

Nagyteljesítményű mikrovezérlők TCP/IP

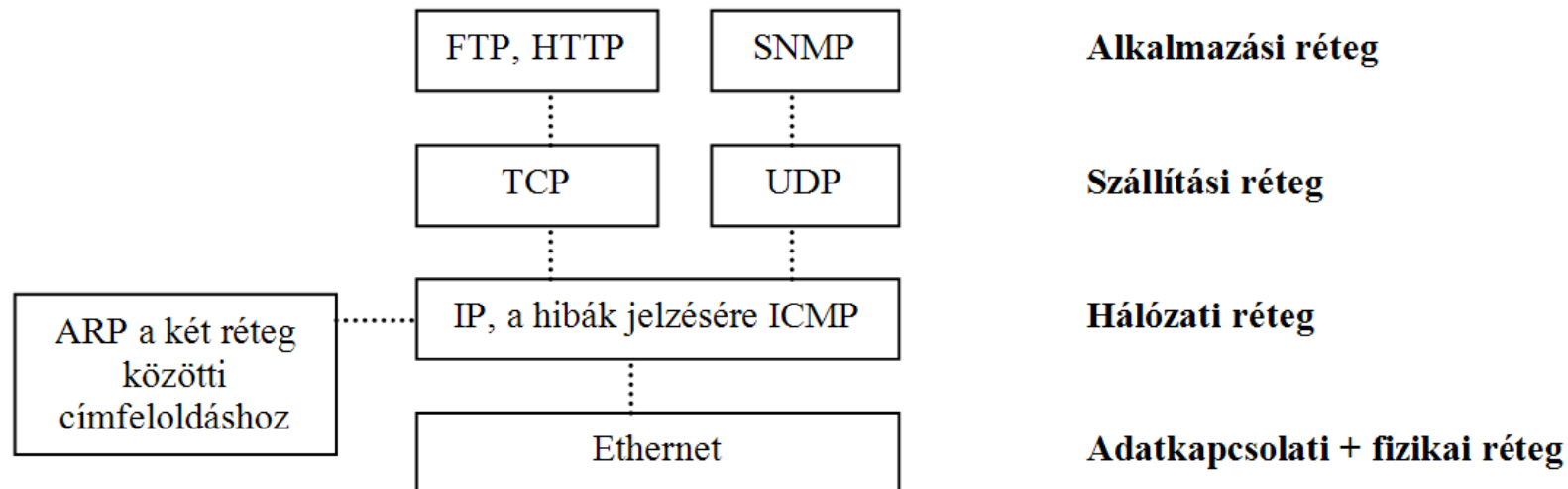
Scherer Balázs



Méréstechnika és
Információs Rendszerek
Tanszék

Alap beágyazott szoftver architektúrák I.

■ Szükséges TCP/IP protokollok



IP : Internet Protocol

ICMP : Internet Control Message Protocol

ARP : Address Resolution Protocol

TCP : Transmission Control Protocol

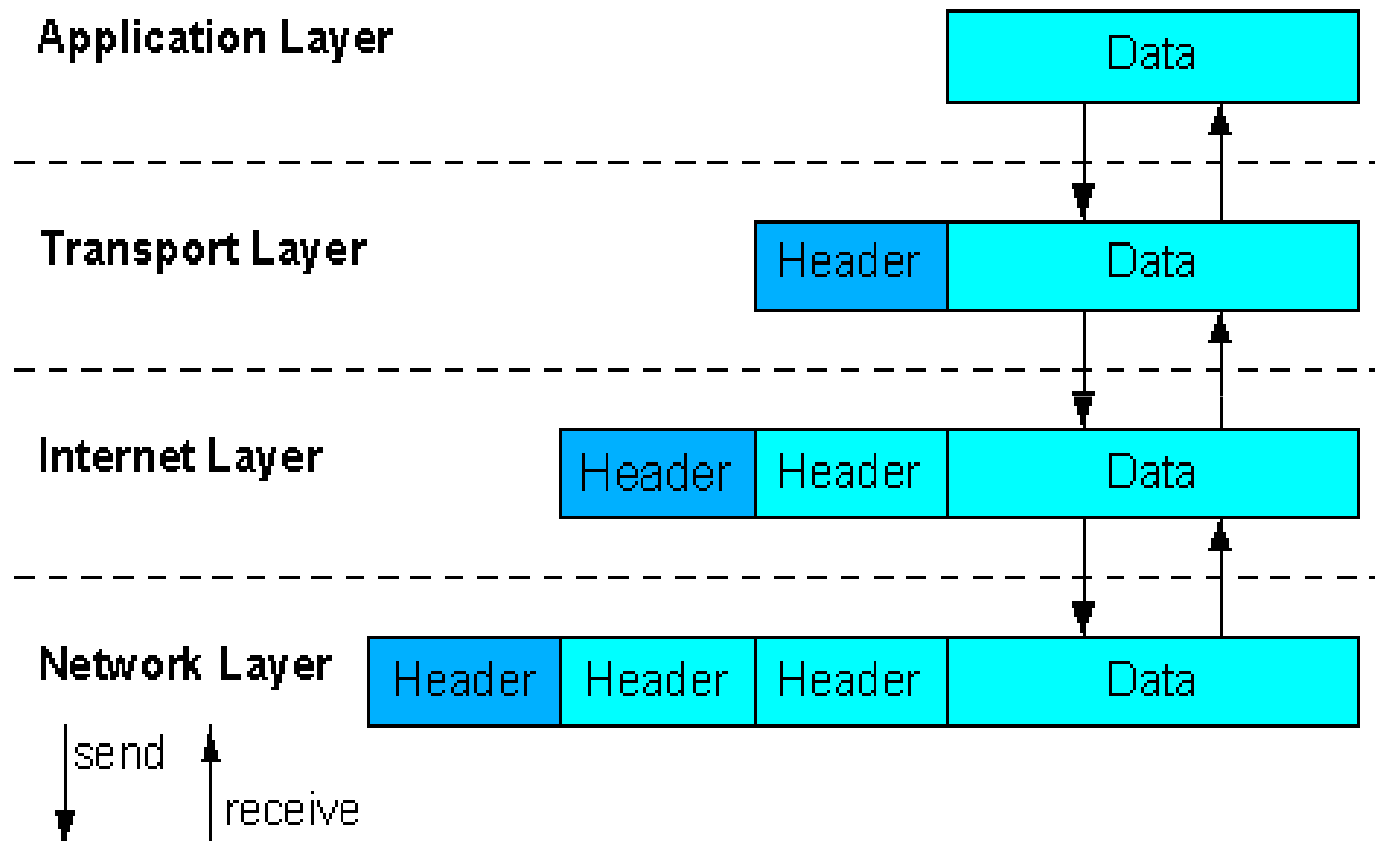
UDP : User Datagram Protocol

FTP : File Transfer Protocol

HTTP : Hyper Text Transfer Protocol

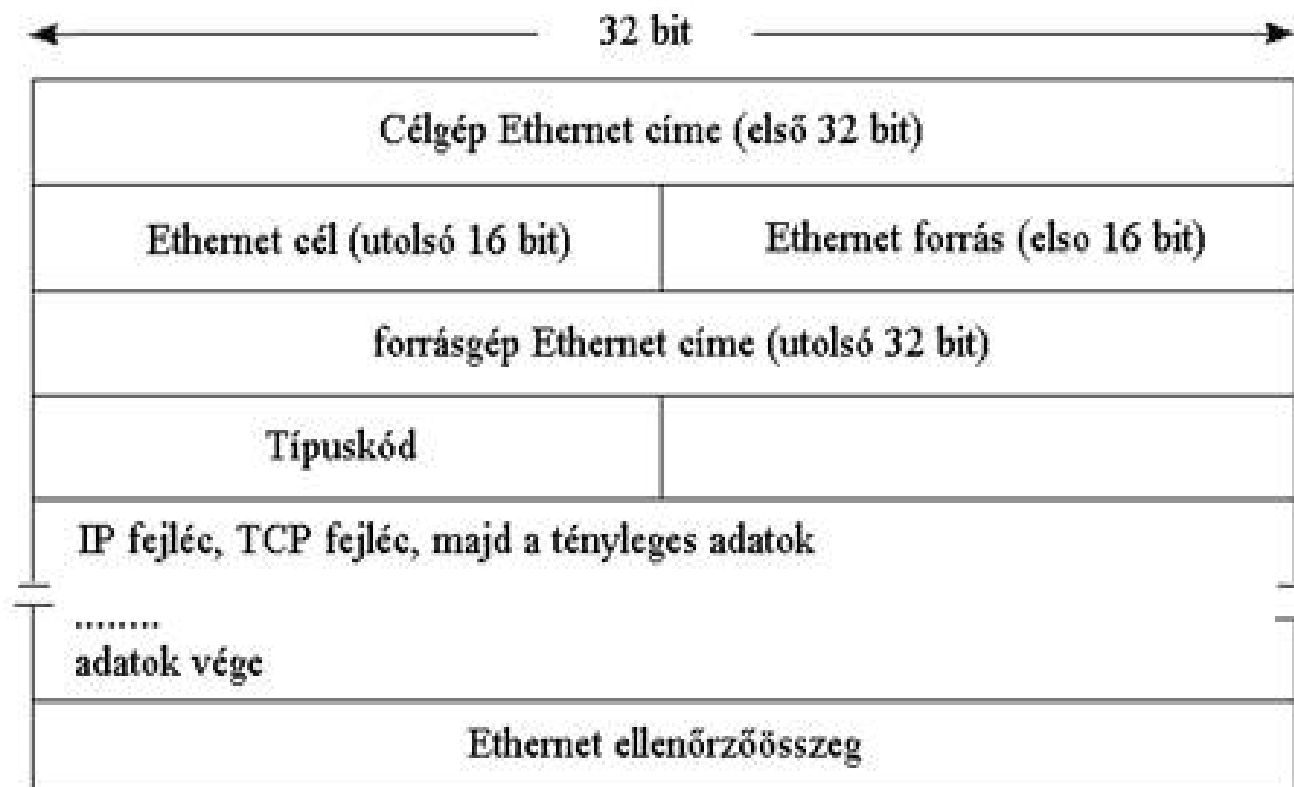
SNMP : Simple Network Management Protocol

Keretezés a hálózaton



Ethernet

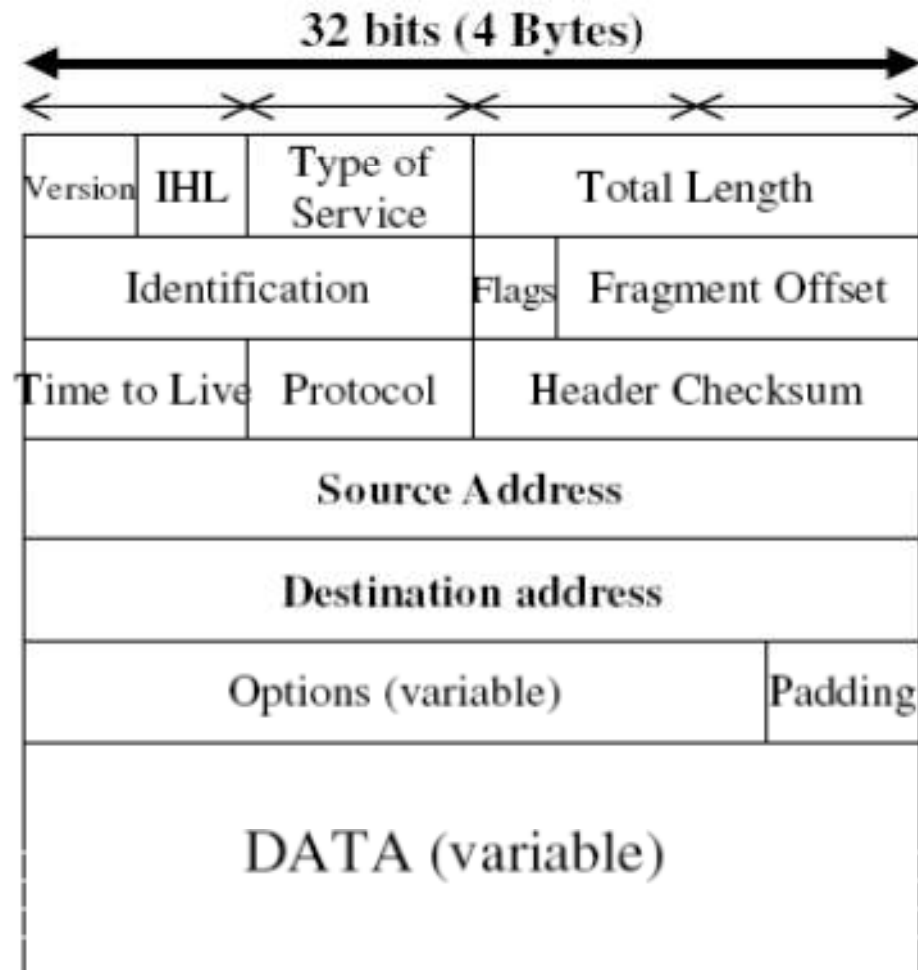
- Típuskód: *TCP/IP, DECnet...*



IP (Internet Protocol)

- Csomagokat továbbít, darabol és összerak
- Megbízhatatlan kapcsolatmentes datagramm szolgáltatást nyújt.
- Nincs garancia a sikeres célba érésre
- Ha bármilyen hiba lép fel: eldobja a datagrammot
- A datagrammok sorrendje megváltozhat

IP (Internet Protocol)



Kérdések az IP protokoll megvalósítással szemben

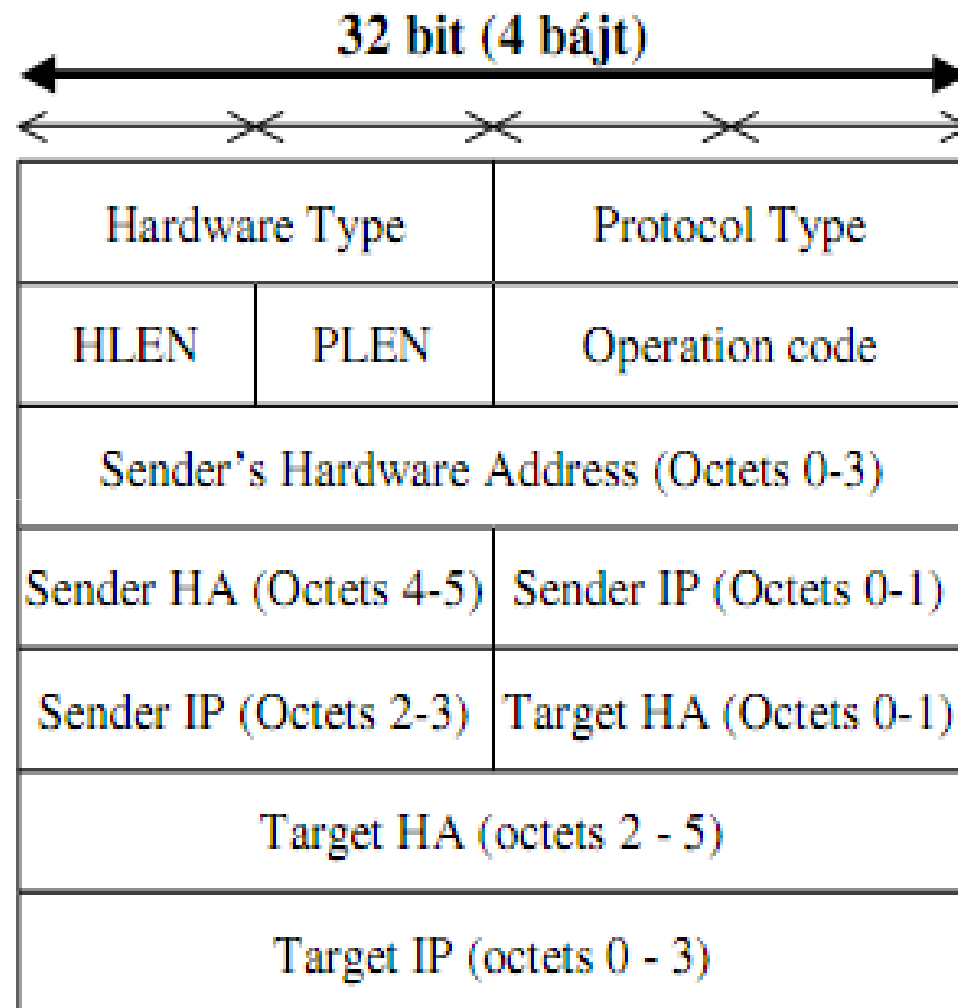
- Option küldés
 - Statisztikai jellegű feladatok:
 - Milyen csomópontokon keresztül jutott el a csomag a célhoz
 - Kis méretű protokollstack-ek nem tudják kezelni
- Fragmentáció
 - Elméletileg mindenkinek tudnia kell
 - Gyakorlatilag ritkán használt és sok memória kell hozzá
- 576byte-os csomag fogadása
 - Elméletileg kötelező gyakorlatilag nem biztos, hogy működik

ARP (Address Resolution Protocol)

- IP üzenetek küldéséhez ismerni kell a célállomás IP címét és a fizikai Ethernet címét is
- Az ARP segítségével az IP cím használatával meg lehet határozni a cél fizikai (Ethernet) címét
- Az ARP Ethernet broadcast-ot használ
- A feloldott Ethernet – IP cím párosokat egy helyi tárban tárolja az ARP

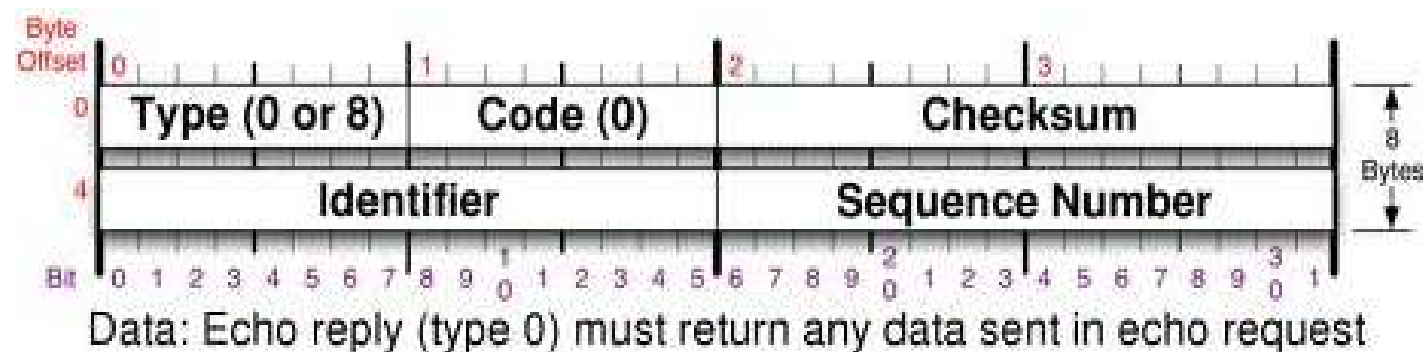
Az ARP fejléc

- Hardware Type:
 - Ethernet: 1
- Protocol Type:
 - Ethernet: 0x0800
- HLEN, PLEN
 - Címek hossza byteokban
- Operation code
 - Request
 - Reply



ICMP (Internet Control Message Protocol)

- Echo, and Echo Reply
- Destination unreachable
- Traceroute
- TTL exceeded



ICMP megvalósítási kérdések

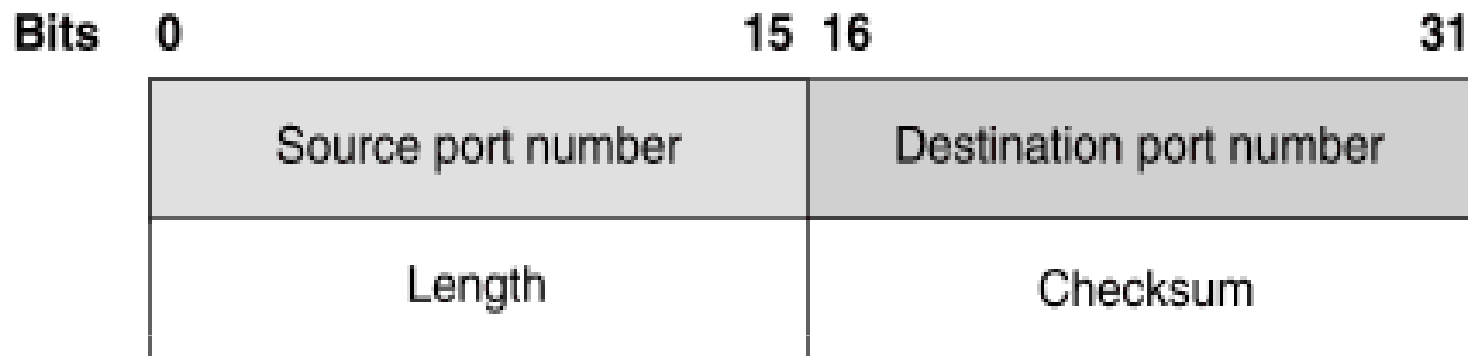
- Echo küldés, válasz
 - Mindig implementált

- Destination Unreachable ...:
 - Ezeket már nem mindig szokták implementálni

UDP (User Datagram protocol)

- Megbízhatatlan üzenettovábbítás
- Nem kapcsolat alapú
- Üzenet szórásra alkalmas
- Rendkívül kis erőforrás igényű

UDP (User Datagram protocol)



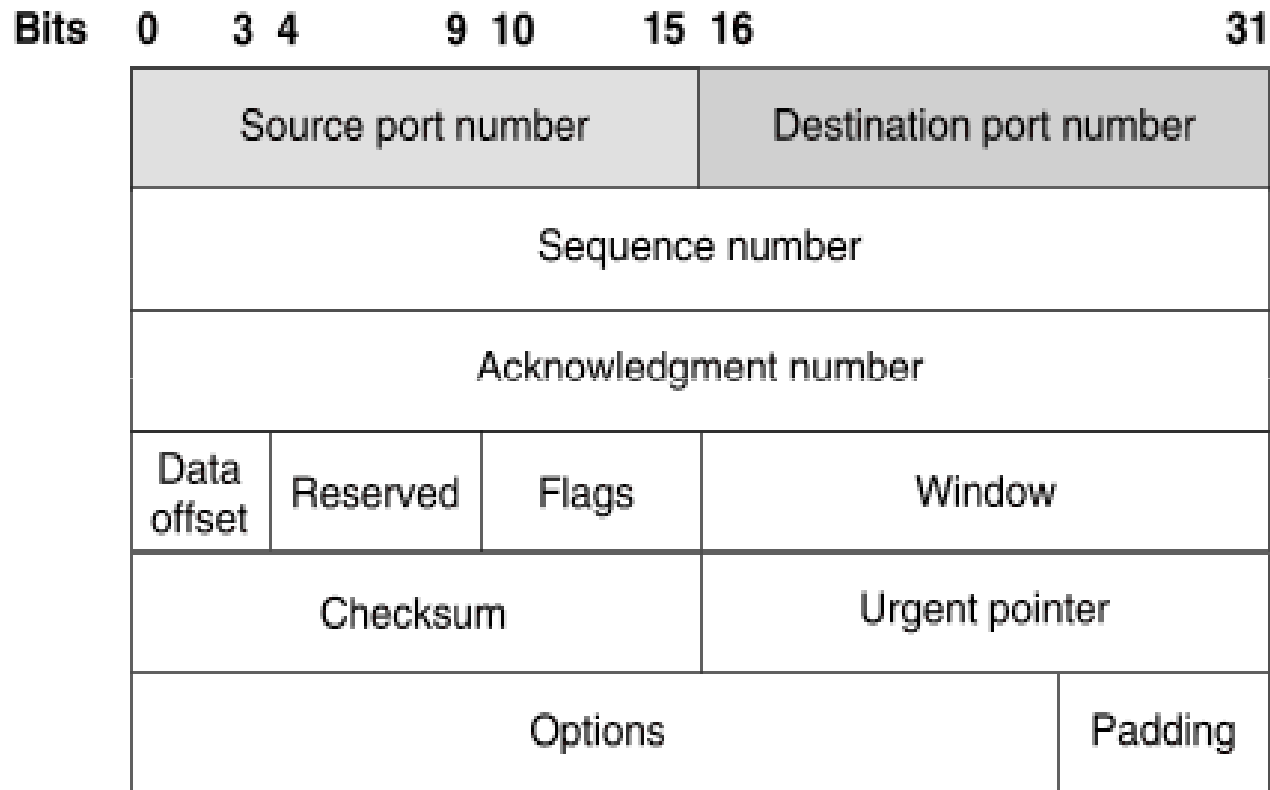
■ Checksum mező

- Opcionális a használata, nem mindig töltik ki.
 - Ethernet kereten úgyis van CRC ellenőrzés

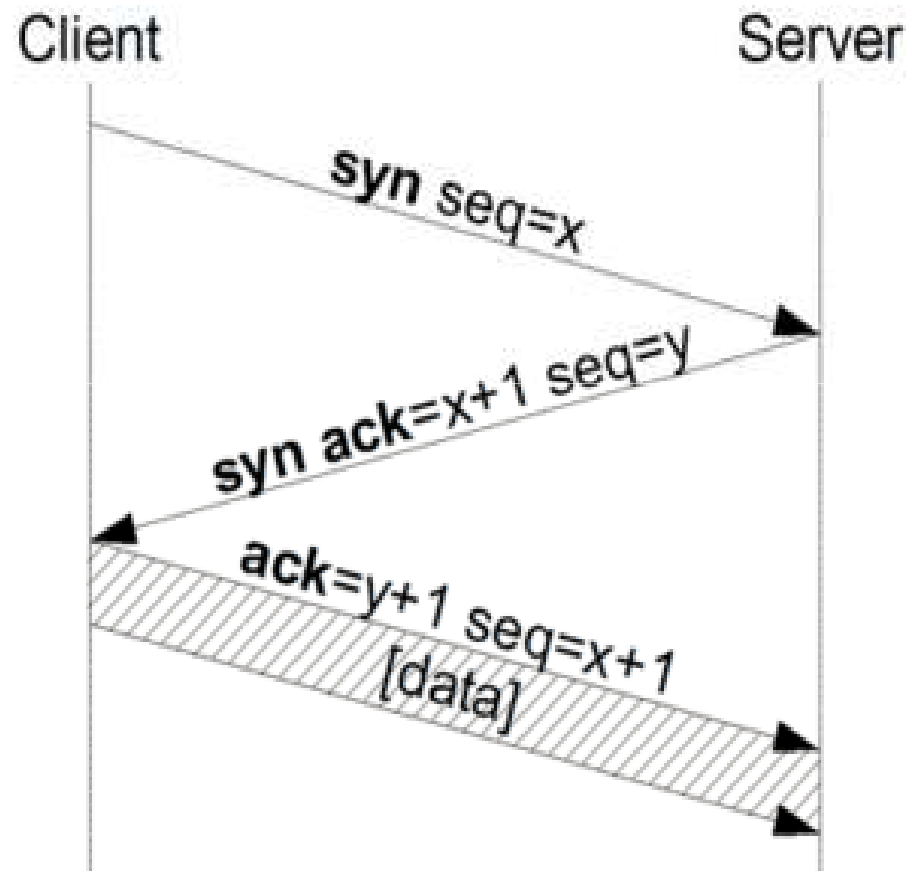
TCP (Transmission Control Protocol)

- Kapcsolat alapú megbízható adatátvitel
- Nem alkalmazható üzenetszórásra
- Az adatfolyamot részekre bontja
 - Jelentős helyi tárkapacitást igényel a szegmensek összerakásához

TCP (Transmission Control Protocol)



TCP (Transmission Control Protocol)



Web szerver alapú vezérlés

- TCP alapú, tehát nagy erőforrás
- HTML lapok nem túl erőforrás takarékosok
- HTML lapok tárolási módja
 - Flash, be statikusan
 - Dinamikusan valahol
- Web browserben elküldött
 - Post, get parancsokra reagálás

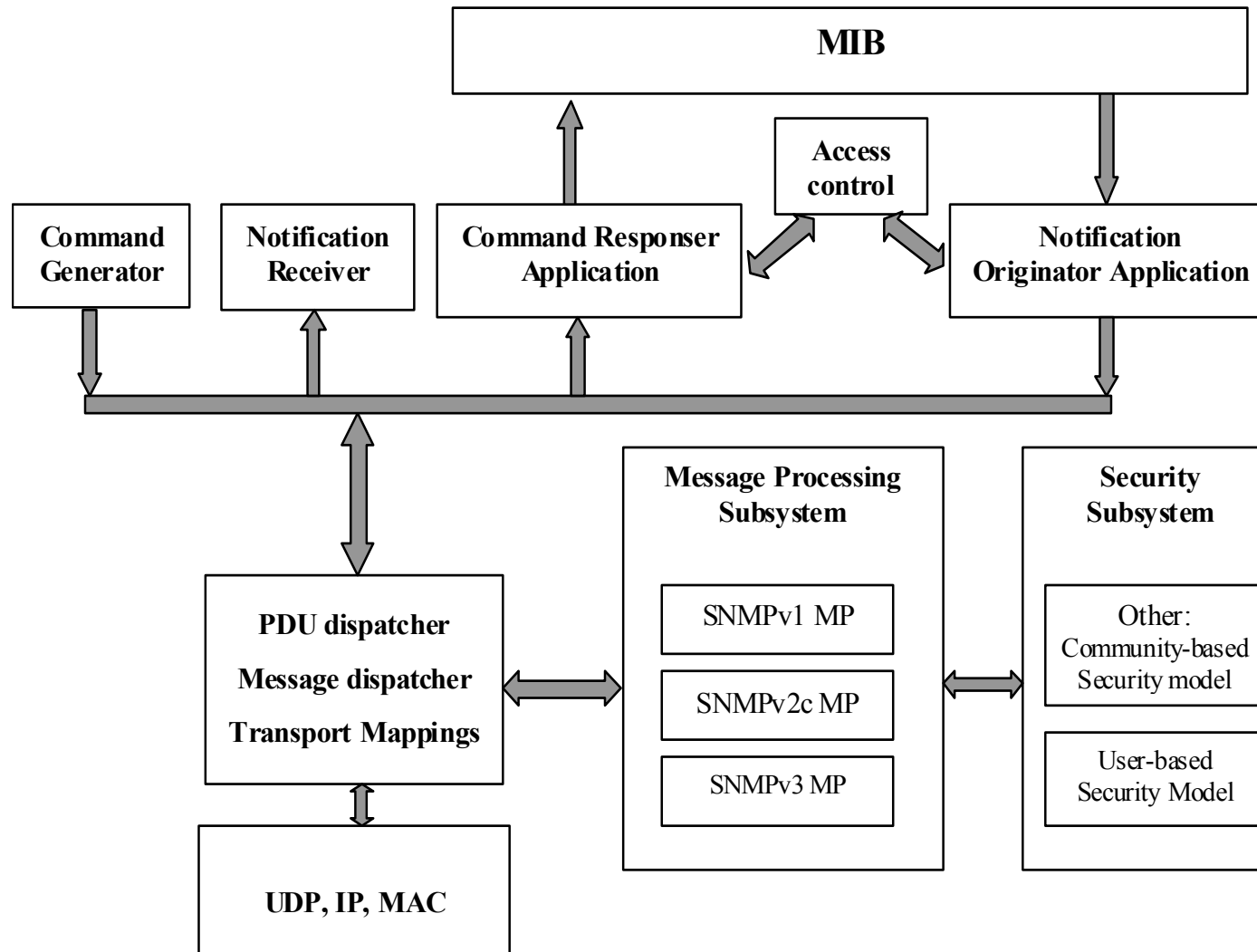
FTP alapú megoldás

- Viszonylag erőforrás igényes, TCP alapú
- Szenzor kezelés nem túl hatékony
- Nehezen konfigurálható
- Nagy mennyiségű adat átvitelére alkalmas

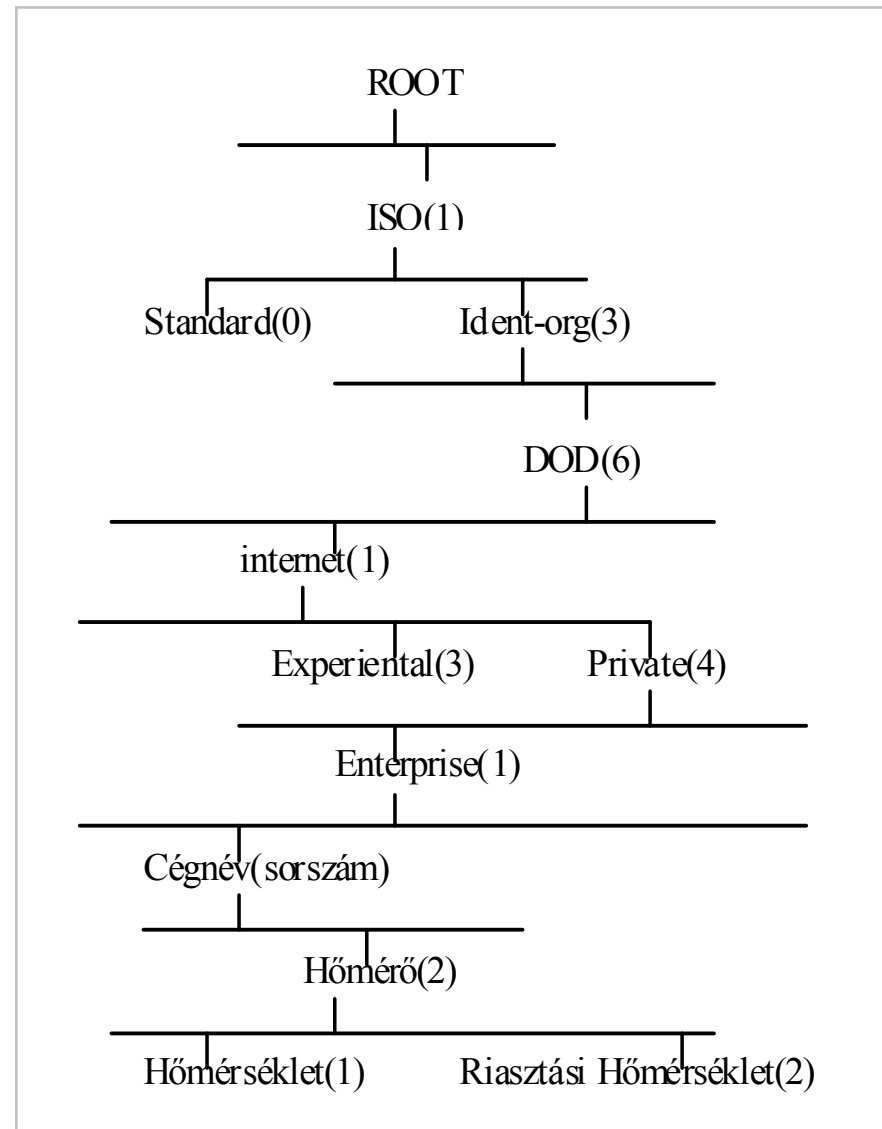
SNMP

- Erőforrás takarékos, UDP
- Jól használható sensor kezelésre, konfigurációra
- MIB-ek kezelése közepesen bonyolult
- Az ID-k kezelése nagyon szószátyár
- Nem hatékony nagymennyiségű információ átvitelére

SNMP ágens



SNMP MIB struktúra

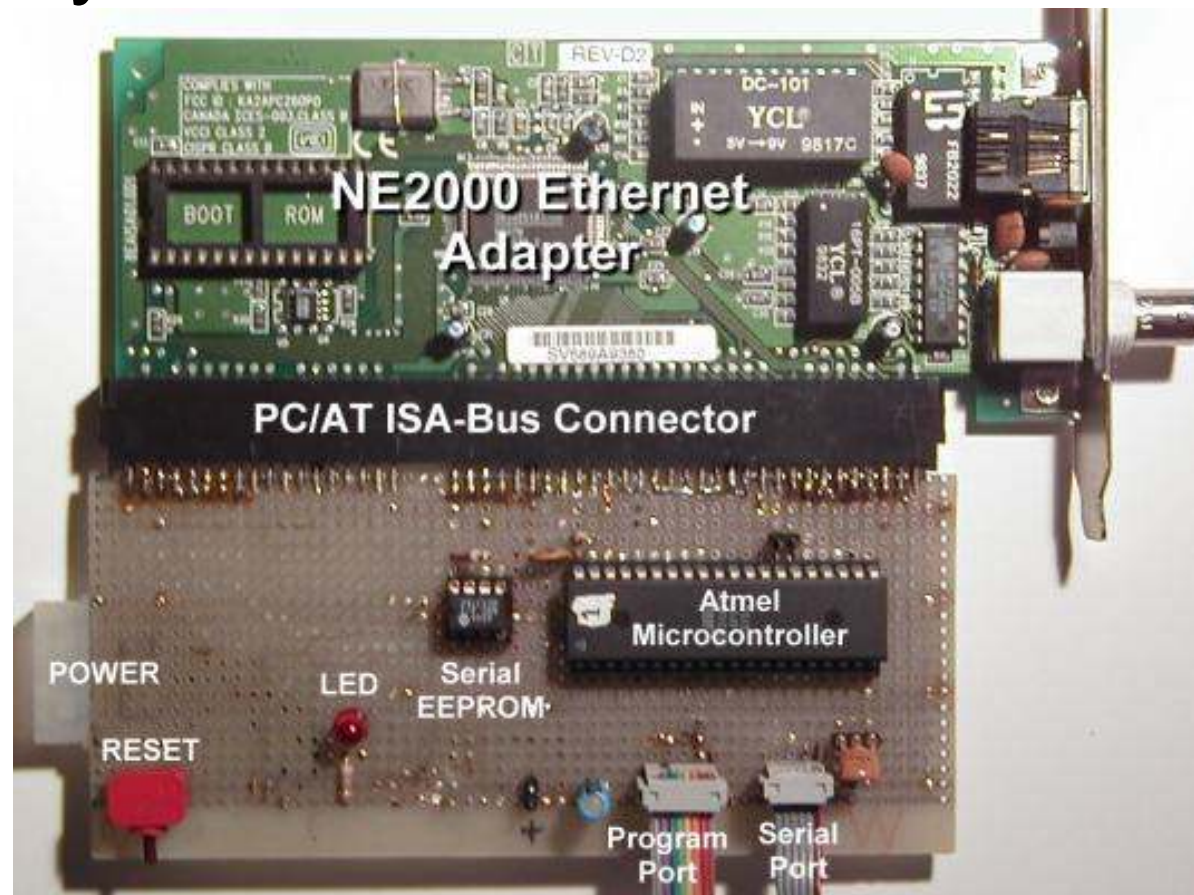


TFTP

- Erőforrás takarékos, UDP
- Igen primitív
- Bootolás-ra szokták használni

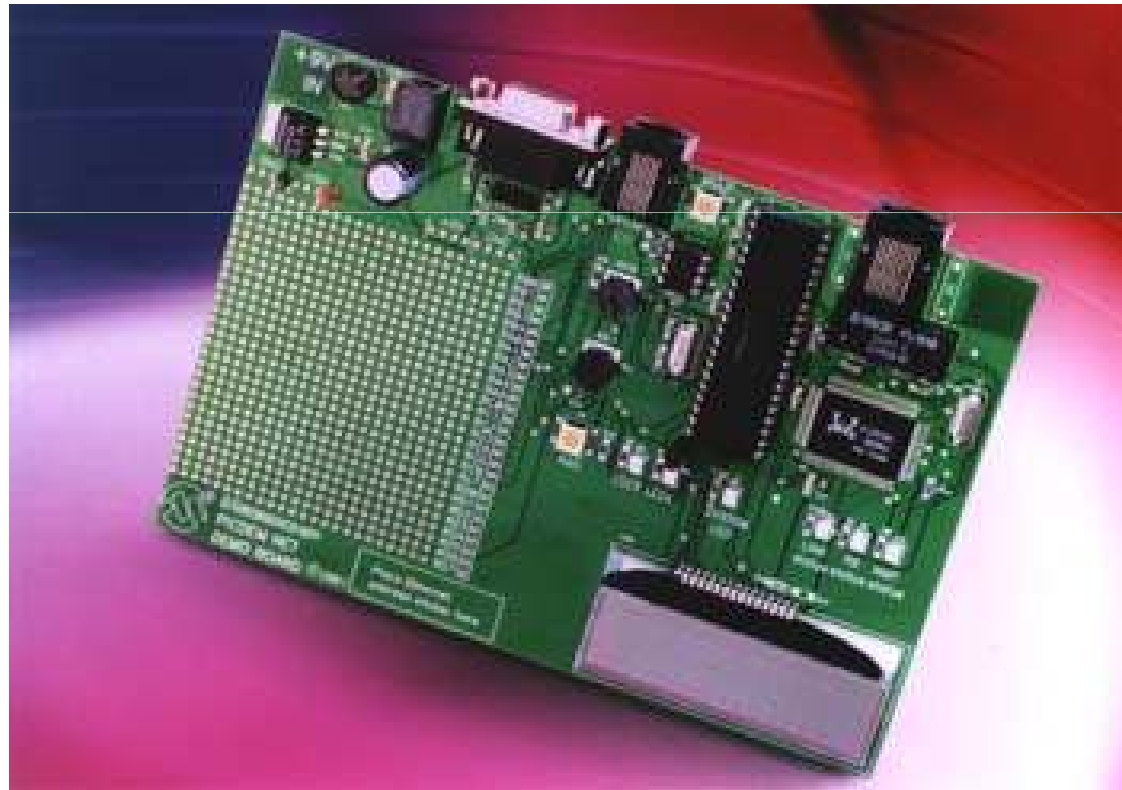
Első generációs megoldások 1995-2000

- ISA kártya + 8 bites controller



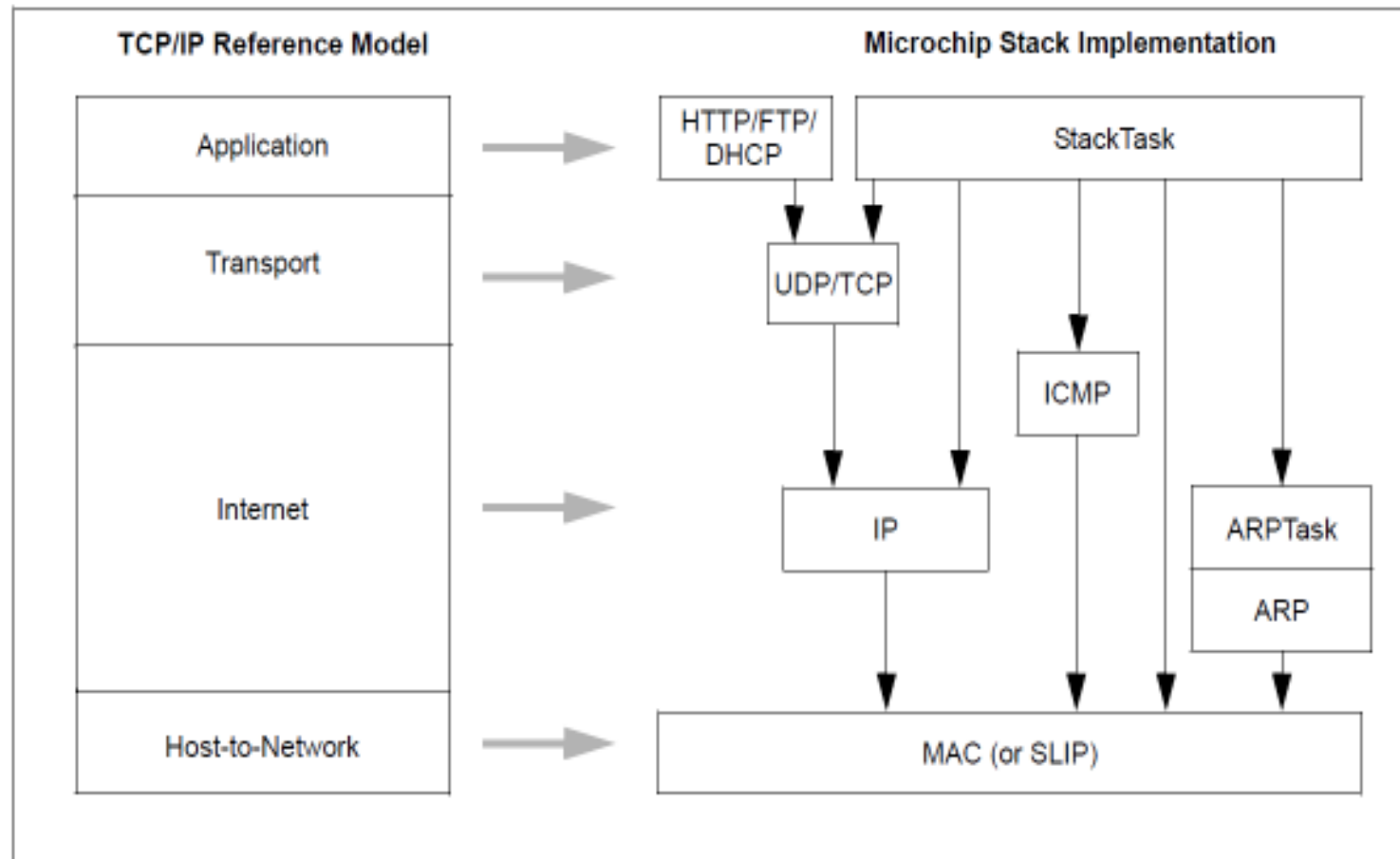
Második generáció megoldások 2002-2005

- 8 bites controller + Ethernet controller
 - Ethernet controller belső RAM-jának használata



Megvalósítások, Microchip AN833

FIGURE 2: COMPARING THE MICROCHIP TCP/IP STACK STRUCTURE TO THE TCP/IP REFERENCE MODEL



Microchip AN833, „ütemezése”

```
// Main entry point
void main(void)
{
    // Perform application specific initialization
    ...

    // Initialize Stack components.
    // If StackApplication is used, initialize it too.
    TickInit();
    StackInit();
    HTTPInit(); // Only if HTTP is used.
    FTPInit(); // Only if FTP is used.

    // Enter into infinite program loop
    while(1)
    {
        // Update tick count. Can be done via interrupt.
        TickUpdate();

        // Let Stack Manager perform its task.
        StackTask();

        // Let any Stack application perform its task.
        HTTPServer(); // Only if HTTP is used.
        FTPServer(); // Only if FTP is used.

        // Application logic resides here.
        DoAppSpecificTask();
    }
}
```

Microchip AN833, működése

EXAMPLE 1: THE STACK MANAGER ALGORITHM

```
If a data packet received then
  Get data packet protocol type
  If packet type is IP then
    Fetch IP header of packet
    Get IP packet type
    if IP packet type is ICMP then
      Call ICMP module
    else if IP packet type is TCP then
      Call TCP module
    else if IP packet type is UDP then
      Call UDP module
    else
      Handle not supported protocol
    End If
  End If
  Else if packet type is ARP then
    Call ARP module
  End If
End If
```

Microchip, AN833 erőforrás használata

TABLE 4: MEMORY USAGE FOR THE VARIOUS STACK MODULES USING HI-TECH® PICC-18™ COMPILER

Module	Program Memory (words)	Data Memory (bytes)
MAC (Ethernet)	906	5 ⁽¹⁾
SLIP	780	12 ⁽²⁾
ARP	392	0
ARPTask	181	11
IP	396	2
ICMP	318	0
TCP	3323	42
HTTP	1441	10
FTP Server	1063	35
DHCP Client	1228	26
IP Gleaning	20	1
MPFS ⁽³⁾	304	0
Stack Manager	334 ⁽⁴⁾	12+ICMP Buffer

Harmadik generáció megoldások 2005+

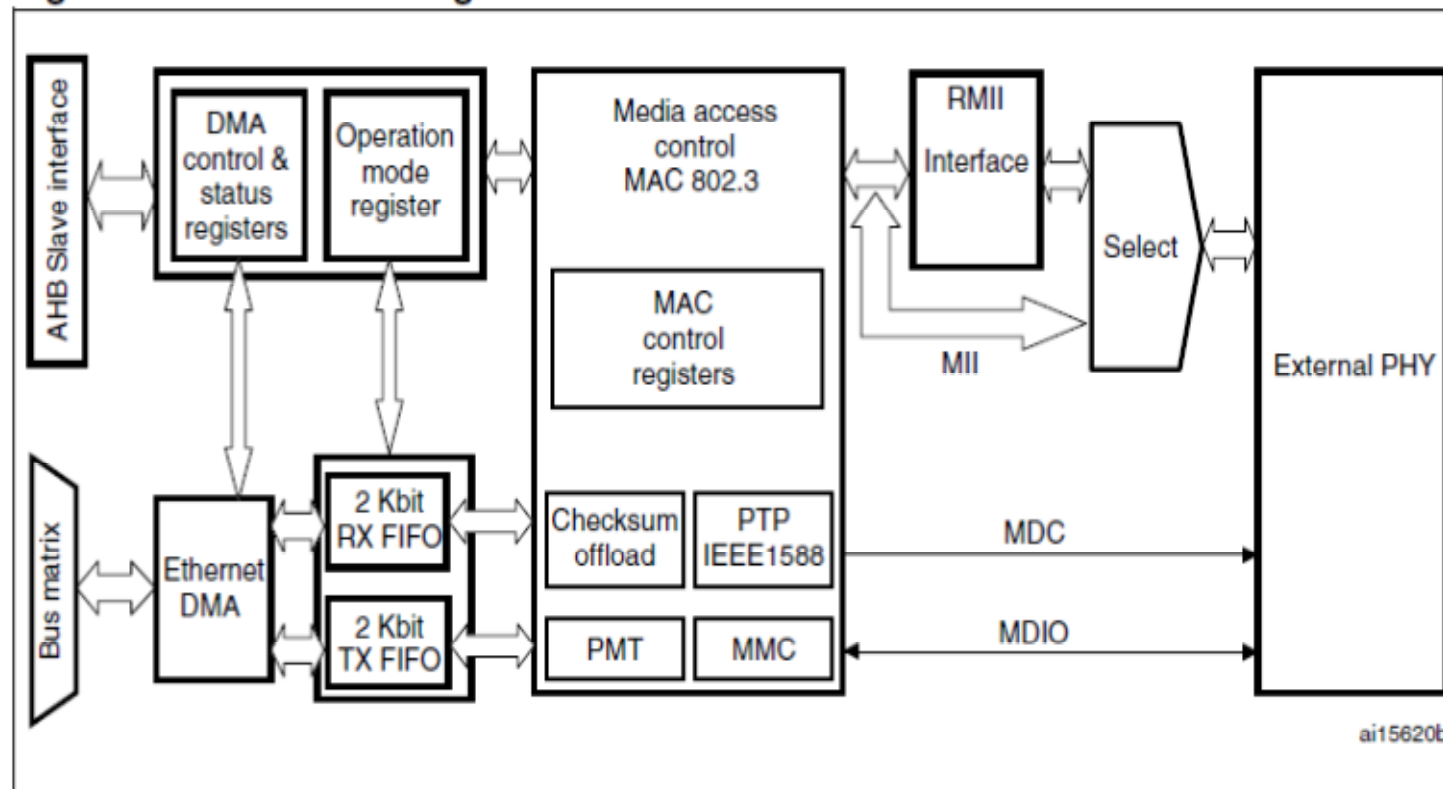
- Különálló TCP/IP chip
 - SPI: Microchip
 - Párthuzamos
 - Sorosporti protokoll: Matcport
- 32 bites vezérlők
 - Integrált Ethernet controller
 - Megfelelő méretű belső RAM
 - Elég gyors processor
- Adam Dunkels protokollstackjei
 - LwIP
 - uIP

Az STM32 Ethernet Controller-e

- 10/100 Mbit/s adatátviteli mód támogatás
- LAN wakeup üzenet azonosítás,
- Segíti az IPv4 header checksum és TCP, UDP, ICMP checksum-ok ellenőrzését.
- 2-KB Transmit FIFO
- 2-KB Receive FIFO
- IEEE 1588-2002. 64 bites timestampek.

Az STM32 Ethernet Controller-e

Figure 295. ETH block diagram



LwIP, uIP

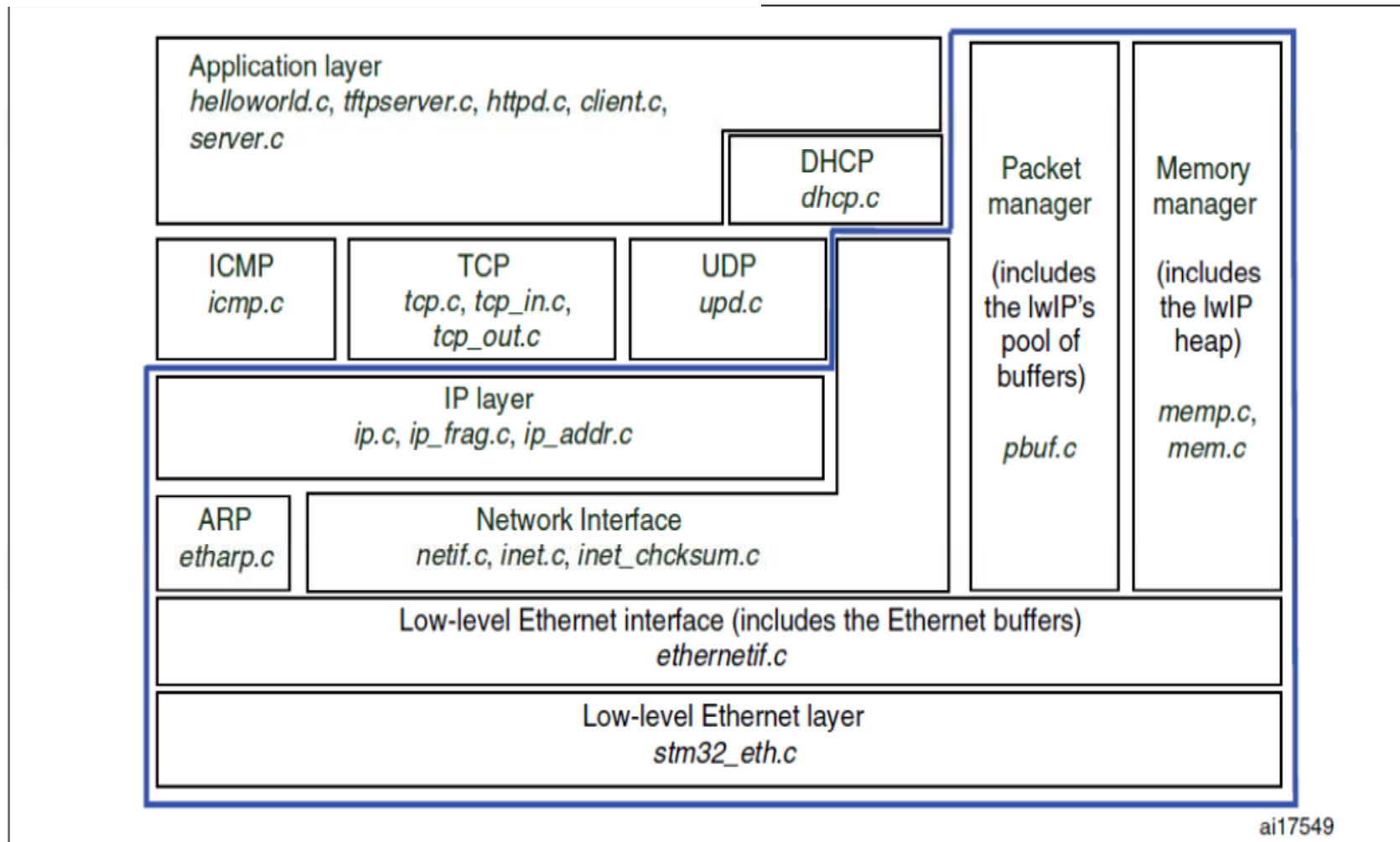
- Mind a kettő RFC kompatibilis
- A uIP alacsonyabb kategóriájú vezérlőkre, mint a LwIP
- Buffer memoria a legkritikusabb,
 - Kevesebb a RAM, mint a ROM
- Traditionális BSD socket API
 - Nem eseményvezérelt programozáshoz
 - Rengeteg plusz kód.
- Saját lwIP és uIP API-k
 - Egyszerűbb kód
 - Esemény vezérelt orientált IF

LwIP könyvtárstruktúra

- **lwip/src/lwip/src/api** - the [Netconn API](#), [Socket API](#), and the [tcpip thread](#)
- **lwip/src/core** - core code: [DHCP](#), [TCP](#), [UDP](#), and support code (memory, netif, etc)
- **lwip/src/core/ipv4** - [IPv4](#), [ICMP](#)
- **lwip/src/core/ipv6** - [IPv6](#)
- **lwip/src/core/snmp** - [SNMP](#)
- **lwip/src/include** - all headers and includes
- **lwip/src/netif** - [ARP](#) and sample Ethernet driver
- **lwip/src/netif/ppp** - [PPP](#)

LwIP architektúra

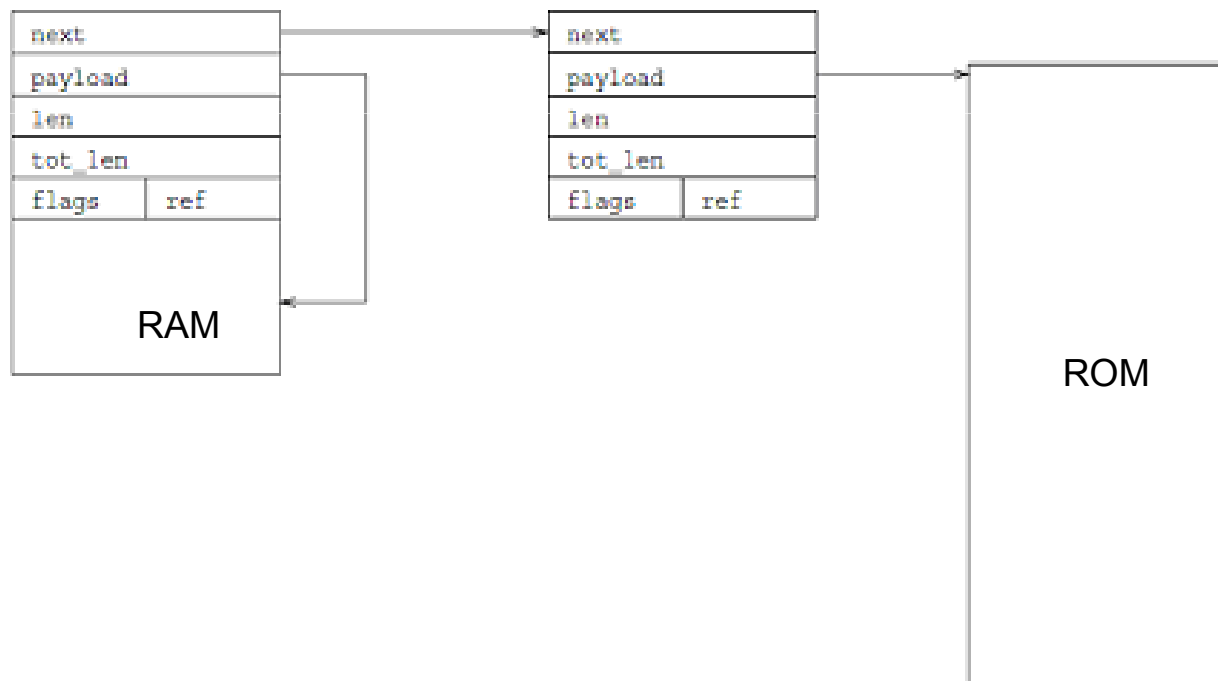
LwIP az STM32 Conectivity Line sorozatra



LwIP memóriakezelés

- pbuf sorok használatával oldja ezt meg.
 - pbuf-ok lehetnek RAM (dinamikusan foglalt)
 - ROM és POOL (fix méretű memóriaterület) típusúak.

Többnyire a kimenő adatok RAM és ROM buffereket használnak, míg a bejövők POOL típusút.



Network Interface

- Linkelt Listában

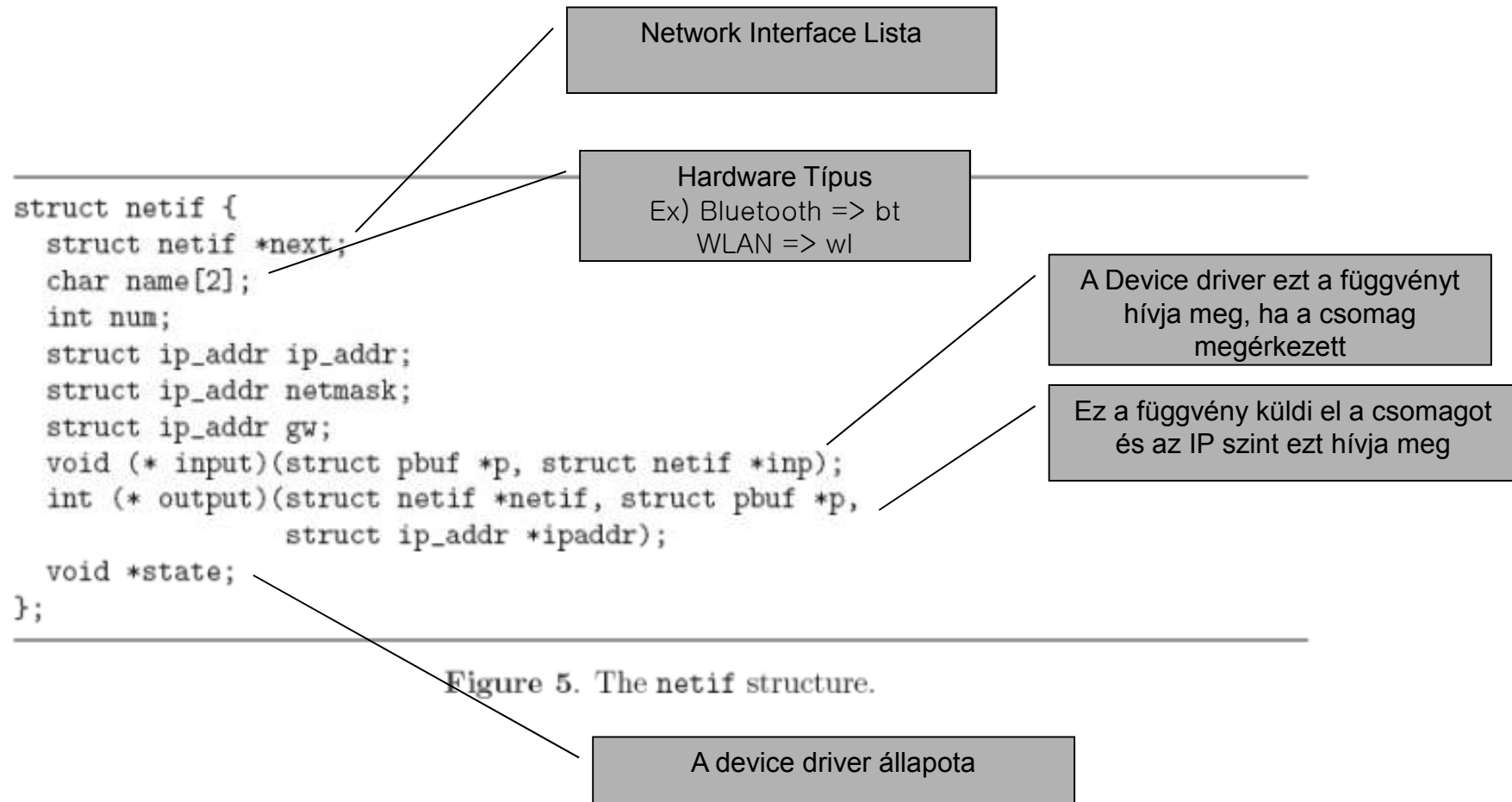


Figure 5. The netif structure.

IP feldolgozás

■ Csomag fogadás

- Network device driver megívja az *ip_input()* függvényt.
 - IP version, header length ellenőrzés
 - Header checksum számolás
 - Destination address ellenőrzés

■ Csomag küldés

- *ip_output()* függvényen keresztül
 - A megfelelő network interface megtalálása
 - IP header mezők kitöltése
 - IP header checksum kiszámítása

IP feldolgozás

- Csomag forwardolás

- Forwardolni kell, ha

- Ha az egyik Network interface-nek sem ugyanaz a címe, mint a kimenő csomag cél IP címe

- *ip_forward()* függvény

- TTL mező csökkentés
 - Ha TTL nulla akkor ICMP error message küldése

IP feldolgozás

- ICMP processing

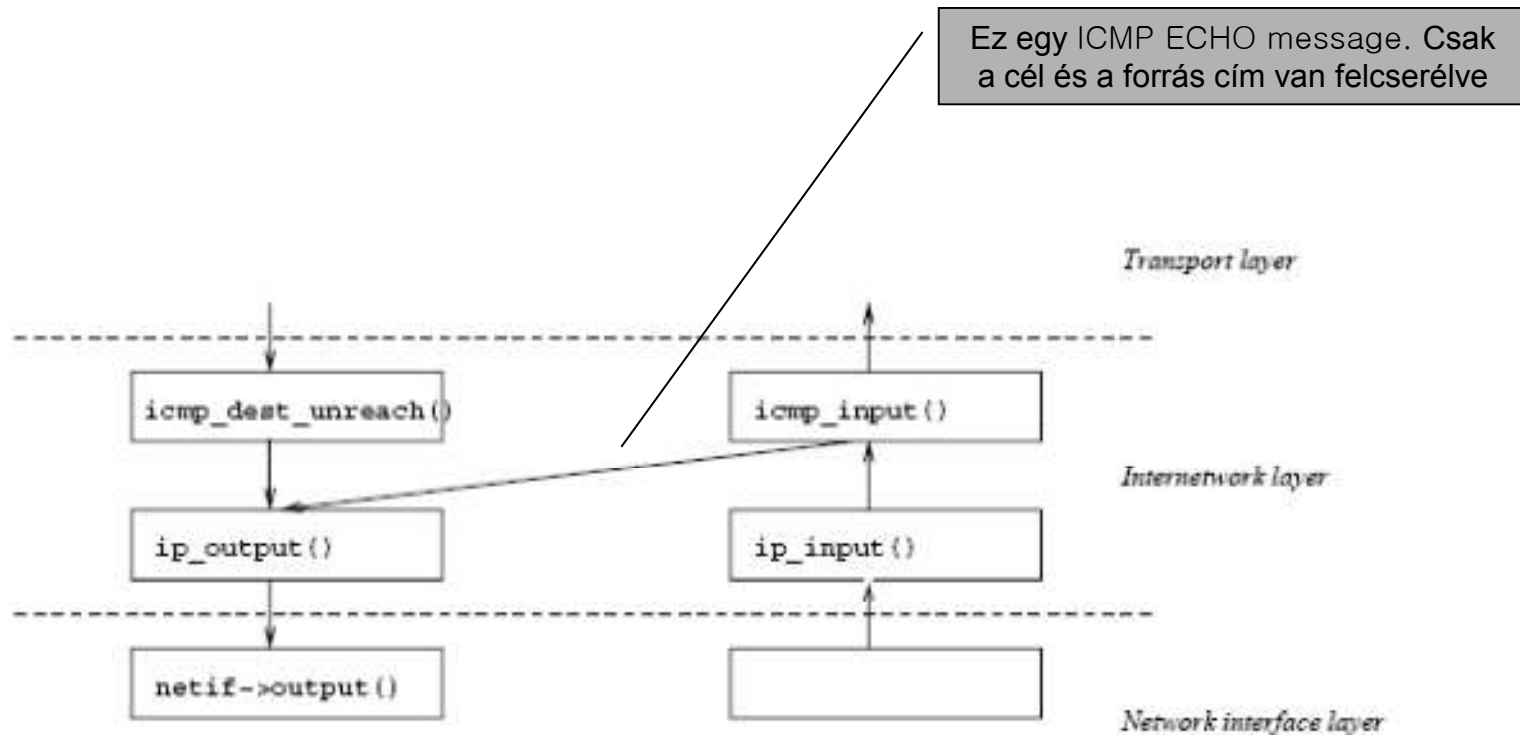


Figure 6. ICMP processing

UDP feldolgozás

- Az udp_pcb structure

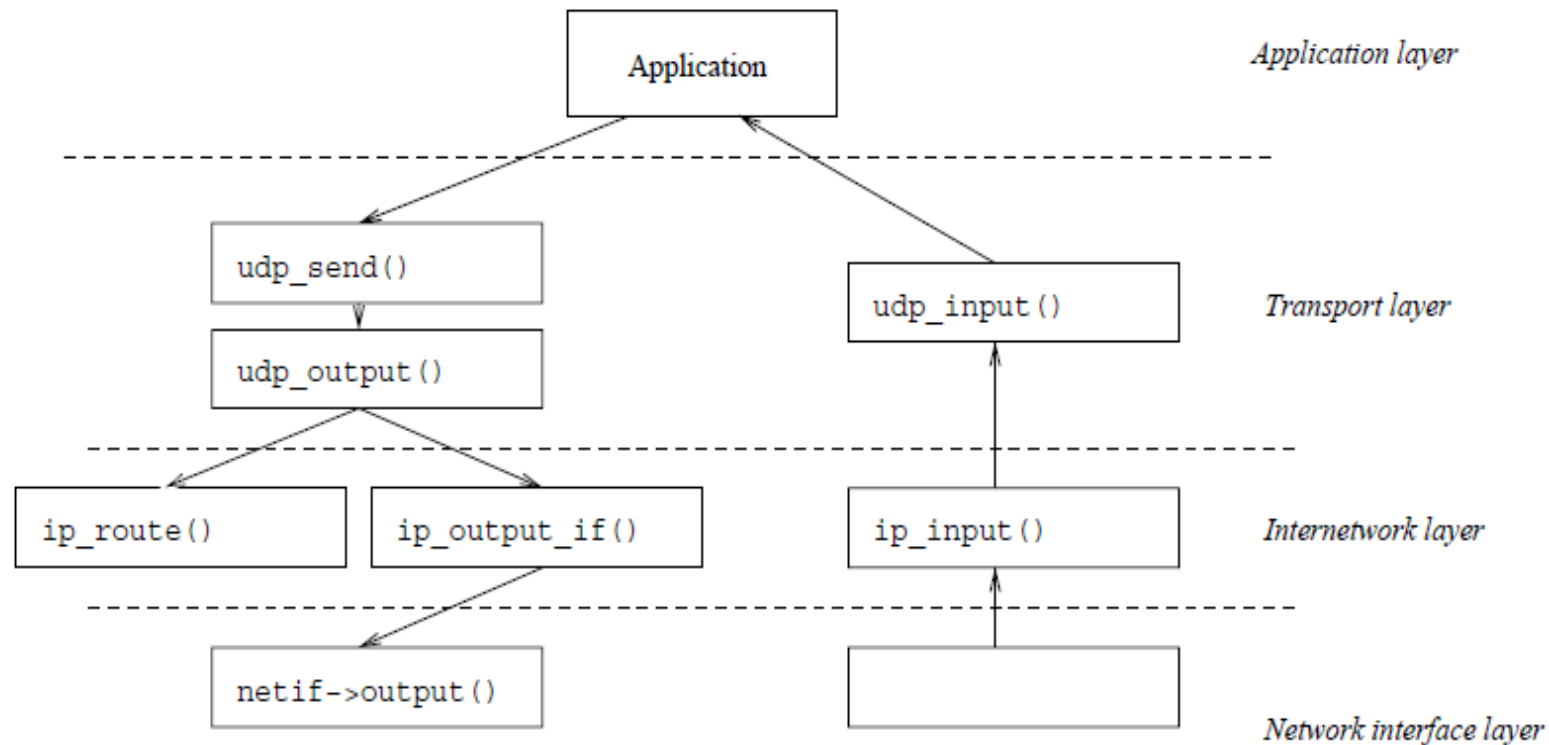
```
struct udp_pcb {  
    struct udp_pcb *next;  
    struct ip_addr local_ip, dest_ip;  
    u16_t local_port, dest_port;  
    u8_t flags;  
    u16_t chksum_len;  
    void (*recv)(void *arg, struct udp_pcb *pcb, struct pbuf *p);  
    void *recv_arg;  
};
```

Az UDP csomagok egy linked listában vannak nyilvántartva

Ez a függvény hívódik meg amikor a datagram megérkezik

Figure 7. The udp_pcb structure

UDP alapú alkalmazás



Feldolgozás

```
struct tcp_pcb {
    struct tcp_pcb *next;
    enum tcp_state state;      /* TCP state */
    void (* accept)(void *arg, struct tcp_pcb *newpcb);
    void *accept_arg;
    struct ip_addr local_ip;
    u16_t local_port;
    struct ip_addr dest_ip;
    u16_t dest_port;
    u32_t rcv_nxt, rcv_wnd;    /* receiver variables */
    u16_t tmr;
    u32_t mss;                 /* maximum segment size */
    u8_t flags;
    u16_t rttest;             /* rtt estimation */
    u32_t rtseq;              /* sequence no for rtt estimation */
    s32_t sa, sv;            /* rtt average and variance */
    u32_t rto;                /* retransmission time-out */
    u32_t lastack;           /* last ACK received */
    u8_t dupacks;            /* number of duplicate ACKs */
    u32_t cwnd, u32_t ssthresh; /* congestion control variables */
    u32_t snd_ack, snd_nxt,   /* sender variables */
        snd_wnd, snd_wl1, snd_wl2, snd_lbb;
    void (* recv)(void *arg, struct tcp_pcb *pcb, struct pbuf *p);
    void *recv_arg;
    struct tcp_seg *unsent, *unacked, /* queues */
        *ooseq;
};
```

Ez a függvény hívódik meg,
ha a Listener csatlakozott

Következő
sequence number

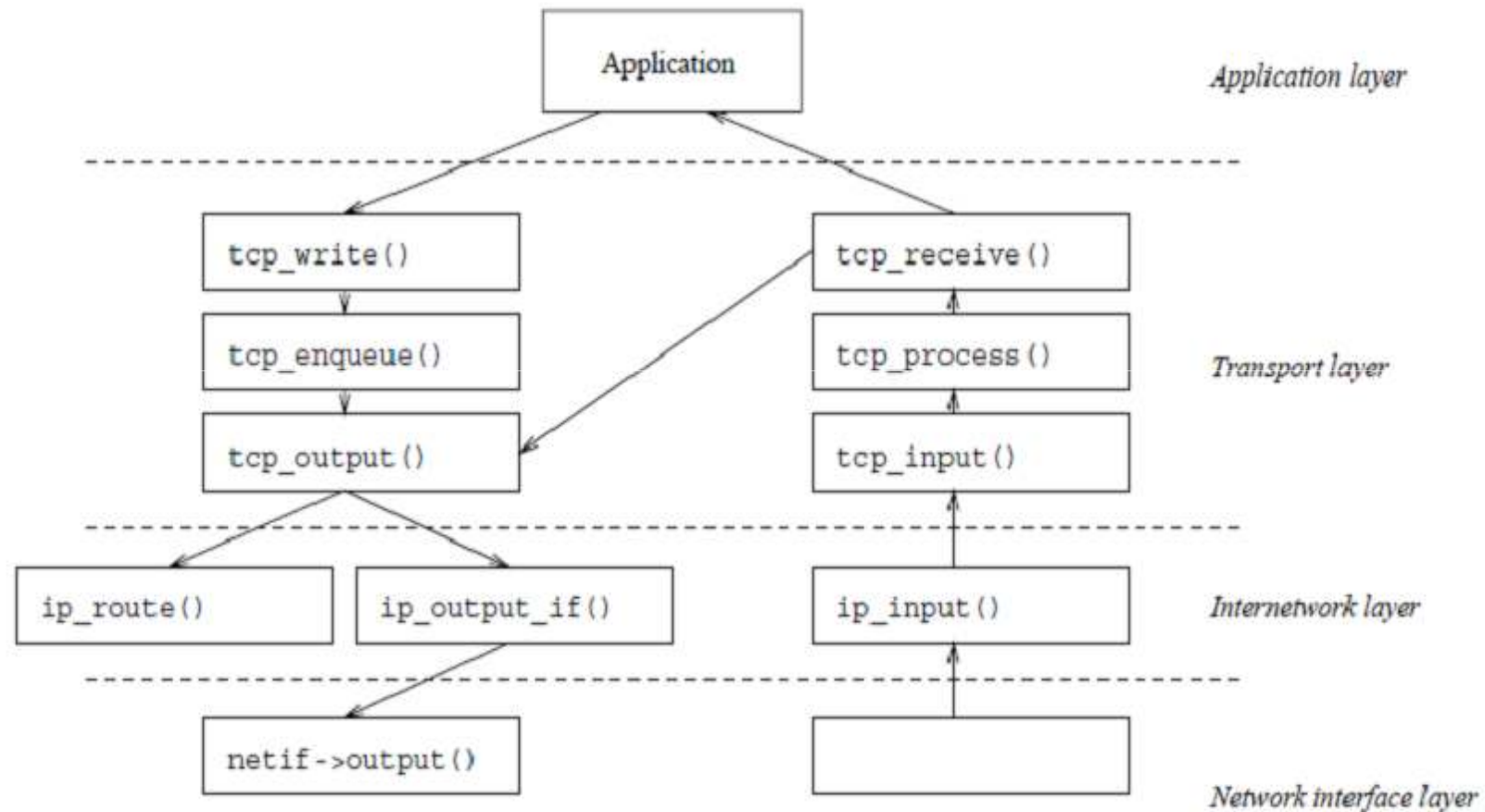
A Receiver's
window

Timer a TIME-
WAIT állapothoz

Ezzel a függvénnyel adjuk
át a vett adatot az
Alkalmazás rétegnek

Figure 10. The tcp_pcb structure

TCP alapú alkalmazás



A LwIP memóriaigénye

Table 2. STM32F107xx lwIP demonstration footprint⁽¹⁾

Modules	Description	Flash memory (bytes)	SRAM (bytes)
Mandatory modules	Ethernet driver and interface, lwIP memory management and IP modules	7848	49590
TCP modules	TCP packet handling using the raw API	7562	80
UDP modules	UDP datagram handling using the raw API	856	4
Optional modules	ICMP	394	0
	DHCP	3164	4
Application modules	Hello word	376	0
	TFTP server	1467	1684
	Web server	32607 ⁽²⁾	4617 ⁽³⁾
	Server	328	0
	Client	412	4
STM32 firmware	STM32F107xx's firmware library	2296	24
STM3210C-EVAL board	STM3210C-EVAL dedicated files	8852 ⁽⁴⁾	64
Main and system initialization	main file and system initialization	2480	1624 ⁽⁵⁾
efsl	File system	8338	0
Others	Standard libraries	1884	105
Total		78764	57800

uIP

- Lényegesen kisebb kód
- Egy globális memória pool
 - Egyszerre egy üzenet tárolása
 - A küldésre is felhasználja
- Nagyon limitált fragmentáció támogatás (egyszerre egy csomag)
- Nagyon Limitált TCP retransmission
 - Csak az utolsó csomag újraküldése
- Nagyon kicsivel tud többet, mint a Microchip AN833