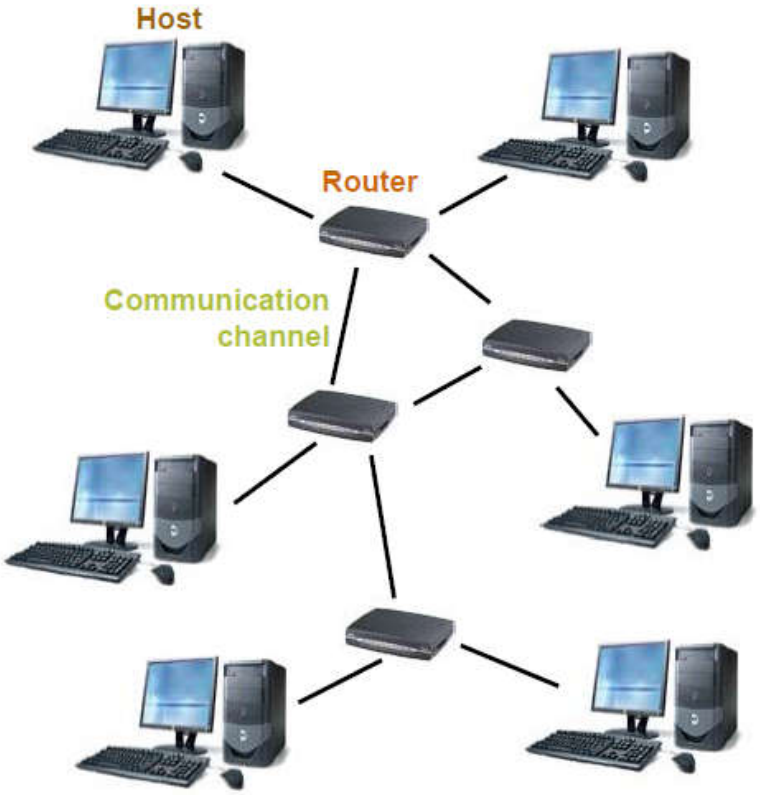
Pregled arhitekture TCP/IP i SW/HW podrška

•Osnovni pojmovi

•Computer Network
  • hosts, router, communication channes
•Hosts run applications
• Routers forward information
• Packets: sequence of bytes
  • contain control information
  • e.g. destination host
• Protocolis an agreement
  • meaning of packets
  • structure and size of packets
•e.g. Hypertext Transfer Protocol (HTTP)



TCP/IP slojevi

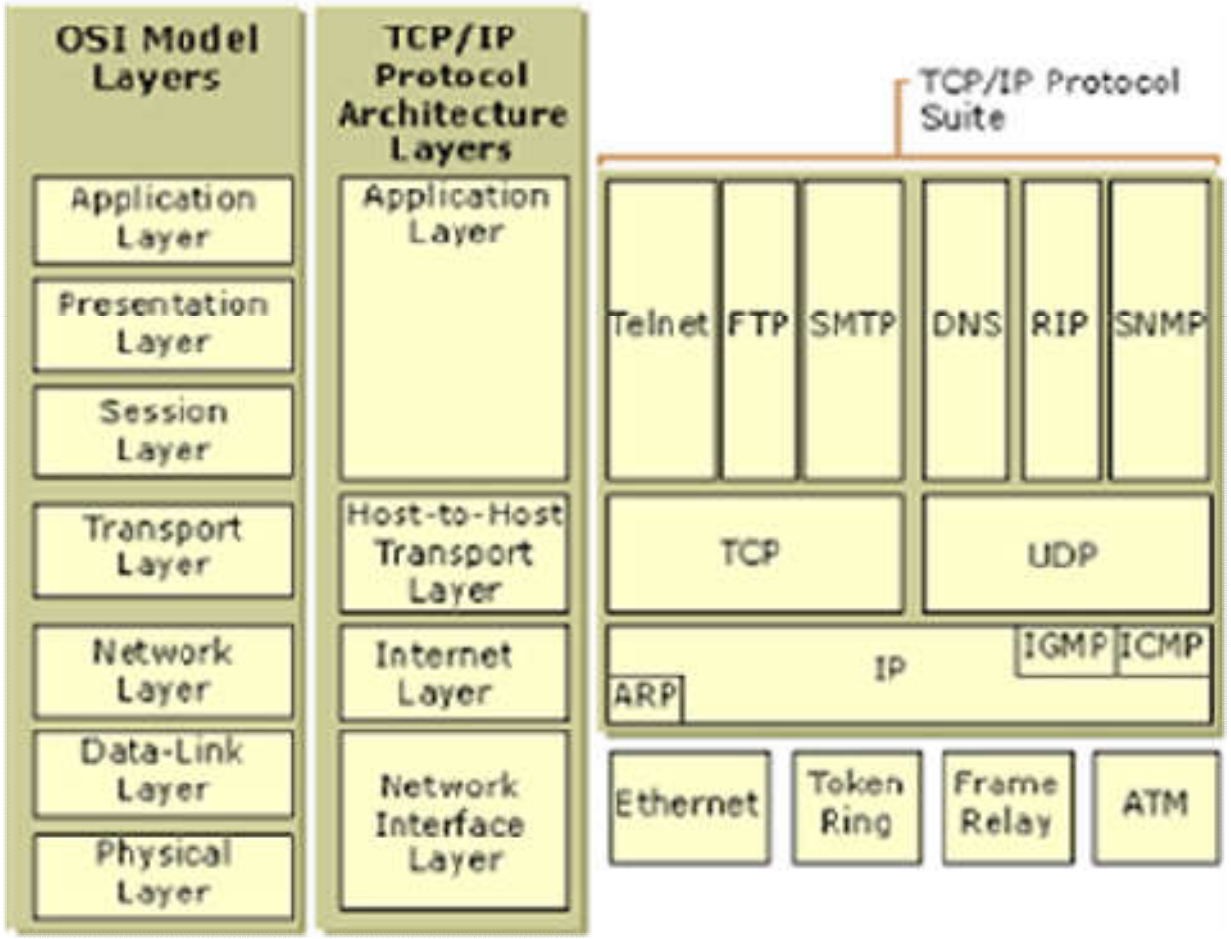
• TCP / IP protokoli mapiraju se u četvorostruki konceptualni model poznat kao DARPA model (Defense Advanced Research Projects Agency), nazvan po vladinoj agenciji SAD koja je inicijalno razvila TCP / IP.

• Četiri sloja modela DARPA su:

- aplikacioni,
- transportni
- Internet
- mrežni interfejs.

• Svaki sloj u DARPA modelu odgovara jednom ili više slojeva sedmostrukog modela interkonekcije otvorenih sistema (OSI).

TCP/IP slojevi



Nivo mrežnog interfejsa

- Nivo mrežnog interfejsa (takođe nazvan i sloj mrežnog pristupa) odgovoran je za postavljanje TCP / IP paketa na mrežni medijum i prijem TCP / IP paketa sa mrežnog medijuma. TCP / IP je dizajniran da bude nezavisan od metoda pristupa mreži, formata frejma i medijuma. Na ovaj način, TCP / IP se može koristiti za povezivanje različitih tipova mreža. Ovo uključuje LAN tehnologije kao što su Ethernet i Token Ring i WAN tehnologije kao što su X.25 i Frame Relay . Nezavisnost od bilo koje specifične mrežne tehnologije daje TCP / IP mogućnost prilagođavanja novim tehnologijama kao što su asinhroni transfer režim (ATM)

Nivo mrežnog interfejsa

- Nivo mrežnog interfejsa obuhvata Data Link i Fizičke slojeve OSI modela.
- Internet sloj ne koristi usluge sekvence i potvrde koji mogu biti prisutni na sloju Data Link.
- Pretpostavljen je nepouzdan sloj mrežnog interfejsa, a pouzdana komunikacija kroz uspostavljanje sesije, sekvenciranje i potvrđivanje paketa je odgovornost transportnog sloja

Internet sloj

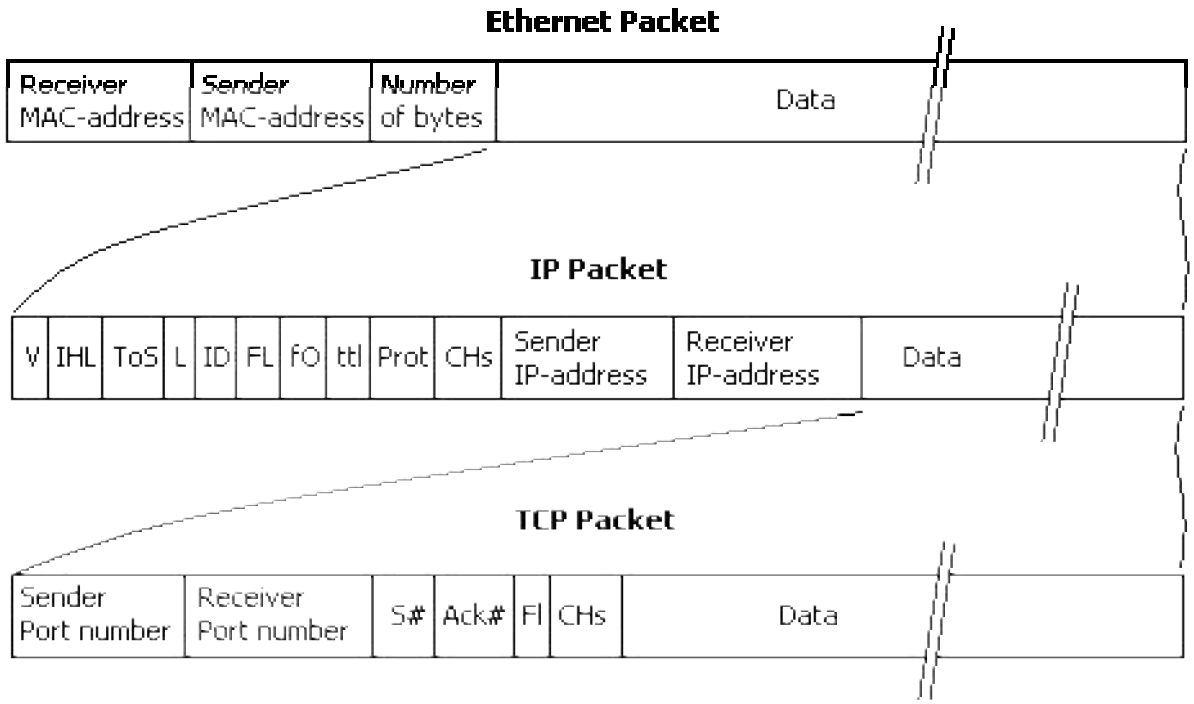
- Internet sloj je odgovoran za funkcije adresiranja, pakovanja i rutiranja. Osnovni protokoli Internet sloja su
- IP,
- ARP,
- ICMP i
- IGMP
- Internet sloj je analogan mrežnom sloju OSI modela

Internet sloj

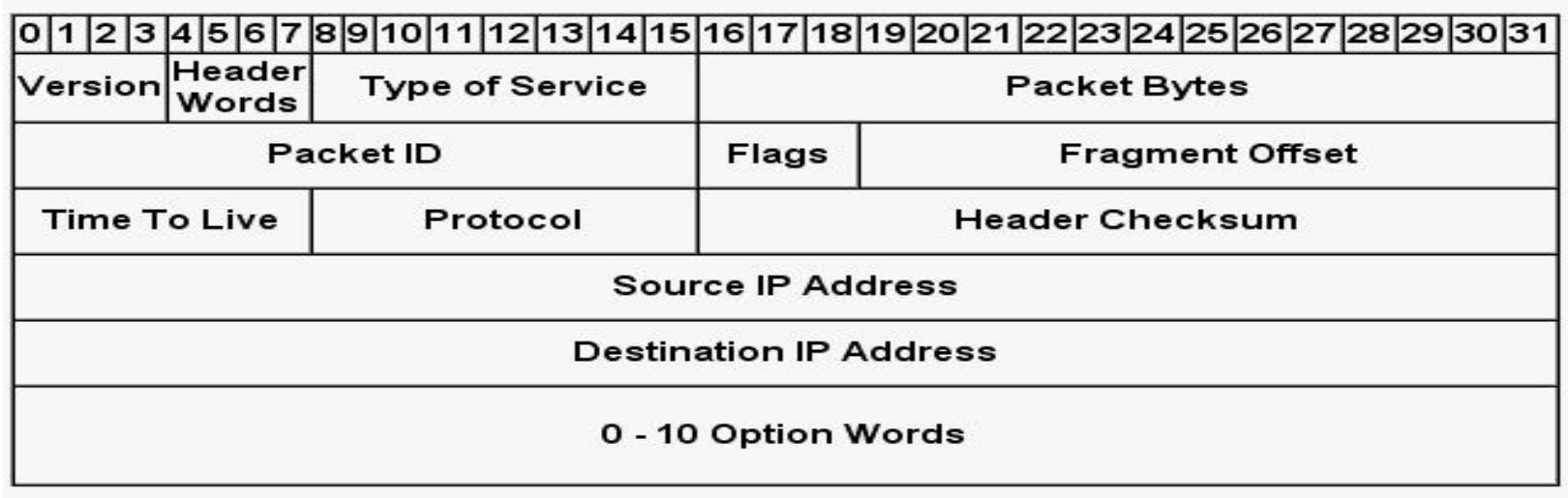
- The *Internet Protocol* (IP) is a routable protocol responsible for IP addressing, routing, and the fragmentation and reassembly of packets.
- The *Address Resolution Protocol* (ARP) is responsible for the resolution of the Internet layer address to the Network Interface layer address such as a hardware address.
- The *Internet Control Message Protocol* (ICMP) is responsible for providing diagnostic functions and reporting errors due to the unsuccessful delivery of IP packets.
- The *Internet Group Management Protocol* (IGMP) is responsible for the management of IP multicast groups.

IP paket

- IP layer se definiše iznad ehternet layer-a, i osnovni cilj uvođenja ovog layera je mogućnost ostvarivanja komunikacije među računarima koji se nalaze u različitim mrežama



IP paket

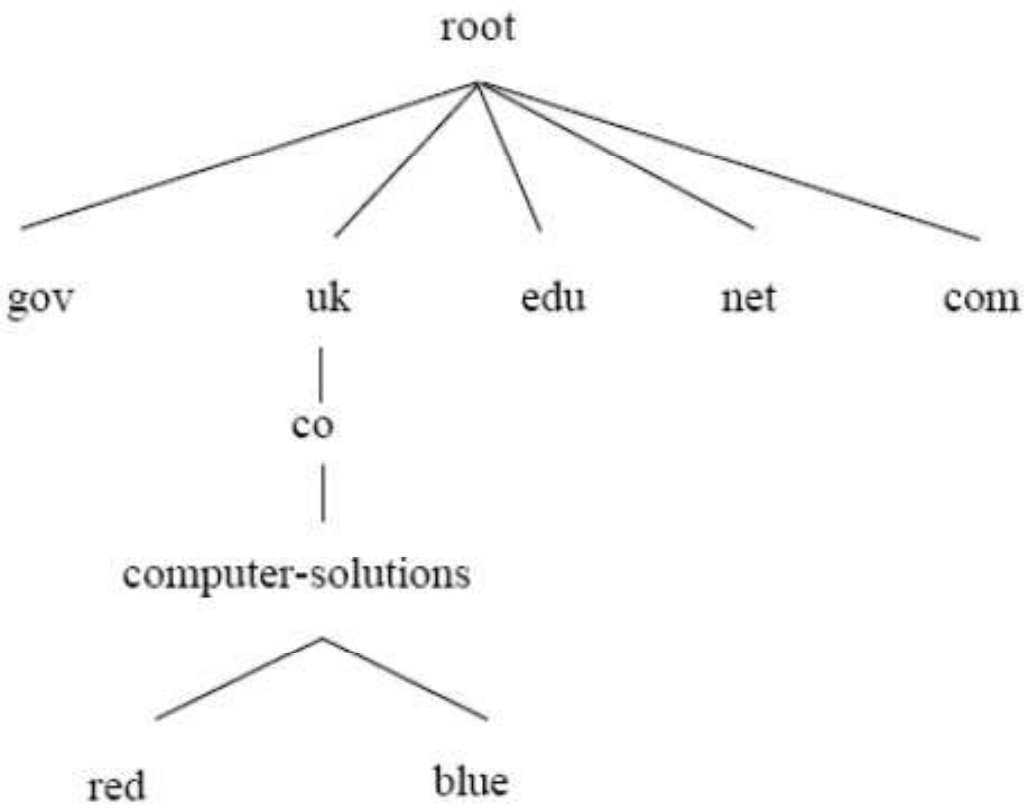


- Version 4 bits određuje koja verzija IP je u pitanju (uglavnom je V4)
- Header words 4 bits određuje veličinu IP hedera kao broj 32 bitnih reči
- Type of Service 8 bits, specificira tip servisa. Određuje zahtevani kvalitet infrastrukture za prenos IP paketa
- Total Length 16 bits, veličina IP paketa u bajtovima, uključujući heder i bajtove podataka
- Identification 16 bits, pomoć primaocu prilikom grupisanja paketa (fragmenata) u celinu
- Flags 3 bits, kontrolni bitovi odnose se na fragmentaciju
- Fragment Offset 13 bits indikacija gde se u celini ovaj fragment nalazi
- Time to Live 8 bits, maksimalno vreme koje paket može ostati u internet sistemu, posle toga se uništava i odbacuje
- Protocol 8 bits, definiše koji se protokol dalje koristi u IP paketu (kao viši layer)
- Header Checksum 16 bits, CRC hedera (detektuje ukoliko postoji greška u hederu IP paketa)

IP Address and Hostnames

- In order for an application to exchange data with a remote process, it must have several pieces of information.
- The first is the IP address of the MCU that the remote program is running on.
- Although this address is internally represented bya 32-bit number, it is typically expressed in either **dot-notation** or by a **logical name** called a ***hostname***.
- Hostnames are divided into several pieces separated by periods, called *domains*. Domains are hierarchical; with the toplevel domains defining the type of organization that network belongs to, with subdomains further identifying the specific network

IP Address and Hostnames



tnt.etf.rs
elektronika.etf.bg.ac.rs
147.91.14.199 nadji adrese!

IP Address and Hostnames

- In order to use a hostname instead of a dot-address to identify a specific system or network, there must be some correlation between the two. This is accomplished by one of two means:
  - A local host table (text array that lists the IP address of a host, followed by the names that it's known by.
  - A name Server, a program running somewhere on a network which can be presented with a hostname and which will return that host's IP address.

Transportni sloj

- Transportni sloj (poznat i kao sloj host-to-host transporta) je odgovoran za obezbeđivanje sloja aplikacije sa komunikacijskim servisima sesije i datagrama. Osnovni protokoli transportnog sloja su Protokol kontrole prenosa (TCP) i Protokol User Datagram (UDP). Transportni sloj obuhvata odgovornosti sloja OSI transporta i neke od odgovornosti OSI Session sloja

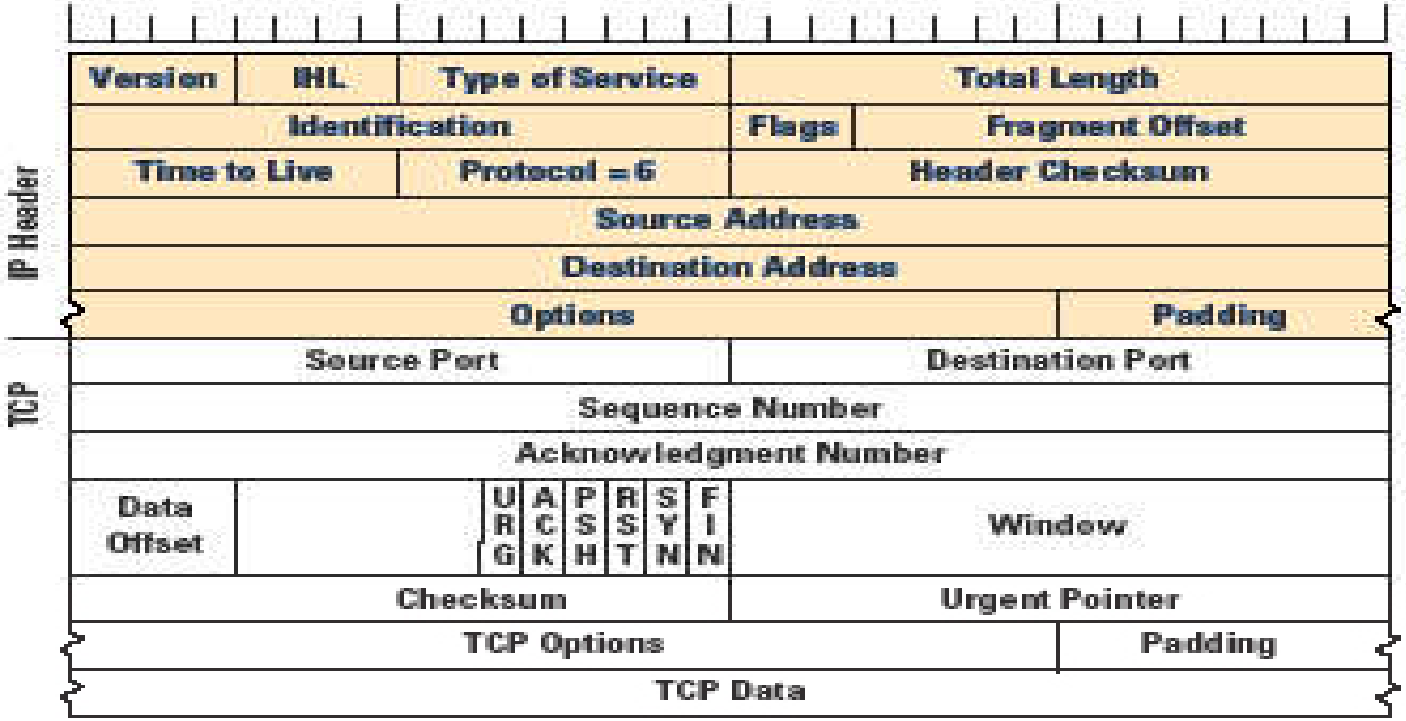
Transportni sloj

- TCP provides a one-to-one, connection-oriented, reliable communications service. TCP is responsible for the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.
- UDP provides a one-to-one or one-to-many, connectionless, unreliable communications service. UDP is used when the amount of data to be transferred is small (such as the data that would fit into a single packet), when the overhead of establishing a TCP connection is not desired or when the applications or upper layer protocols provide reliable delivery.

TCP-IP, UDP-IP

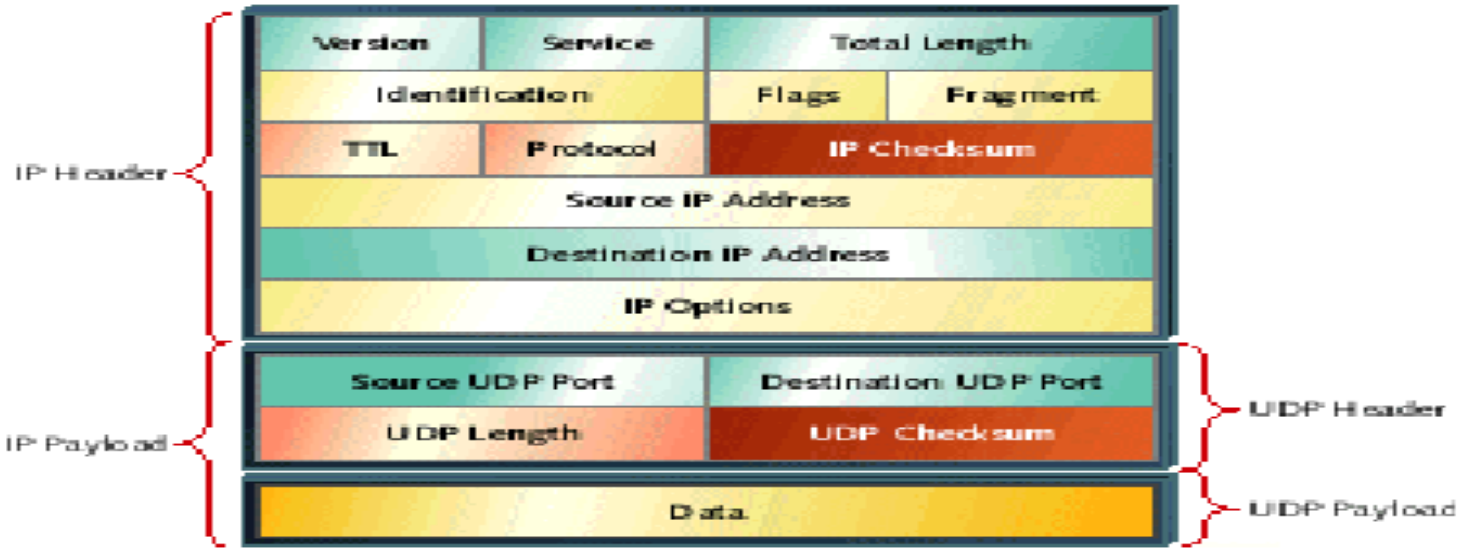
- Protokoli definisani na layeru iznad IP
- TCP najkorišćeniji protokol u svetu interneta. Detektuje izgubljene pakete i zahteva njihovo ponovno slanje. Detektuje pakete sa obrnutim redosledom pristizanja od poslatog i ispravlja inverziju. Za svaki pristigli paket šalje se ACK paket koji pošiljaocu signalizira ispravan prijem. Ukoliko ne dobije ACK paket, pošiljalac opet šalje izgubljeni paket.
- UDP protokol nema podršku za detekciju izgubljenih paketa ili onih koji su stigli mimo predviđenog redosleda. Nije pouzdan kao TCP ali zato efektivno ne smanjuje propusni opseg. Koristi se za strimovanje multimedajlih sadržaja.

TCP-IP



- Source Port i Destination Port fields (16 bits) određuju izvorišni i odredišni port
- Sequence Number (32 bits), određuje redni broj paketa, ovaj broj se korisi za određivanje inverzije u redosledu pristizanja paketa
- Acknowledgment number (32 bits), kada dobije paket, primalac odgovara sa ACK paketom koji u ovom polju sadrži sequence number narednog paketa koji očekuje da primi
- Data offset (4 bits) specificira veličinu TCP hedera u 32b rečima
- Flags (8 bits) biti koji se koriste prilikom sinhronizacije

UDP-IP



- Source Port i Destination Port fields (16 bits) određuju izvorišni i odredišni port
- Length (16 bits), određuje veličinu celog UDP paketa, heder+podaci
- Checksum (16 bits), CRC za heder+data

Aplikacioni sloj

- Aplikacioni sloj pruža aplikacijama mogućnost pristupa uslugama drugih slojeva i definiše protokole koje aplikacije koriste za razmjenu podataka.
- Postoji mnogo aplikacionih protokola
- I dalje se razvijaju novi protokoli.
- Naj poznatiji protokoli aplikativnog sloja su oni koji se koriste za razmenu korisničkih podataka

Aplikacioni sloj

- Protokol hipertekstnog prenosa (HTTP) se koristi za prenošenje datoteka koje čine Web stranice World Wide Weba.
  • File Transfer Protocol (FTP) se koristi za interaktivni prenos datoteka.
  • Simple Mail Transfer Protocol (SMTP) se koristi za prenos poštanskih poruka i priloga.
  • Telnet, protokol emulacije terminala, se koristi za konekciju na mrežne hostove.

Aplikacioni sloj

Dodatno, sledeći protokoli aplikativnog sloja pomažu u korištenju i upravljanju TCP / IP mrežama:

- Sistem imena domena (DNS) se koristi za razrešavanje imena hosta na IP adresu.
- Protokol RIP-a (RIP) je protokol rutiranja koji ruteri koriste za razmenu informacija o rutiranju na IP mreži.
- Jednostavni protokol za mrežni menadžment (SNMP) koristi se između konzole za upravljanje mrežom i mrežnih uređaja (rutera, mostova, inteligentnih čvorišta) za sakupljanje i razmenu informacija o upravljanju mrežom.

Aplikacioni sloj

- Primeri interfejsa prema aplikacionom sloju za TCP / IP aplikacije su Windows Sockets i NetBIOS

TCP vs UDP

- TCP is used for services with a large data capacity, and a persistent connection
- UDP is more commonly used for quick lookups, and single use query-reply actions.

- Some common examples of TCP and UDP with their default ports:

DNS lookup	UDP	53
FTP	TCP	21
HTTP	TCP	80
POP3	TCP	110
Telnet	TCP	23

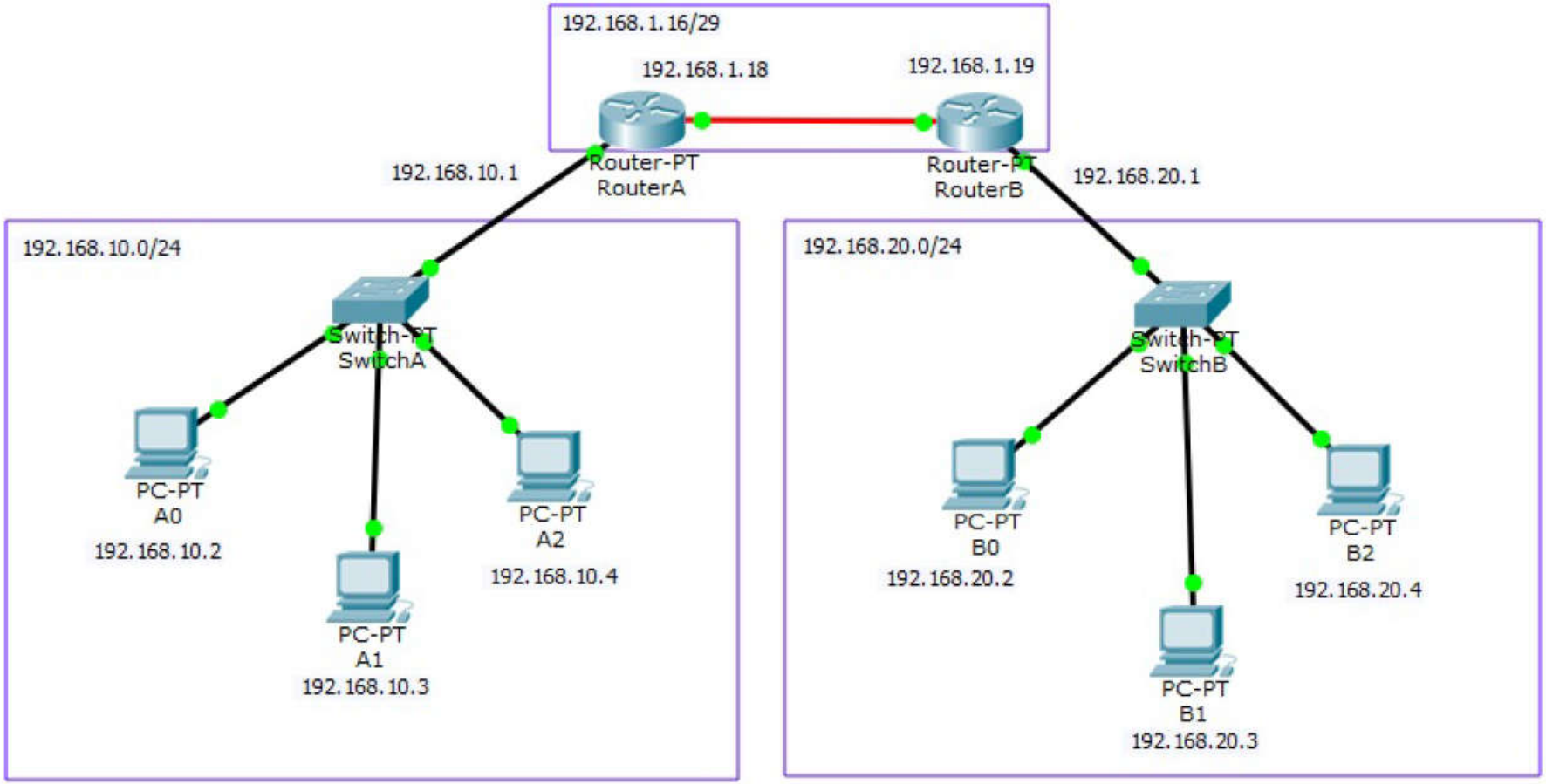
Service Ports **

• In addition to the IP address of the remote CPU/MPU, an application also needs to know how to address the specific program on the CPU/MPU that it wishes to communicate with.
• This is because large processors running TCP/IP will typically be multi-tasked and running a number of different links.
• This is accomplished by specifying a *service port*, a 16-bit number that uniquely identifies an application running on the CPU/MPU.

•A number of standard service ports and names are used by Internet-based applications and these are referred to as *well-known services*.
• These services are defined by a standards document and include common application protocols such as FTP, POP3, SMTP and HTTP.
• Port numbers 1 – 1023 are reserved for well-known services.

Service Ports **

•**Registered Port (Ports 1024-49151)**-These can be registered for predefined services and should be treated as **semi-reserved.** User written programs should not use these ports.
•**Ports 49152-65535**– These are used by **client programs** and you are free to use these in client programs. When a Web browser connects to a web server the browser will allocate itself a port in this range. Also known as **ephemeral ports**.
• When setting up an application specific link (say a straight TCP link between a micro and a PC) you should avoid using the well-known services range.



A2	192.168.10.4	192.168.10.1
B0	192.168.20.2	192.168.20.1

Gateway Adresa

Gateway predstavlja adresu mrežnog uređaja koji omogućuje ostalim uređajima u LAN-u da komuniciraju sa uređajima van LANa. Ova adresa se podešava na svakom mrežnom uređaju posebno i ista je za sve uređaje u jednom LAN-u. Bilo koji IP paket sa IP adresom koja ne pripada opsegu IP adresa u LAN-u u kojem se nalazi računar koji šalje taj IP paket, se šalje do uređaja sa gateway adresom.

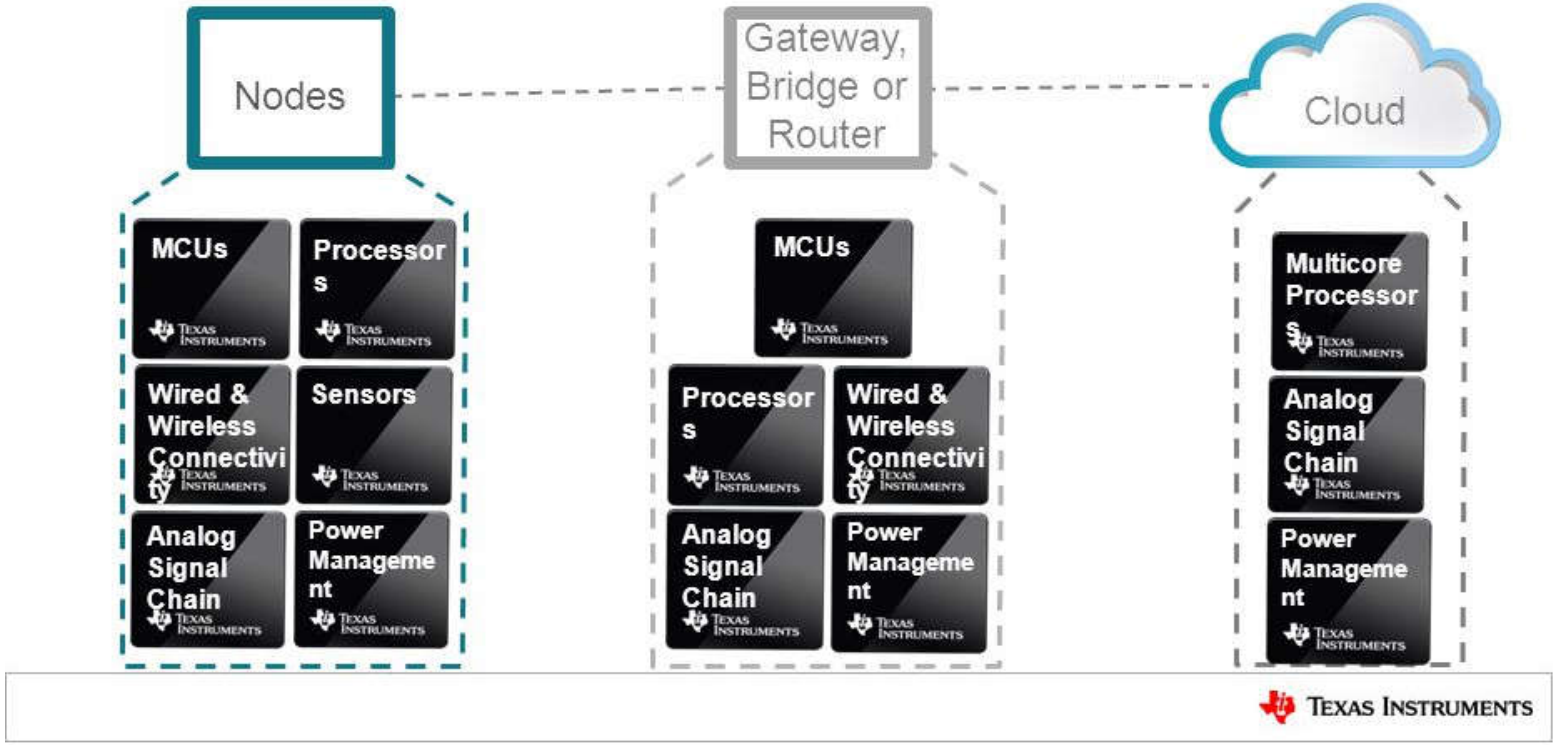
Mikrokontroleri i TCP-IP

Ethernet or other multi drop protocols

As Ethernet cards have become standard on PC's their price has fallen dramatically. The availability of Ethernet chips that are easily interfaced to both 8 and 16 bit micros, at less than $10 each, along with the ease and familiarity of using networked PC's will make this an increasingly popular communication media for industrial systems.

It is also clear that the software components required to control an Ethernet network could be easily adapted to operate multi station Transport Layers such as RS485, CAN and other proprietary multi drop links by the provision of appropriate hardware drivers.

Only TI has all the IoT building blocks



Drivers and Stacks

The need for something more complex than a conventional driver arises from the nature of communications networks - multiple interactions is going on at the same time.

Stack is extension of a device driver concept

A Stack is a set of co-operating programs written to work together in many different combinations (~*inheritance*).

Stack underlying structure is provided by a commonly agreed set of protocols (or message standards)

Each level of the stack hiding the messy detail of the level below as we become more and more application oriented.

Network API and sockets

- Operating system provides Application Programming Interface (API) for network application
- API is defined by a set of function types, data structures, and constants
- Desirable characteristics of the network interface
  - Simple to use
  - Flexible
    - independent from any application
    - allows program to use all functionality of the network
  - Standardized
    - allows programmer to learn once, write anywhere
- Application Programming Interface for networks is called socket

Sockets **

• A ***socket*** is a communications end-point
• If you need to establish a connection with the other program, you need the *socket address* of the application that you want to connect to.
• Once a connection has been established to a socket in the addressee the applications programmer need only consider the data to be read/written.

Using the socket interface

```
socket = mn_open(dest_ip, src_port, dest_port,
client, TCP, recv_buff, buff_len);

status = mn_send(socket, msg_ptr, msg_len);
status = mn_recv(socket, buff_ptr, buff_len);
status = mn_close (socket);
```

Blocking and Non-Blocking Sockets

• A *blocking socket*: the program is "blocked" until the request for data has been satisfied. When the remote system does write some data on the socket, the read operation will complete and execution of the program will resume.
• *A non-blocking socket* requires that the application recognize the error condition and handles the situation appropriately.
• The default behavior for socket functions is to "block" and not return until the operation has completed

Client-Server Applications

Programs written to use TCP are developed using the *Client-Server model*.

•The Client application initiates what is called an *active open*. It creates a socket and actively attempts to connect to a Server program.
•The Server application creates a socket and passively listens for incoming connections from Clients, performing what is called a *passive open*.
•When the Client initiates a connection, the Server is notified that some process is attempting to connect with it.
•By *accepting* the connection, the Server completes what is called a *virtual circuit*, a logical communications pathway between the two programs.

Client-Server Applications

• In order to keep overheads low, when MCU accepts a connection to a socket (if the port numbers match) usually it allocates it to the socket that was listening.

•On larger CPUs such as the TCP/IP stack running on a PC, the original socket may remain listening for additional connections and the link may communicate via a new socket.

•When the Server no longer wishes to listen for connections, it closes the original passive socket.

• There are five main steps that a program, must take to establish and complete a connection.

The Server side

•Create a socket.
•Listen for incoming connections from Clients.
•Accept the Client connection.
•Send and receive information.
•Close the socket when the Client has finished or when the Server wishes to no longer be available.

The Client side:

•Create a socket.
•Specify the address and service port of the Server program.
•Establish the connection with the Server.
•Send and receive information.
•Close the socket when finished, terminating the conversation.

Layers

The TCP/IP stack is broken down into layers as shown in the table

Layer	Examples	Function
Client Application	DHCP, FTP, WEB, YOUR APP	Do Something Useful
Transport	TCP, UDP	Send a Message
Internet	IP, PING, ARP	Component parts, Connect, Data, handshake
Link	PPP, SLIP	Customize for & talks to specific hardware
Physical	RS232, Ethernet	The hardware chip voltage, frequency, signaling techniques

Physical Layer

- The traditional used physical connection to a MCU is the RS232 serial connection. For long distances this is converted from its absolute voltage, bit signaling form to a voltage independent, frequency modulated form by a modem.
- For high speed connections between PC's the most common form of connection is Ethernet using CSMA/CD (Carrier Sense, Multiple Access with Collision Detection).
- All these are examples of different Physical Layers and their definitions include voltages, signaling mechanisms and timing details.
- Each different mechanism will have particular characteristics that require data to be specially formatted for it and this is done by the link layer.

Link Layers - SLIP & PPP

•SLIP stands for Serial Line Internet Protocol and is one of the simplest conventions for sending TCP/IP packages along a serial line.

• Typical use: on a low noise RS232 link between two fixed processors.

• Disadvantages:

- ➢Each end assumes it knows the identity of the device at the other end.
- ➢Only a single conversation can go on at one time.
- ➢There is no protection against data corruption at this level and it relies on the levels above to detect transmission errors.

• Advantage: the overhead added by SLIP is minimal.

• PPP stands for Point to Point Protocol and is the link layer protocol most commonly used for TCP/IP package communications over modems.
• The message structure contains a word (called the Frame Check Sequence or FCS.) that provides error detection. It also defines a number of options including a protocol for configuring and testing lines.
•The PPP Client on dialing may be allocated an IP address by the Server. This is necessary as many Servers allocate a different IP address every time you dial them (dynamic IP addressing).
• Password option built into the protocol – Password Authentication Protocol (PAP).

Modems

• Modem functions are usually a part of the core TCP/IP stack and support both Dial in and Dial out functions along with commands to simplify logging into an ISP.
• All necessary Modem functions must be executed prior to the start of a PPP dialogue.

Internet Layer

- Each Ethernet interface card or chip has a 48bit unique physical address called its MAC or OUI.
- The **Internet** Layer includes two protocols that are used to
- perform translations between IP addresses and physical addresses:
  - ARP (Address Resolution Protocol) and
  - RARP (Reverse Address Resolution Protocol)

- If you ask to talk to IP xxx then ARP will look in its table for the MAC that it knows is associated with that IP address.
- If it fails to find a MAC associated with the IP address then a message is sent to all systems on the network asking if they are the IP xxx. If one successfully acknowledges with its MAC then the table is updated and an address returned.
- It is this MAC address that then appears in the Ethernet packet as the destination address.

Ping

- A Not a real protocol, but a useful program - utility to determine whether a specific IP address is accessible.
- It works by sending a packet to the specified address and waiting for a reply.
- PING is used primarily to troubleshoot Internet

File Transfer Protocol (FTP)

This application copies a complete file from one CPU to another – it does not allow one CPU to read individual records at a time from files held on the other CPU (NFS Network File System does that).

Getting an IP address (BOOTP, DHCP & TFTP)

If a device on a network is to successfully communicate with the other devices on the network it must have a name – its IP address. There are many practical reasons why it is better to have the network allocate an IP address rather than have it built into the device ROM or manually set up. As has already been explained that PPP Servers may allocate an IP address whenever a connection is made. When using multi drop links such as Ethernet, the most common ways of allocating IP addresses are BOOTP and Dynamic Host Configuration Protocol (DHCP) Servers. For either of these a Server running on the network is required

BOOTP & DHCP

•BOOTP is the simpler protocol, the Client requests an IP address and at the same time can ask for a named file transfer. This file is often used to load the application code into the device.

•DHCP (Dynamic Host Configuration Protocol)is more sophisticated. It can ask for an IP address which will only be valid for a limited time, after which it will have to re-apply. The time can be infinite or if required the device can apply for a time extension. DHCP can also request a file transfer which is done with TFTP.

TFTP
(Trivial File Transfer Protocol)

• In order too **minimize the code space** required in the embedded device before an initial file transfer is made, these protocols can use the simpler TFTP Client

• TFTP does not have any password handling and uses UDP rather than TCP for the data transfers.

• It optimizes transfer times but does implement a handshake to overcome UDP's inherent unreliability.

Web Server (HTTP)

• The Web Server is significantly smaller than the rest of the TCP/IP stack.
• When used with a windows browser such as Explorer, it adds significant capabilities to the system.
• The Web Server usually includes the virtual file system that holds the web pages to be requested/browsed.
• By linking a PC running Windows and a web browser to a micro running the Web Server we immediately have a **sophisticated color GUI whose use will be immediately familiar to a technically literate audience**

Simple Mail Transfer Protocol (SMTP)

Based on TCP this Client application allows a micro to send an email to any Server that supports it (most ISP's will accept SMTP).

Post Office Protocol (POP3)

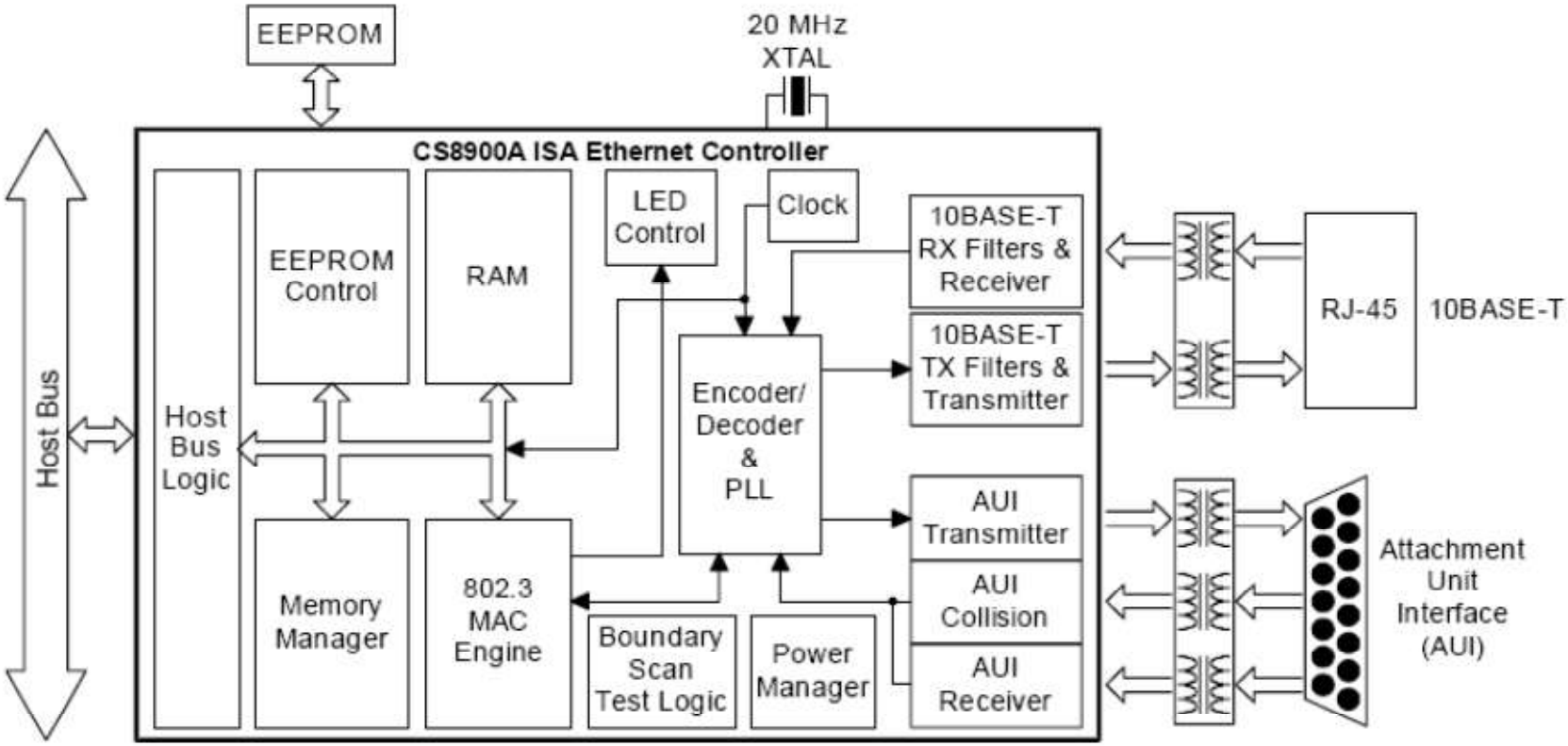
POP3 Client is the most common way of receiving emails and is supported by most ISP's. It allows the Client to request the number of emails waiting for it, their origin, size and the subject line of each message. It then can retrieve selected messages and delete them from the Server.

Simple Network Management Protocol (SNMP)

This protocol based on UDP is used to manage devices attached to a network. It is typically used for devices such as routers and Servers. It provides network managers with the ability to access and change network settings as well as to be informed when specific events occur.

Hardware support for Internet

Ethernet support chips such is the Cirrus Crystal CS8900 Ethernet controller can be easily interfaced to embedded processors with substantial transmit and receive buffers that take much of the load of running Ethernet off the processor.



Problems:

- High volume of broadcast packets floating around a loaded network
- Malicious attacks

Solutions:

- Bigger processor powerful enough to allow a suitable bandwidth. This is the case with Embedded Linux devices such as x86/x64 or ARM based devices.
- Hardware TCP/IP stack

XPort Embedded Device Server

- Ethernet 10Base-T or 100Base-TX RJ45
- Supports TCP/IP, UDP/IP, SNMP, TFTP, DHCP, BOOTP, HTTP...
- 300 to 230 kbps UART
- 3 PIO pins (Software selectable)
- Can serve web pages and Java applets with 384Kbytes internal Flash Storage
- 256Kbytes SRAM and 512Kbytes flash

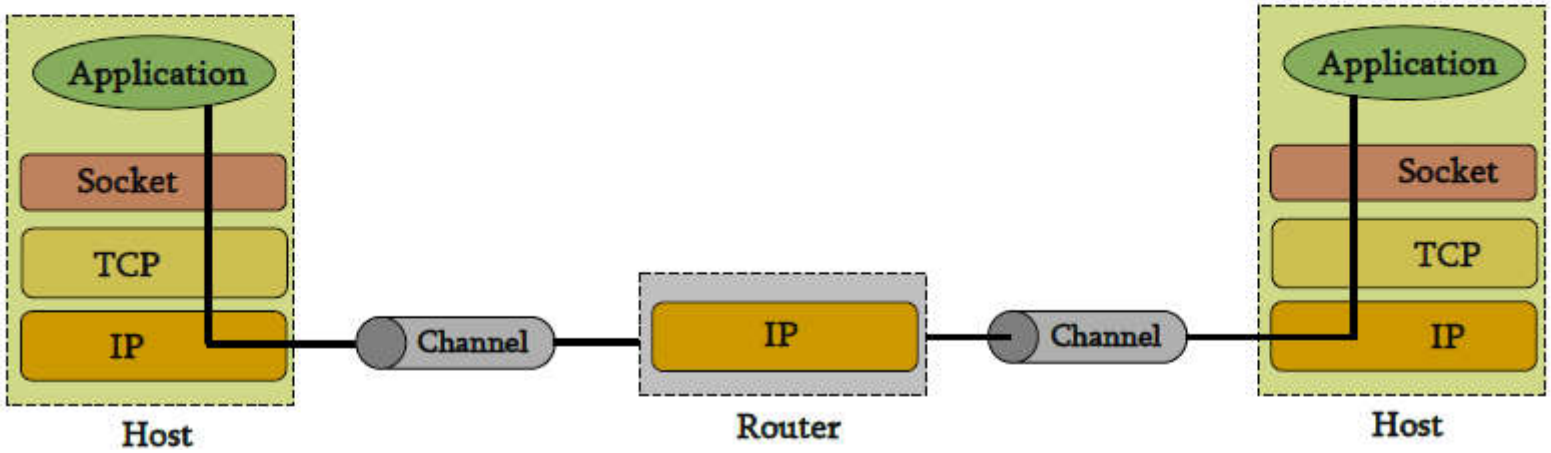
WiPort Wireless Embedded Device Server

- Wireless 802.11b and 10/100 Ethernet
- Supports TCP/IP, UDP/IP, SNMP, TFTP, DHCP, BOOTP, HTTP, ...
- Dual 300 to 921.6 kbps UART
- 11 GPIO pins
- Can serve web pages and Java applets with 1.8MBytes (or 3.8MB) internal Flash 256Kbytes zero wait state SRAM, 1024KB SRAM and 2048KB (or 4096) flash

Socket programming basics

Berkley Sockets

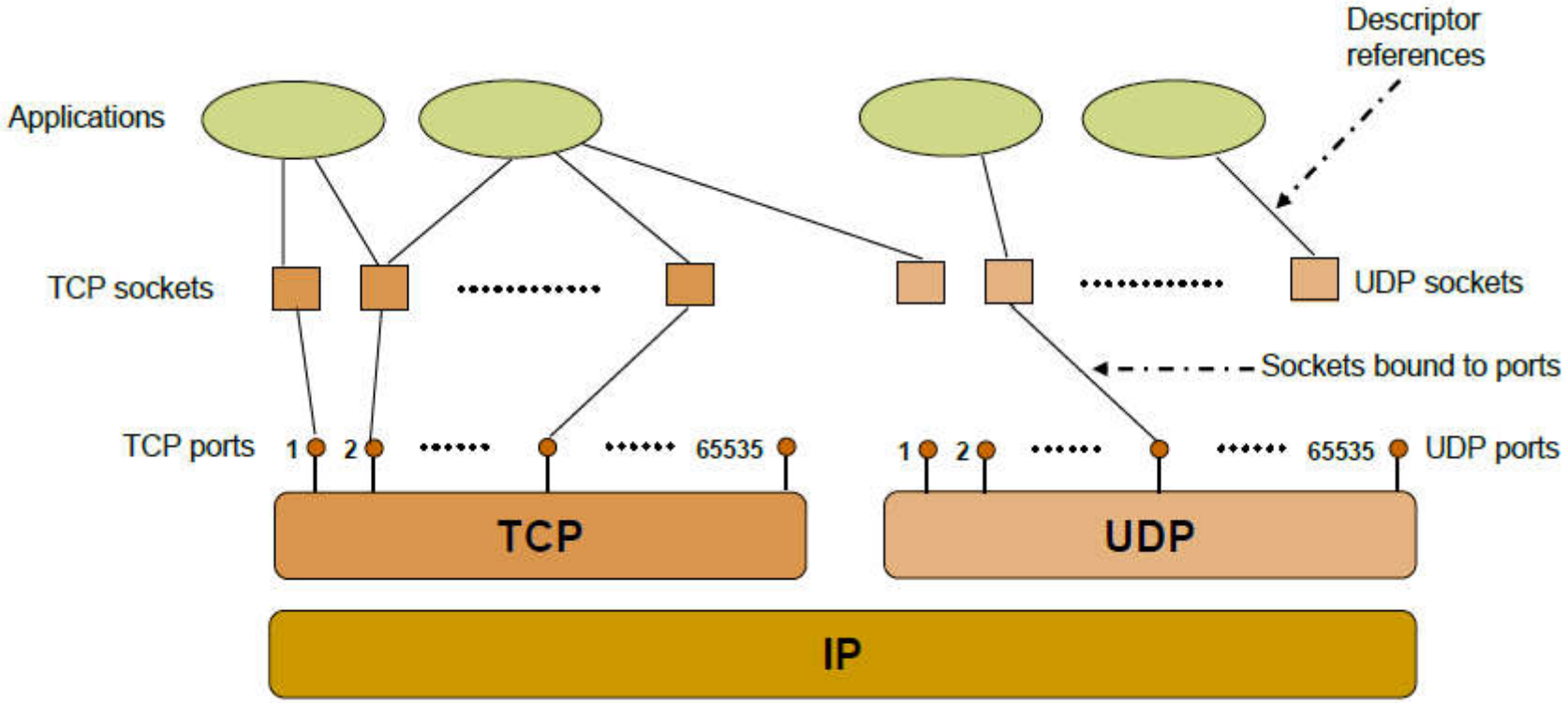
- Universally known as **Sockets**
- It is an abstraction through which an application may send and receive data
- Provide **generic access** to interprocess communication services
  - e.g. IPX/SPX, Appletalk, TCP/IP
- Standard API for networking



Sockets

- Uniquely identified by
  - an internet address
  - an end-to-end protocol (e.g. TCP or UDP)
  - a port number
- Two types of (TCP/IP) sockets
  - Stream sockets (e.g. uses TCP)
    - provide reliable byte-stream service
  - Datagram sockets (e.g. uses UDP)
    - provide best-effort datagram service
    - messages up to 65.500 bytes
- Socket extend the convectional UNIX I/O facilities
  - file descriptors for network communication
  - extended the read and write system calls

Sockets

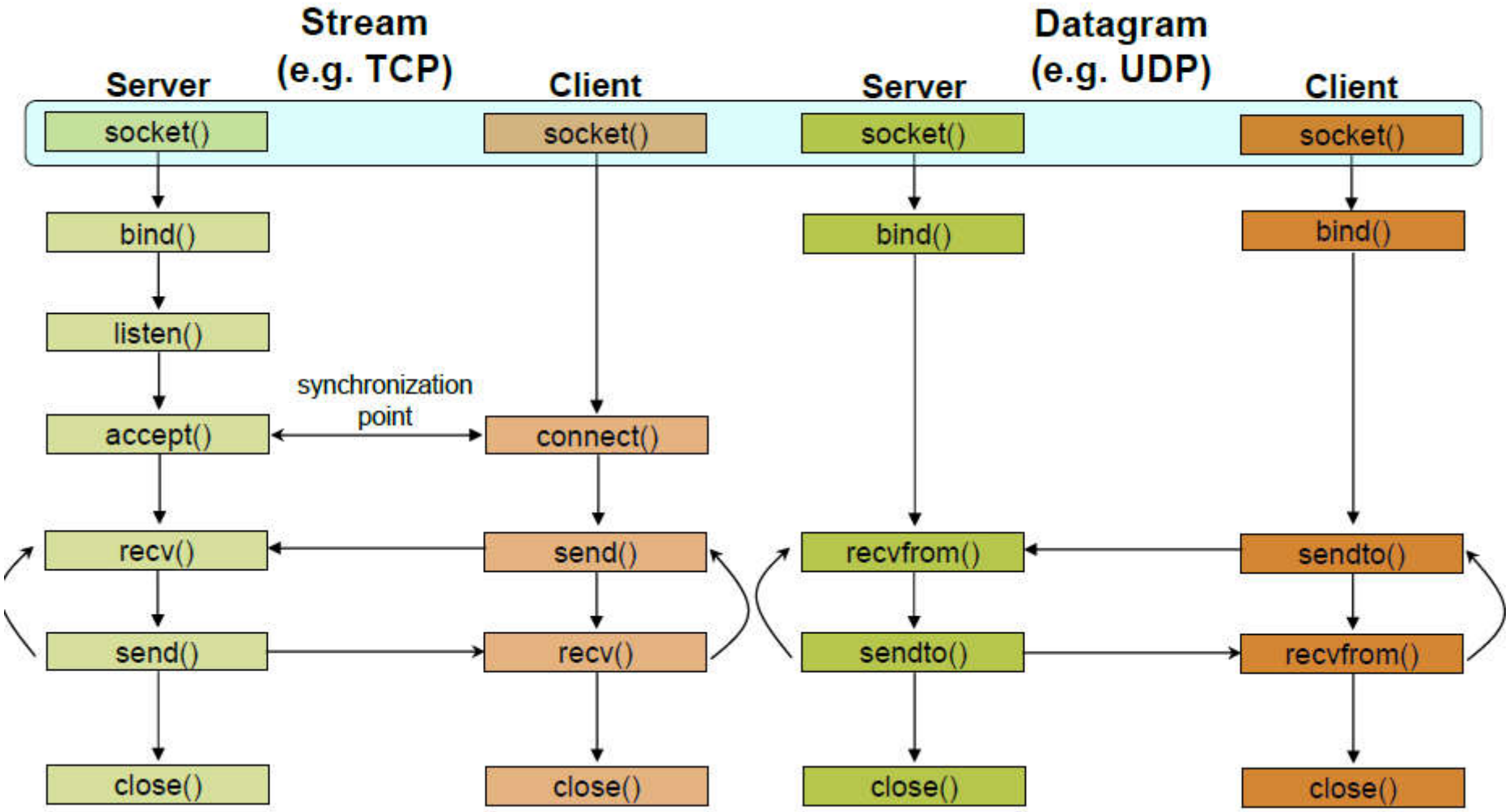

Descriptor
references
Applications
TCP sockets
UDP sockets
Sockets bound to ports
TCP ports
1
2
65535
1
2
65535
UDP ports
TCP
UDP
IP

Client-Server communication

- **Server**
  - passively waits for and responds to clients
  - **passive** socket
- **Client**
  - initiates the communication
  - must know the address and the port of the server
  - **active** socket

Sockets - Procedures

Primitive	Meaning
Socket	Create a new communication endpoint
Bind	Attach a local address to a socket
Listen	Announce willingness to accept connections
Accept	Block caller until a connection request arrives
Connect	Actively attempt to establish a connection
Send	Send some data over the connection
Receive	Receive some data over the connection
Close	Release the connection


Stream
(e.g. TCP)
Server
Client
socket()
socket()
bind()
listen()
accept()
synchronization
point
connect()
recv()
send()
send()
recv()
close()
close()
Datagram
(e.g. UDP)
Server
Client
socket()
socket()
bind()
bind()
recvfrom()
sendto()
sendto()
recvfrom()
close()
close()