic Host Configuration Protocol*), possui a função de distribuir a gerenciar endereços IP em uma rede de computadores. Mais do que isso, este protocolo em conjunto com um servidor DHCP é capaz de distribuir endereços, *gateway*, máscaras, entre outros recursos necessários a operação e configuração de uma rede de computadores.

Para que o DHCP possa operar de forma plena é necessário:

- Que o computador cliente (que necessita de um número IP) possua o pacote DHCP cliente instalado.

- A partir deste momento o computador cliente envia uma requisição (pacote) na rede solicitando um número IP (requisição DHCP).

- Cabe a um servidor DHCP disponível na rede responder a requisição do computador solicitante, com um pacote contendo o endereço IP, *gateway* padrão, máscara de rede, servidores de DNS, entre outros.

Um servidor DHCP, utiliza o modelo cliente/servidor, mantendo o gerenciamento centralizado dos IPs utilizados pelos dispositivos conectados a rede.

4.2.7 SNMP

O protocolo SNMP (*Simple Network Management Protocol*), ou Protocolo Simples de Gerência de Rede tem a função de monitorar as informações relativas a um determinado dispositivo que compõe uma rede de computadores.

É através do protocolo SNMP que podemos obter informações gerais sobre a rede como: placas, comutadores, *status* do equipamento, desempenho da rede, entre outros. A obtenção destas informações é possível graças a um *software* denominado agente SNMP presente nos dispositivos de rede, que extrai as informações do próprio equipamento, enviando os mesmos para o servidor de gerenciamento. Este por sua vez recebe as informações, armazena e analisa.

4.2.8 SSH

O protocolo SSH (*Secure Shell*), tem uma função importante na pilha de protocolos da camada de aplicação que é permitir a conexão segura (criptografada) a outro computador (da mesma rede ou de outra rede distinta) e poder controlá-lo (dependendo do nível de acesso e privilégios) remotamente. Esta função de acessar um computador distante geograficamente e poder utilizá-lo/manipulá-lo como se o usuário estivesse presente fisicamente em frente do computador e ainda de forma criptografada, faz com que o protocolo SSH seja utilizado amplamente nas redes de computadores.

Existem diversos programas aplicativos que permitem gerenciar computadores *desktop* e servidores a distância e através de um outro computador ou a partir de seu próprio *smartphone*. A seguir, alguns exemplos destes programas aplicativos de administração remota de computadores.

- OpenSSH (utilizado para a plataforma Linux, tanto para máquinas clientes (que geram a conexão) como máquinas servidoras (que recebem as conexões através da linha de comandos).

- Putty (*software* amplamente conhecido na administração remota de computadores possui versões do aplicativo tanto para Linux quanto para sistemas operacionais Windows).

- WebSSH (aplicativo *on-line* que permite a conexão a um computador remoto sem a necessidade de instalação de aplicativos clientes).

O protocolo SSH opera por padrão na porta 22, sendo possível e indicado a sua modificação (alteração nas configurações do servidor) para operação em uma porta diferente (por questões de segurança).

4.3 Protocolos da camada de transporte

Na arquitetura TCP/IP, a camada de transporte encontra-se logo abaixo da camada de aplicação e diretamente provê um serviço para esta camada. A camada de Transporte oferece um serviço de circuito virtual fim-a-fim entre uma entidade (processo ou aplicação) na máquina de origem e outra entidade na máquina de destino.

Um conceito importante introduzido na camada de transporte da arquitetura TCP/IP é o de portas. As portas provêm um mecanismo interessante para identificação e endereçamento correto dos pacotes aos processos correspondentes nas máquinas de origem e de destino. Cada aplicação, normalmente, está associada a uma porta conhecida pelas máquinas de origem e destino.

Os dois principais protocolos da camada de transporte, o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*) oferecem as aplicações em diferentes níveis de serviço e confiabilidade. Normalmente cada aplicação usa um dos dois protocolos, conforme a necessidade de confiabilidade e desempenho, para transporte das mensagens geradas na aplicação do cliente e do servidor. Nessa seção analisaremos mais detalhadamente esses dois principais protocolos.

4.3.1 O TCP (*Transmission Control Protocol*)

O TCP (*Transmission Control Protocol* – Protocolo de Controle de Transmissão) é o protocolo mais importante da camada de transporte e juntamente com o IP (*Internet Protocol*), da camada de rede, forma a dupla de protocolos mais

importantes na arquitetura do TCP/IP. O TCP permite a criação de um canal virtual confiável, livre de erros, fim-a-fim, entre uma aplicação ou serviço na máquina origem e uma aplicação na máquina de destino.

O TCP é um protocolo robusto e confiável, por isso um grande número de aplicações dos usuários faz uso deste para transferência de dados. Algumas características importantes do TCP são:

- **Orientado a conexão** – significa que antes que qualquer transmissão de mensagens ou dados da aplicação seja feita, a camada de transporte, por meio do TCP, deve estabelecer uma conexão. Basicamente, uma conexão é estabelecida após o envio de um pedido de conexão de uma das máquinas envolvidas e a confirmação de ambas. Somente após o estabelecimento da conexão é que as mensagens da aplicação começam a ser enviadas. Todos os pacotes de dados trafegados após o estabelecimento da conexão são associados com uma conexão específica.

- **Ponto-a-ponto** – uma conexão é estabelecida entre duas entidades, mais especificamente, ligando um processo na máquina de origem e um processo na máquina de destino.

- **Confiabilidade** – o TCP usa um mecanismo para tratar erros durante a transmissão, como pacotes perdidos ou pacotes com dados corrompidos. Todos os pacotes transmitidos devem ser confirmados pelo receptor. Simplificadamente, a falta de uma confirmação do receptor, significa que o pacote foi perdido no caminho e deve ser automaticamente retransmitido. O TCP usa uma soma de verificação (*checksum*) em campo de cabeçalho (Figura 4.1), que é verificado pelo receptor. Se a soma de verificação não estiver correta, significa que os dados foram corrompidos no caminho, o pacote é descartado e a origem deve retransmitir o pacote.

- ***Full-duplex*** – transferência simultânea em ambas as direções, envio e recebimento ao mesmo tempo.

- **Entrega ordenada** – o TCP possui um campo de cabeçalho para identificação da sequência (Figura 4.1) do pacote dentro da conexão. Mesmo que os pacotes cheguem fora de ordem no destino, a mensagem da aplicação é reconstruída na ordem correta.

- **Controle de fluxo** – o TCP usa um campo Janela (Figura 4.1) para determinar a quantidade de dados que o receptor pode receber e processar.

Quando o emissor recebe uma confirmação de um pacote enviado, juntamente ele toma conhecimento do tamanho da janela de dados que o receptor pode trabalhar. Esse mecanismo de controle de fluxo evita que o emissor envie pacotes excessivamente, congestionando o receptor.

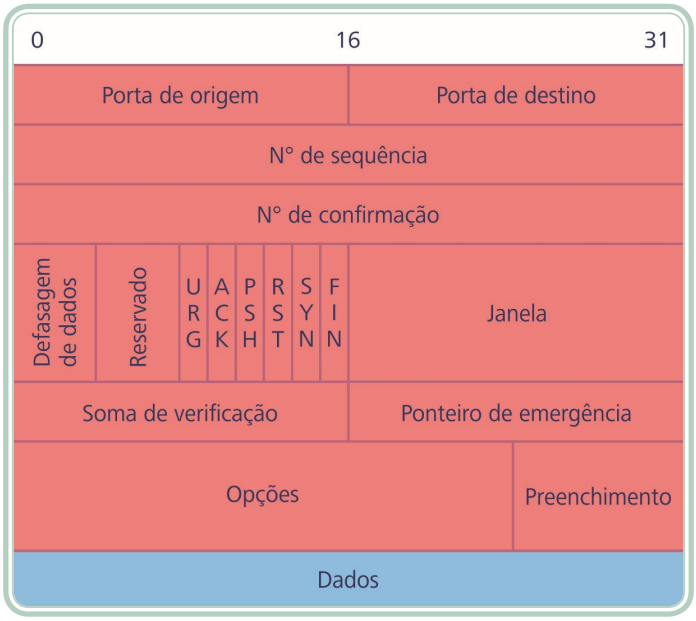

Figura 4.1: Representação dos campos de cabeçalho de um pacote TCP
Fonte: CTISM, adaptado de Tanenbaum, 2003

A Figura 4.1 traz uma representação dos campos de cabeçalhos de um pacote do TCP. Cada campo do cabeçalho tem uma função no funcionamento do TCP.

Para a criação de uma conexão TCP, normalmente são necessários um serviço (processo) rodando em uma máquina servidora, "escutando" em uma porta conhecida e uma aplicação (outro processo) em uma máquina cliente. Um serviço "escutando" em uma porta significa que o processo fica esperando um pedido de conexão nesta porta.

Na outra ponta deverá haver uma aplicação no cliente desejando iniciar uma conexão usando uma porta de origem qualquer. As conexões estabelecidas no cliente ou no servidor são associadas a *sockets*, identificados por: endereço IP de origem (no cabeçalho do protocolo IP), porta TCP de origem (cabeçalho do protocolo TCP, Figura 4.1), endereço IP de destino e porta TCP de destino. *Sockets* permitem a ligação entre a camada de transporte, neste caso pelo protocolo TCP e um processo da camada de aplicação, para o envio e o recebimento de mensagens da aplicação.

Tipicamente, uma conexão TCP envolve três fases: estabelecimento da conexão, transferência de dados e finalização da conexão.

O estabelecimento de uma conexão TCP inicia-se com um cliente desejando estabelecer uma conexão em um servidor já esperando por um pedido de conexão. Uma conexão TCP bem sucedida envolve a troca de uma sequência de pacotes especiais, com *flags* especiais de cabeçalho setadas (*bit* igual a 01) (Figura 4.1):

- O **cliente** requisita uma conexão enviando um pacote TCP especial, com a *flag* **SYN** (***synchronize***) do cabeçalho setada ao servidor. Esse pacote é conhecido simplificadamente como pacote do tipo **SYN.**

- O **servidor** confirma esta requisição respondendo com um pacote do tipo **SYN-ACK** ao cliente, ou seja, um pacote TCP com as *flags* de cabeçalho **SYN** e **ACK** setadas.

- O **cliente** por sua vez responde com um pacote do tipo **ACK**, *flag* **ACK** setada, e a conexão é estabelecida. Essa sequência é conhecida como **aperto de mão em três etapas** (*Three-Way Handshake*).

Somente após esse processo inicial (*Three-Way Handshake*) a conexão está disponível para a transferência das mensagens das aplicações.

Durante a fase de transferência de dados, cada pacote enviado é identificado com um número de sequência em um campo de cabeçalho e um número de confirmação (**ACK** *nowledgement*). O número de confirmação serve para o receptor informar ao emissor os pacotes que já recebeu. O emissor providenciará a retransmissão do pacote se não receber uma confirmação dentro de um intervalo de tempo estabelecido (*timeout*).

A finalização de uma conexão TCP, por sua vez, ocorre com uma das partes envolvidas enviando um pacote do tipo **FIN**, ou seja, com a *flag* de cabeçalho **FIN** setada, e normalmente com confirmação (**ACK**) do outro lado da conexão, em ambos as direções da conexão.

Continuando a discussão da Figura 4.1, quanto ao cabeçalho TCP, os campos "Porta de origem" e "Porta de Destino" possuem tamanho de 16 *bits*, o que significa que existem 65.536 (0 a 65.535) portas. Os campos "Número de sequência" e "Número de confirmação" são usados para indicar a ordem do

pacote que está sendo enviado e o último pacote recebido, respectivamente. Estes campos possuem tamanho de 32 *bits* cada. O campo "*flags*" (seis *bits*) possui um *bit* para cada *flag*, que é setado (1) ou permanece nulo (0) conforme a função usada durante o funcionamento da conexão no TCP. A "Soma de verificação" (ou *checksum*) é o resultado de uma soma especial nos dados dos cabeçalhos e é usada para verificar a integridade do cabeçalho (SCRIMGER, 2001).

4.3.2 O protocolo UDP

O protocolo UDP (*User Datagram Protocol*) é um protocolo simples da camada de transporte. Diferentemente do TCP, o UDP é um protocolo não confiável, sem controle de sequência em que não há garantia de entrega dos pacotes.

Ainda comparando-se ao TCP, o UDP possui um cabeçalho simplificado como pode ser visto na Figura 4.2. O campo "Soma de verificação" tem função semelhante à função no TCP, porém é opcional. A Figura 4.2, possui um resumo dos campos de um datagrama UDP.

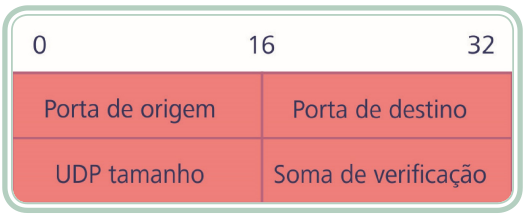

Figura 4.2: UDP
Fonte: CTISM, adaptado de Tanenbaum, 2003, p. 559

Com tantas limitações do UDP é normal nos perguntarmos: Qual a utilidade do UDP, sendo que o TCP faz tudo o que o UDP faz e ainda com confiabilidade?

Apesar da falta de confiabilidade do UDP, ele possui um desempenho melhor que o TCP, pois não há gasto extra (*overhead*) de processamento e de *bits* extras trafegados na rede. Por sua simplicidade o UDP é mais eficiente e rápido. Aplicações em que a confiabilidade na entrega não é tão importante, porém o desempenho é essencial, geralmente, fazem uso do UDP.

Exemplos de aplicação que usa o UDP como protocolo de transporte é o *streaming* de áudio e de vídeo. Nessas aplicações a falta de alguns dados durante a transmissão prejudica apenas a qualidade da imagem ou do áudio quando recebido, sem afetar completamente a transmissão. Na transmissão de áudio ou vídeo em tempo real, a agilidade na entrega dos dados é geralmente o fator mais importante. Outras aplicações podem fazer uso do UDP, por

razões de desempenho e tratar dos possíveis erros de transmissão diretamente dentro da aplicação. No Quadro 4.3 é possível observar algumas das principais características (comparativo) de cada um dos protocolos.

Quadro 4.3: Diferenças entre os protocolos TCP e UDP

TCP	UDP
Orientado a conexão	Não orientado a conexão
Ponto a ponto	Ponto a ponto
Confiável, controle de erros	Não confiável, sem controle de erros
Full duplex	*Full duplex*
Entrega ordenada	Não garante entrega ordenada
Controle de fluxo	Sem mecanismo de controle de fluxo

Fonte: Tanenbaum, 2003

O TCP e o UDP usam o protocolo IP, da camada de rede (internet) para a entrega dos pacotes. Os pacotes TCP ou os datagramas do UDP são encapsulados em datagramas IP e encaminhados (roteados) da origem até o destino. Após o encapsulamento, os roteadores usam basicamente os campos do IP.

Vale ressaltar que o protocolo UDP possibilita além da comunicação ponto-a-ponto, realizar a comunicação de um para muitos, o que significa que um computador origem através do protocolo UDP pode entregar pacotes para diversos computadores destino em uma rede. Este é um diferencial bastante relevante do protocolo UDP.