UNIVERSITY OF MICROCHIP
UofM
MASTERs 2007

11050 TCP

Hands on with the Microchip TCP/IP Stack (8 hours)

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Class Objective

When you finish this class you will:

- Know of the Microchip Ethernet and Internet networking solutions
- Know of protocols such as HTTP, Telnet, NBNS, SNTP, DNS, DHCP, etc. and their value to your application
- Understand how to modify the Microchip TCP/IP stack to develop your applications

University of Microchip UofM MASTERs 2007

Agenda

- **Ethernet – Overview**
  - Demo 1 – What's possible
  - Lab 1 – Join the network
- **TCP & UDP**
  - Lab 2 – Peek in with Wireshark
- **ARP & DNS**
- **HTTP2 & MPFS2**
  - Lab 3 – Change variables, form submission, and authentication options
- **Simple Mail Transport Protocol (SMTP)**
  - Lab 4 – Send email to yourself
- **Microchip Bootloader**
- **Other Modules**
  - Lab 5 – Bootloader, Telnet, SNTP, ICMP, DHCP
- **Frequently Asked Questions**

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

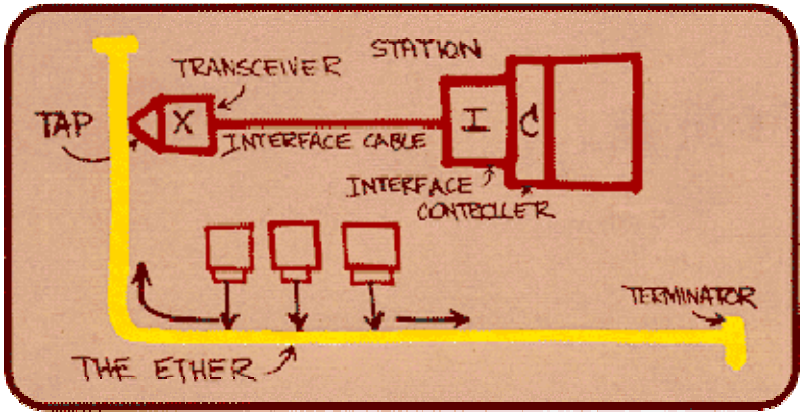
Summary

- Overview of Ethernet products and status
- Wireshark and TCP/IP Stack protocols
- Editing web pages to suit your application
- Experience using SMTP, HTTP2, MPFS2, Telnet, Bootloader, SNTP, and other modules

Ethernet – Overview

Why Ethernet?

- Ethernet is the most widely deployed network in offices and industrial buildings
- Ethernet's infrastructure, interoperability and scalability ensure ease of development
- Once equipment is connected to an Ethernet network, it can be monitored or controlled through the Internet



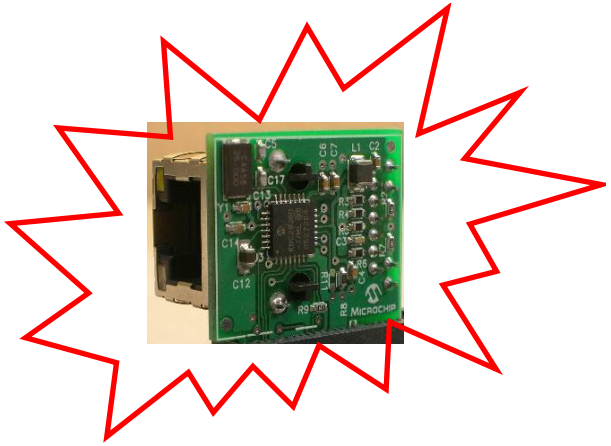
Ethernet Properties

- **Frame based**
  - All packets must be 64 to 1518 bytes long
- **Data is protected from corruption through hardware CRC checking**
- **Hardware usually filters unwanted traffic from arriving at your protocol stack**
  - Ethernet switches/routers do not send packets to you if they are not addressed to you
  - Ethernet controller contains receive hardware filters
- **Ethernet provides best effort transmission**
  - Packets that cannot be delivered or are damaged in transmit are thrown away, requiring application intervention

What can I do?

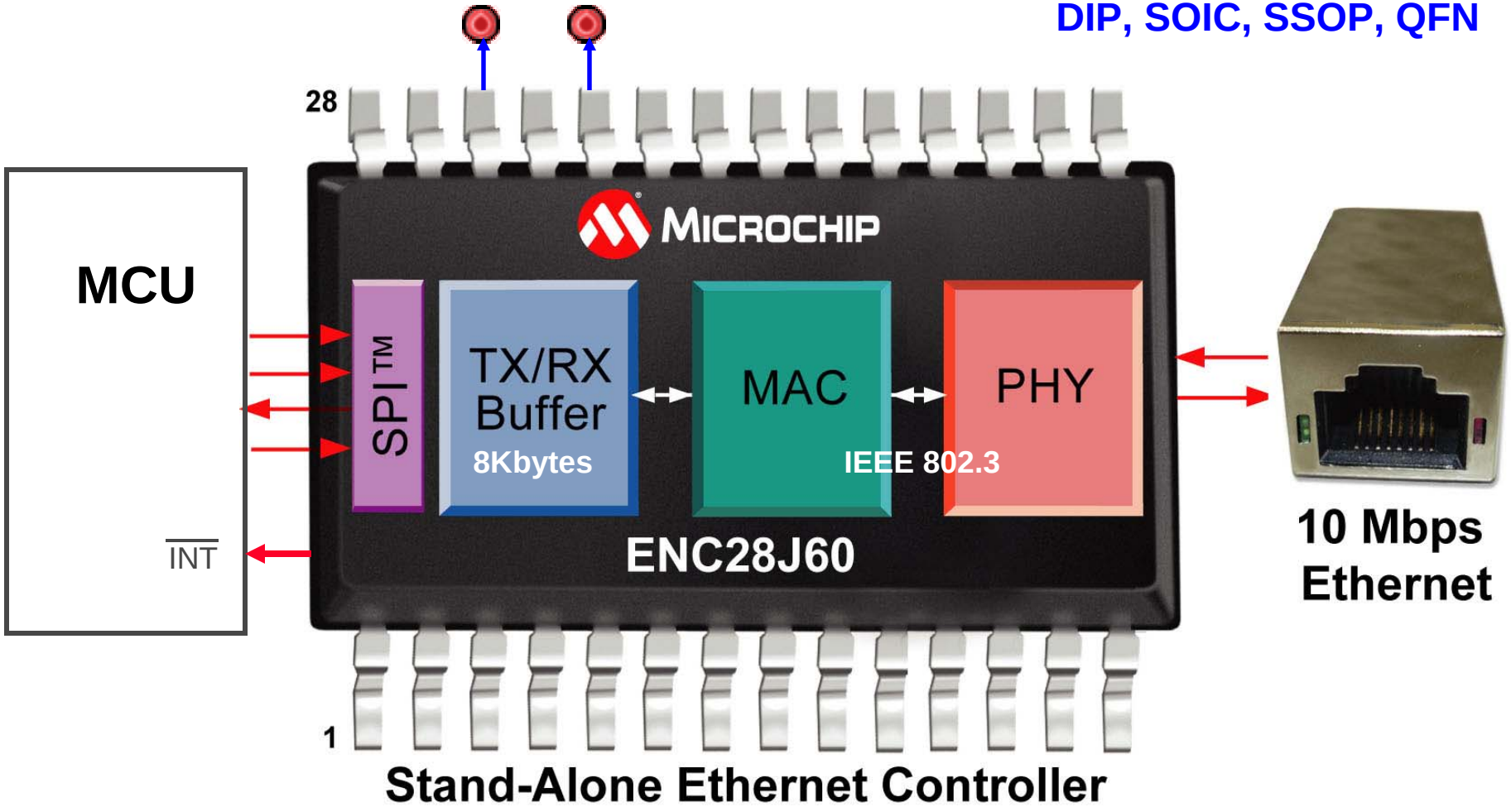
Provides Remote Application Access

Demo 1



- **Some possibilities…**
  - A stand alone Internet MP3 radio
  - On demand status and control system
  - Legacy RS232/Serial to Internet bridge

ENC28J60 Ethernet Controller



PIC18F97J60 Family PIC® MCU with Integrated Ethernet

- PIC18F97J60
  - 64K, 96K or 128K bytes Flash
  - 3.8K bytes RAM
  - 8K bytes TX/RX Buffer
  - 41.667 MHz max @ 3.3V
- Package (TQFP)
  - 64/80/100-Pins
- Other:
  - External Memory Bus
  - 16 channel 100ksps 10-bit A/D
- Communication
  - Ethernet: MAC+PHY (IEEE 802.3 10BASE-T)
  - 2 EUSARTs
  - 2 MSSP (SPI /MI^{2}C)

Silicon Products

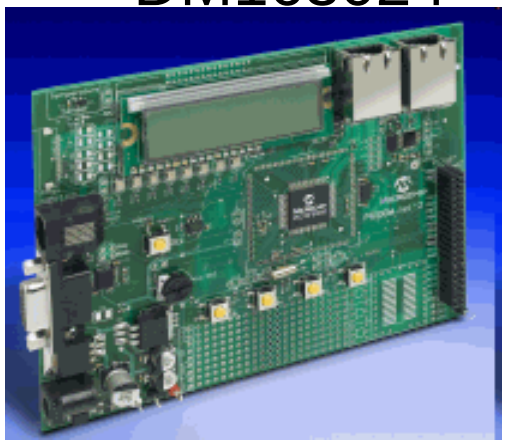
- **ENC28J60 – In production**
  - Stand-alone 10BaseT with 20MHz SPI
  - 3.3V, 8KB RAM
  - 28-pin SOIC, SSOP, QFN, SPDIP
- **PIC18F97J60 family – In production**
  - Integrated PIC18, features similar to PIC18F87J10 + ENC28J60 memory mapped
  - 3.3V, 64-128K flash, 3.8KB + 8KB RAM
  - 64, 80, or 100-pin TQFP
- **ENC624J600 family – Under development**
  - Stand alone 10/100 with ~20MHz SPI or ~20MHz 8/16 parallel bus interface
  - 28-pin SOIC, QFN, SPDIP and 44, 64-pin TQFP

External Components For Ethernet Solution

- 25MHz 50ppm crystal
- Ethernet transformer module
- RJ45 jack
- ~11 resistors
- ~10 capacitors
- 1 ferrite bead
- Unique MAC address
- 3.3V 200mA power supply

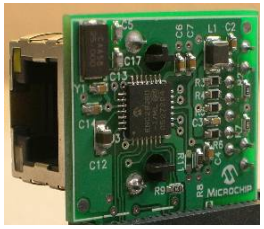
Boards

PICDEM™.net 2
-DM163024



Ethernet PICtail™
-AC164121

Ethernet PICtail™
Plus
-AC164123

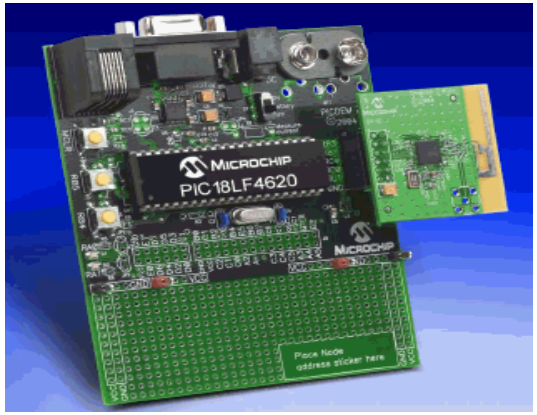


Ethernet PICtail™ Supporting Demo Boards

PICDEM™ HPC Explorer Board (DM183022)

Recommended due to 128KB Flash size

PICDEM Z Board (AC163027-1)

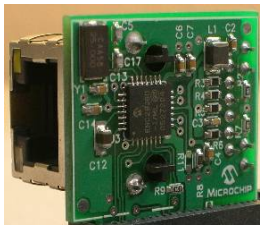


Ethernet PICtail™ Daughter Board

Others (dsPICDEM™ 1.1, etc)
- Requires air-wiring to PICtail connector

Ethernet PICtail™ Plus Supporting Demo Board

Explorer 16
(DM240001)



Ethernet PICtail™ Plus
Daughter Board

Stack Software

- **Version 3.75 and below**
  - Obsolete, strongly encourage updating
- **Version 4.02, 4.10 and above**
  - 4.10 is latest available
  - Many, many improvements and robustness enhancements from years past
  - Includes TCP, UDP, DHCP, DNS, HTTP, FTP, Telnet, NBNS, ICMP, SNTP, SNMP, SMTP, UART bridging, and more

University of Microchip UofM MASTERs 2007

Lab 1

- **Join our network**
  - Program your board with a unique MAC address and hostname
  - Edit MY_DEFAULT_MAC_BYTE# in `TCPIPConfig.h`.
  - Edit MY_DEFAULT_HOST_NAME in `TCPIPConfig.h`.
- **Upload web pages to the board**
  - Use MPFS2.exe
- **Access the board using the hostname, ex: http://mchpboard/**

Microchip's TCP/IP Stack

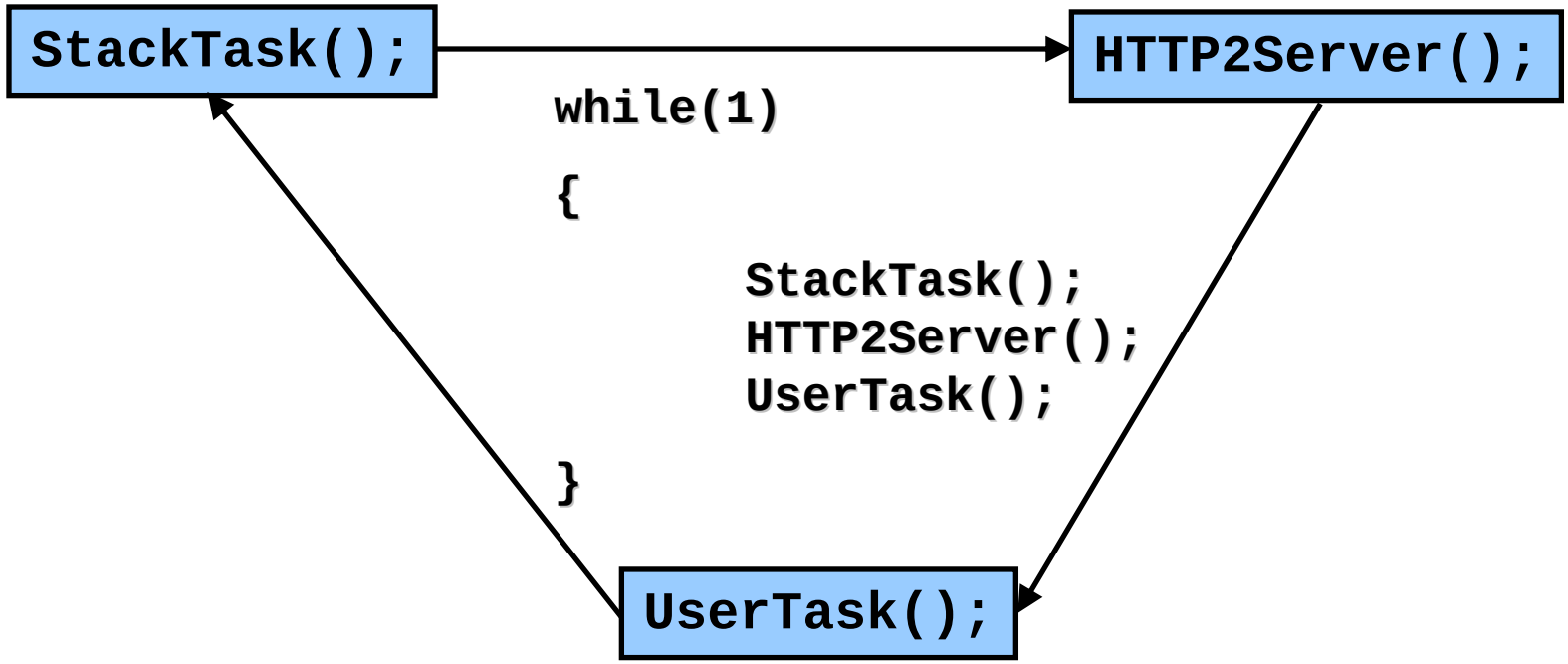
Microchip's TCP/IP Stack

- **Microchip App Note AN833 & AN870**
- **Source code available for free**
  - No fee license agreement
  - Must use with Microchip PIC® MCU or dsPIC® DSC
  - Download off Microchip website
- **Standard Microchip technical support**
- **Suite of files**
  - 'C' Source files and PC based utilities

TCP/IP Stack Features

- **Out-of-box support for Microchip MPLAB® C18, C30, and HI-TECH PICC-18 compilers**
  - Supports PIC18, PIC24F, PIC24H, dsPIC30F, and dsPIC33F processors
- **RTOS independent – Cooperative Multitasking Environment**
- **Modular Design**
  - All big features can be selectively removed by commenting out a single line
  - Implements full TCP state machine
  - Multiple socket support for TCP and UDP

Round Robin Task Switching



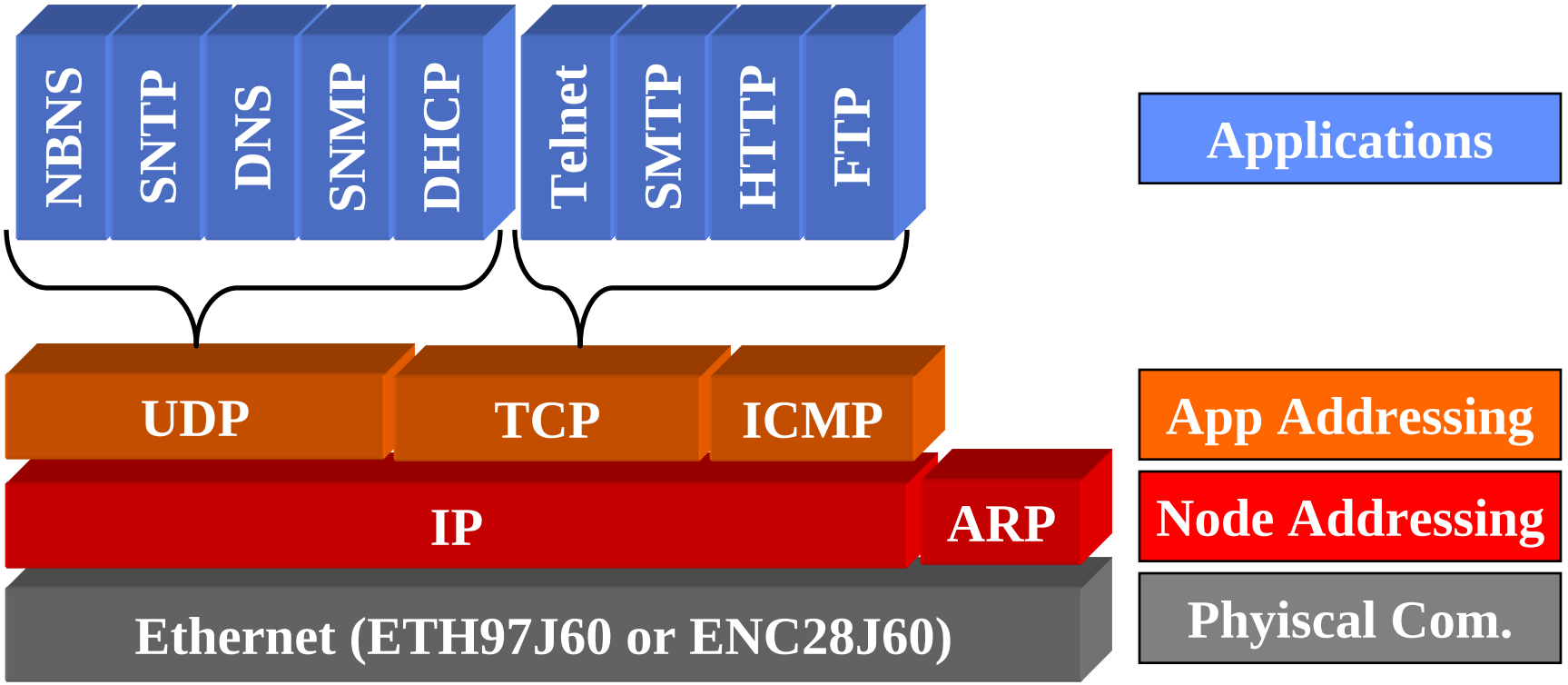
Cooperative Multitasking

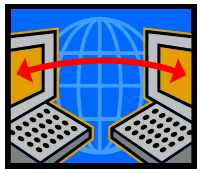
- **No task can block the processor**
  - Blocking will prevent stack from processing packets, lowering throughput
- **Break up long jobs into small chunks**
- **Use a state machine to keep track of position in UserTask()**
- **If nothing to do in UserTask(), return the extra cycles to the stack**

TCP and UDP

Stack Layers

Internet Protocol Stack





User Datagram Protocol (UDP)

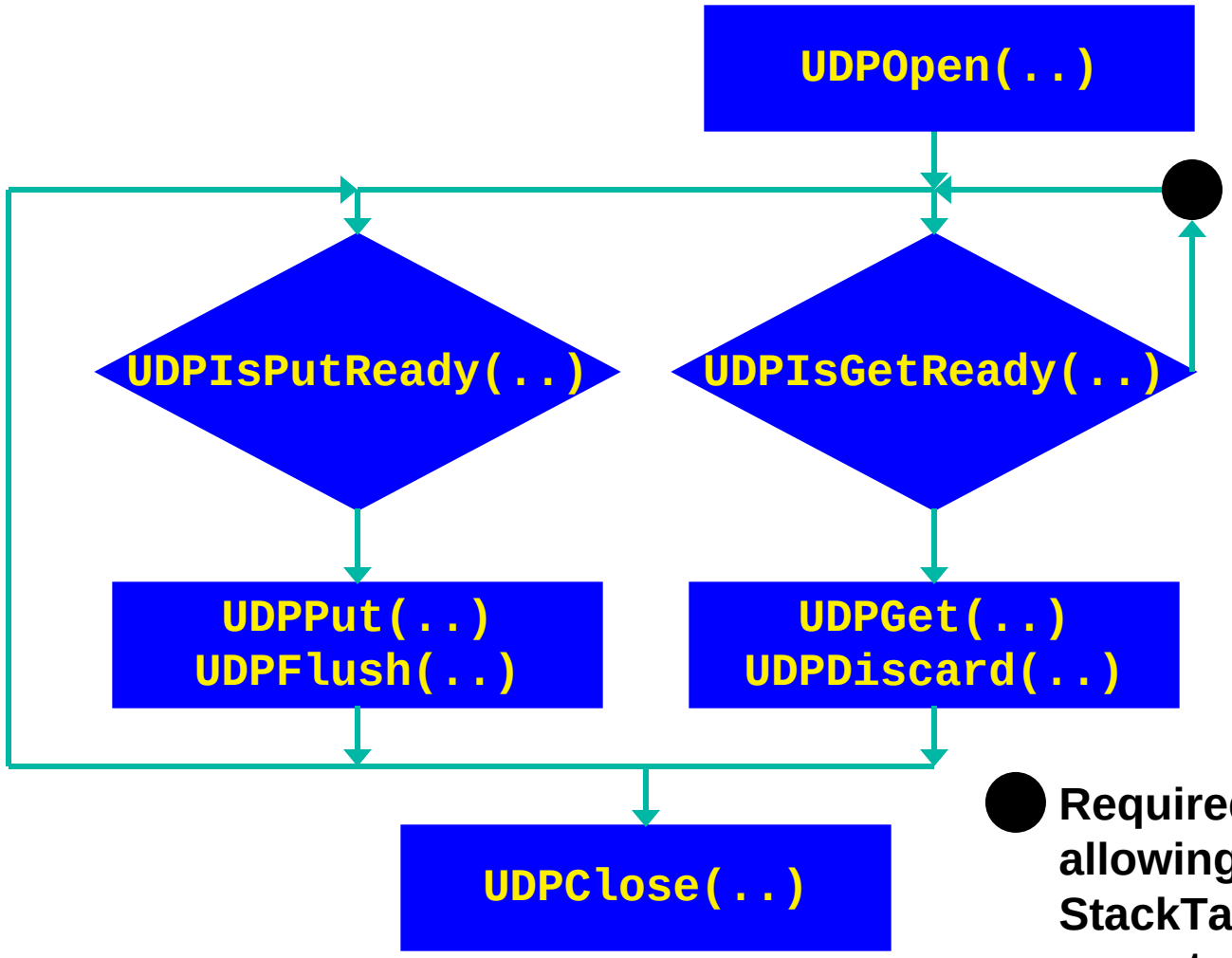
- **Provides high performance but unreliable communication to applications**
  - No connection set up overhead
  - Out-of-order, duplicate, and corrupted frames may arrive at application
  - Frames may get lost
  - Single packets can be multicasted or broadcasted to multiple nodes

User Datagram Protocol (UDP)

- **Datagram based**
  - Data sent from one application arrives at the destination application in a defined packet
  - Maximum recommended size is 512 bytes of application data

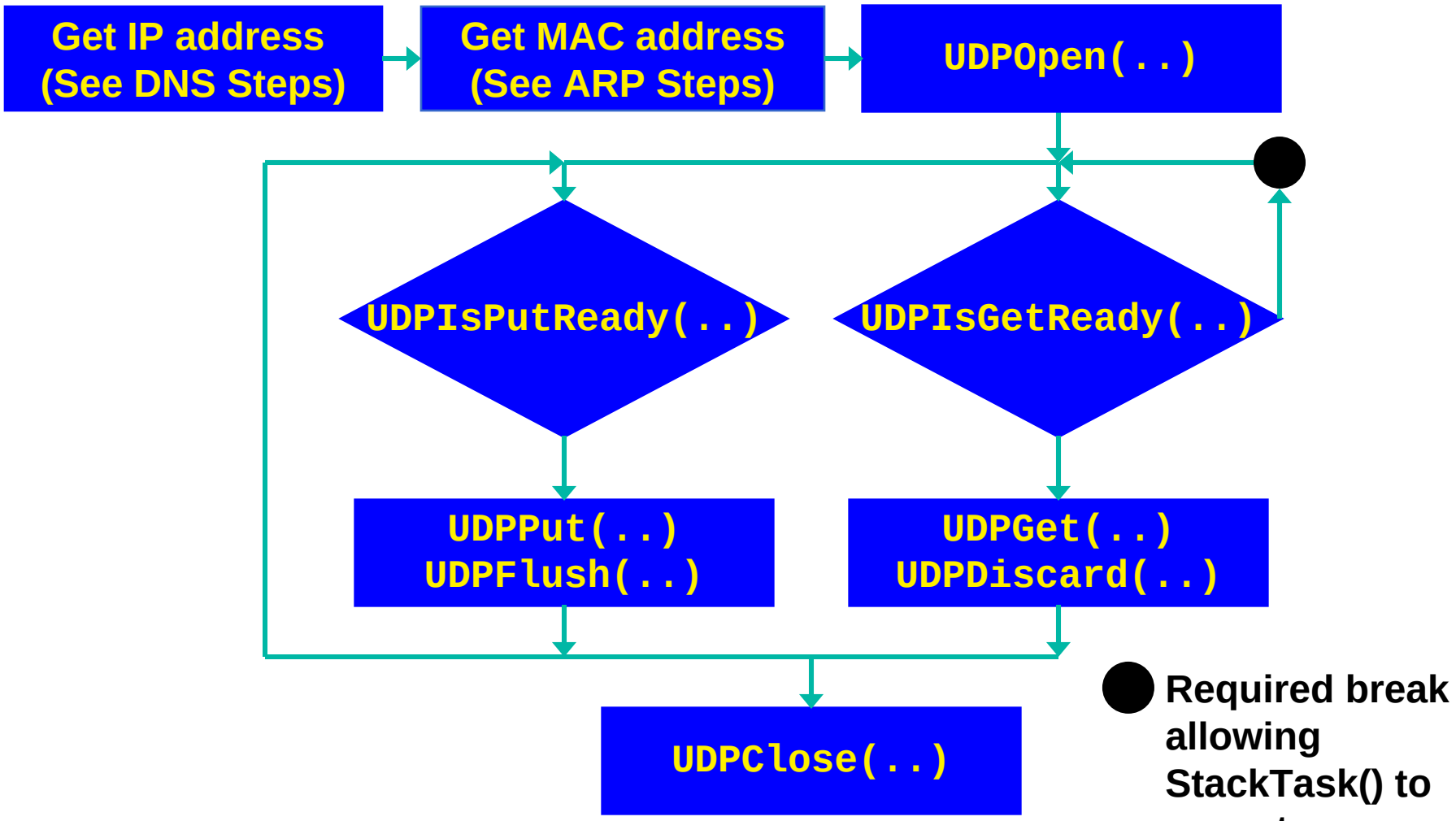
- **Microchip stack does not save UDP packets between task times**
  - Must save packet if needed later
  - Must build and transmit packets in one task time

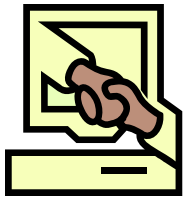
UDP Server Steps



● Required break allowing StackTask() to execute

UDP Client Steps





Transmission Control Protocol (TCP)

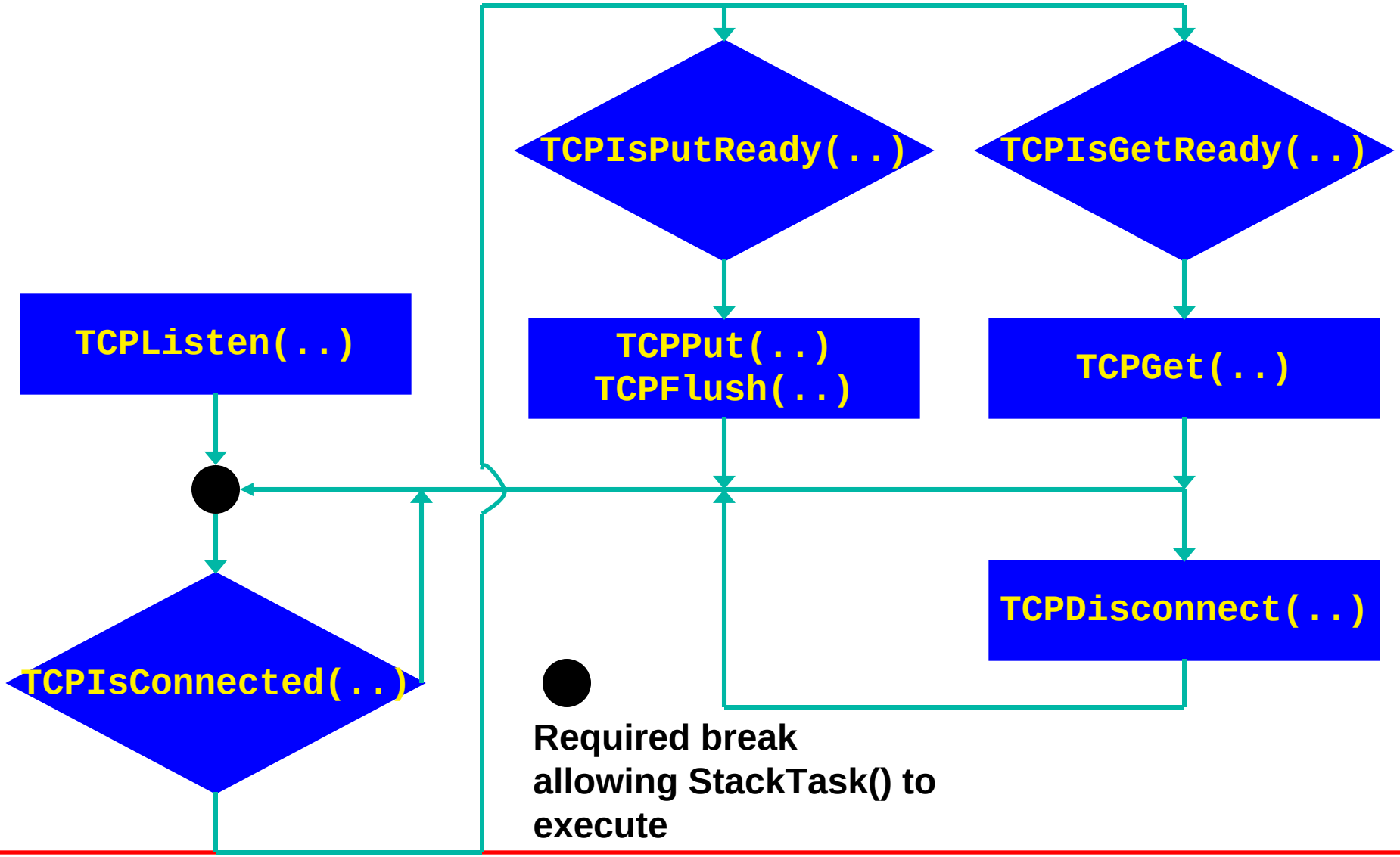
- **Provides reliable communication to applications**
  - Must establish a 1-to-1 connection to talk and receive
  - All data is acknowledged
  - Corrupted/lost data is automatically retransmitted
  - Duplicated data is discarded
  - Out-of-order segments are placed in correct order
  - Automatic flow control

Transmission Control Protocol (TCP)

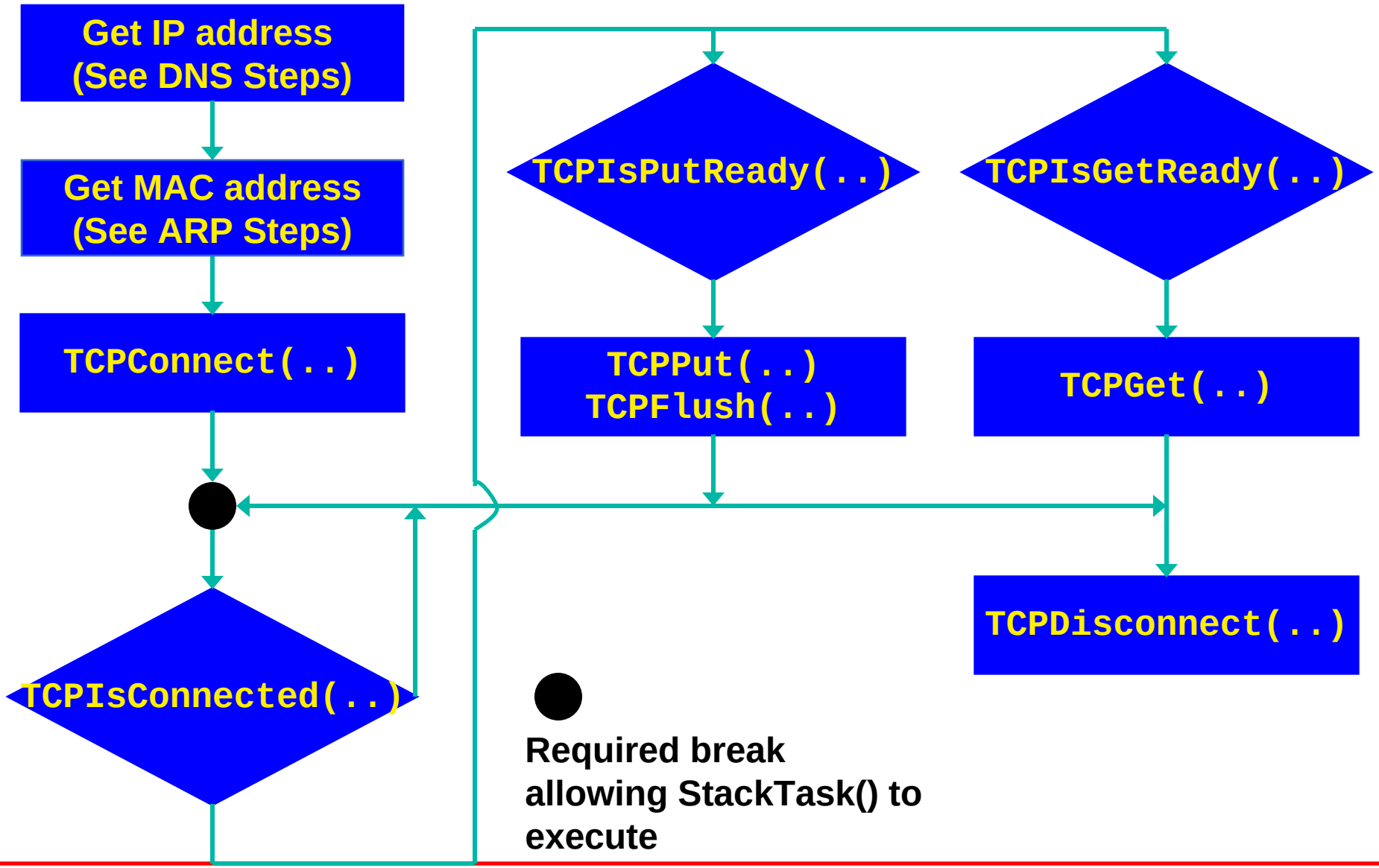
- **Stream based**

  - Routers may fragment/combine packets at random
  - Two adjacent application data bytes may arrive in two separate Ethernet packets at the destination
- **Microchip stack saves all TX & RX data for each application between task times**
  - Each socket gets its own FIFO buffers for transmitting and receiving
  - Performance drops substantially with increased round trip acknowledgement latency

TCP Server Steps



TCP Client Steps

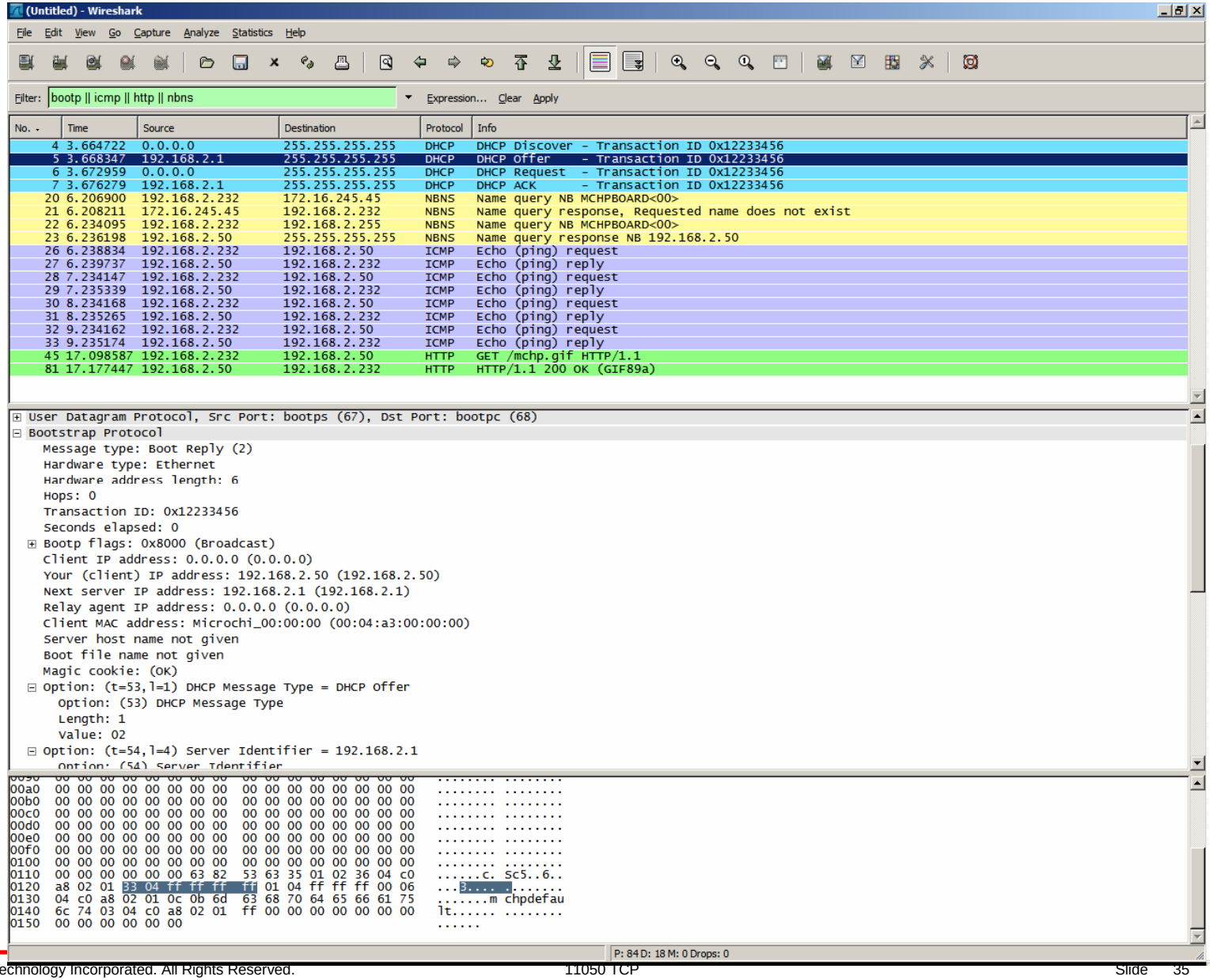


Take a look at it all

- **Wireshark captures and decodes Ethernet frames**
  - Free (GNU GPL license)
  - Multiplatform
  - Very large number of protocols and filters supported
  - Download from: http://www.wireshark.org/
  - Formerly called Ethereal

University of Microchip UofM MASTERs 2007

Wireshark



Wireshark

- **Hide unwanted traffic using proper filter:**
  - ip.addr == 192.168.2.101
    arp.dst.proto_ipv4 == 192.168.2.101 ||
    arp.src.proto_ipv4 == 192.168.2.101
- **Use the Expression builder to learn how to filter using other protocol fields**

Lab 2

- **Peek in with Wireshark:**
  - Get your PICDEM.net™ 2 up and serving web pages
  - Capture all traffic between your PC and board
  - Filter out junk traffic on the network
  - Right click and follow TCP stream to observe application communications
  - Find key pieces of information:
    - **PC's MAC address**
    - **Measure time it takes to download mchp.gif**
    - **Figure out what DHCP lease is obtained after reset without looking at LCD**

Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP)

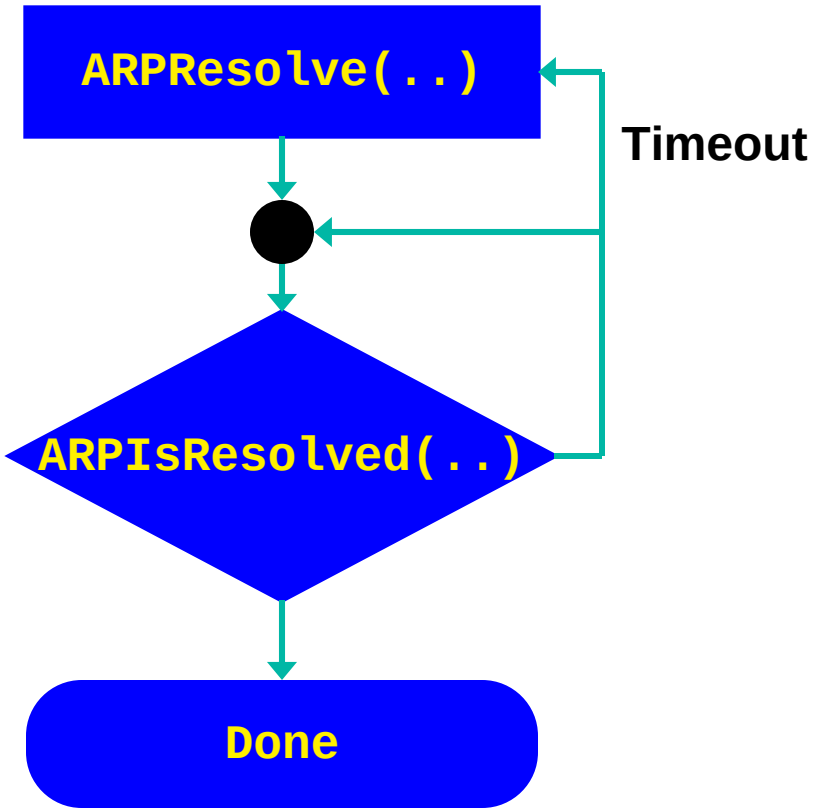
- **Translates IP addresses to MAC addresses**
  - Exposes API to IP or application protocol
  - Broadcast based
    - Broadcast: "Who has 192.168.1.123?"
    - Unicast reply: "I have 192.168.1.123 ; My MAC address is 00:12:34:00:00:01"
- **Required for Ethernet and 802.11 networks only**
  - PPP/serial links do not use MAC addresses

ARP Functions

- **Note: ARP functions may be unneeded in future stack versions**
  - IP address to MAC address translation can be done internally in the TCP and UDP modules so the application won't have to

- **ARPInit**
- **ARPProcess**
- **ARPResolve(…)**
- **ARPIsResolved(…)**

ARP Client Steps



● Required break allowing StackTask() to execute

Domain Name System (DNS)

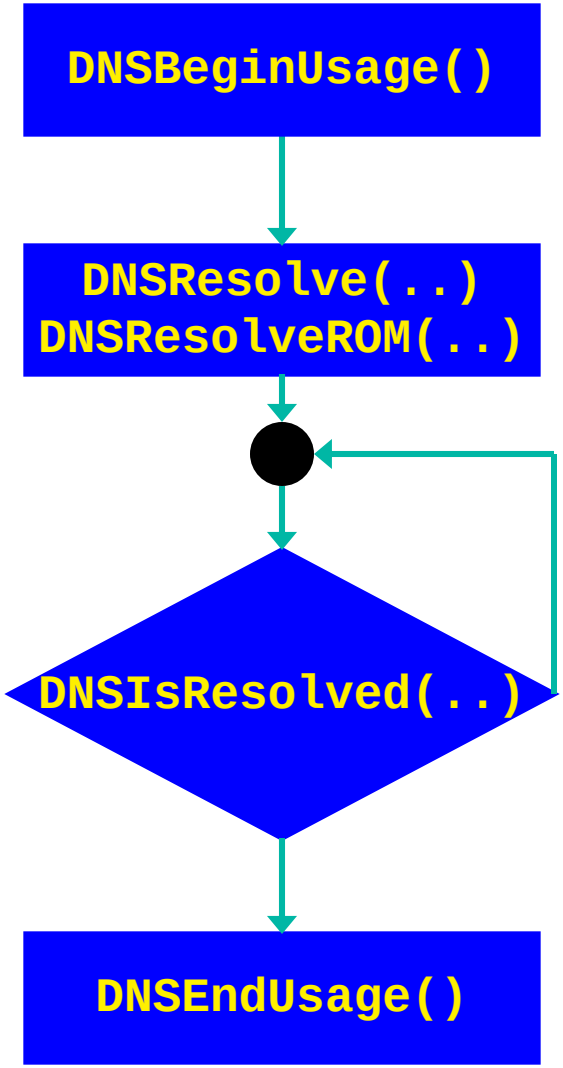
What is DNS?

- **Provides fully qualified host name to IP address translation**
  - Ex: "www.microchip.com" to 198.175.253.32
- **Provides addresses of SMTP servers associated with email addresses**
  - Ex: "support@microchip.com" to mx.microchip.com
- **Stack supports DNS client only**
- **Client automatically finds DNS servers via DHCP**

DNS Functions

- DNSBeginUsage
- DNSResolve(…)
- DNSResolveROM(…)
- DNSIsResolved(…)
- DNSEndUsage

DNS Steps



- Required break allowing StackTask() to execute

HTTP2 & MPFS2

HTTP2 Server

- Multiple simultaneous connections
- Supports HTML Forms
- Supports HTTP GET and POST commands
- Supports Cookies
- Supports Authentication
- Dynamic web page creation
- Pages stored in Program Memory or external non-volatile memory with MPFS2

Dynamic Variables

- **Escape sequence used to do dynamic substitution**
  - 1. Web browser requests `index.htm`
  - 2. Server begins transmitting `index.htm` to browser
  - 3. Each *~variable~* sequence encountered causes HTTP2 callback function in stack to execute
  - 4. Application returns dynamic values to browser in place of *~variable~*
  - 5. Remainder of `index.htm` is sent to browser

Dynamic Variables

- **~variable~ tags specify a file**
  - Ex: `~inc:header.inc~`
  - Whole file will automatically get returned to browser
  - Useful for applying global styles to all pages
- **Non-filename variables cause code to execute:**
  - Ex: ~myVariable~
  - HTTP2 server will call HTTPPrint_myVariable() when ~myVariable~ is reached while transmitting `index.htm`
- **Variables are pre-parsed at MPFS generation**
  - Improves performance

Form Submission

- **Browser form submission executes callback**
  - HTTPExecuteGet() for GET method
  - HTTPExecutePost() for POST method
- **Parameters available in myConn->data parameter**
- **HTTPGet[ROM]Arg() can help parsing**

Form Submission Example

- **1. Browser issues:**
  - GET /forms.htm?led1=0&led2=1 HTTP1.1
- **2. HTTPExecuteGet() executes**
  - myConn->data contains "led1\00\0led2\01\0"
- **3. User code calls HTTPGetROMArg() to obtain value associated with led1**
  - User code sets LED1 I/O pin to value to off
- **4. User code calls HTTPGetROMArg() to obtain value associated with led2**
  - User code sets LED2 I/O pin to value to on

Authentication

- **All files that start with a certain character require username and password**
  - Default character is ‘x’
- **Example**
  - `xSecurePage.htm` is protected
  - `InsecurePage.htm` is not protected
- **HTTPAuthenticate() callback specifies proper username(s) and password(s)**

HTTP2 Functions

Server Implemented

- HTTPInit
- HTTPServer
- HTTPGet[ROM]Arg
- HTTPURLDecode

User Implemented

- HTTPPrint_*
- HTTPExecuteGet()
- HTTPExecutePost()

Web Page Design Guidance

- **Hand-code the pages**
  - Or use appropriate visual web authoring tool
  - Microsoft® Visual Web Developer Express Edition™ is free and potentially suitable
- **Try to shrink the graphics**
  - Use correct file formats
  - **PNG** and **GIF** are usually smaller than **JPG** for images in embedded devices
- **Auto refresh dynamic content only**

MPFS2

- **Provides uniform file retrieval between internal program memory or external SPI EEPROM/Flash**
  - External SPI EEPROM can be updated through MPFS2.exe PC GUI tool
  - Internal program memory is updated by recompiling project and programming
- **PC GUI tool source code available for rebranding and distribution with your application**
  - Still must use PIC® microcontroller

University of Microchip UofM MASTERs 2007

Lab 3 – HTTP2

- **Use MPFS2.exe tool**
- **Add extra LED output to `dynvars.htm`**
  - Update `CustomHTTPApp.c`
- **Add extra LED form input to `forms.htm`**
  - Update `CustomHTTPApp.c`
- **Update authentication credentials and password protect `upload.htm`**
  - Update `CustomHTTPApp.c`

Simple Mail Transfer Protocol (SMTP)

What is SMTP?

- **Provides ability to transmit Internet email messages**
  - Can be sent to cell phone via SMS bridging
- **Microchip TCP/IP stack implements an SMTP client only**
  - Embedded device can send email, not receive it
  - Receiving mail requires other protocols, ex: POP3 or IMAP

SMTP Client

- **Message headers in ROM or RAM**
- **Message body in RAM or created on-the-fly**
- **Requires about 5 KBytes of ROM, 100 bytes of RAM**
- **Uses external SMTP server (proxy), or**
  - Can try to send emails directly to the destination mail server by doing a Mail eXchanger (MX) record look-up in the DNS.
- **Supports authentication (clear text)**

SMTP Functions

- **SMTPClient Structure**
  - Server
  - Username
  - Password
  - To, CC, BCC
  - From
  - Subject
  - Body
- **SMTPBeginUsage**
- **SMTPEndUsage**
- **SMTPSendMail**
- **SMTPIsBusy**
- **SMTPIsPutReady**
- **SMTPPut(…)**
- **SMTPPut[ROM]Array(…)**
- **SMTPPut[ROM]String(…)**
- **SMTPFlush**
- **SMTPPutDone**
- **SMTPTask**

SMTP Steps

- See `SMTP Application Flow.pdf` in Microchip Solutions\Microchip\TCPIP Stack

Lab 4 – Send email to yourself

- **Goal:**
  - Send yourself two emails, one containing the state of the buttons (in RAM), and one containing the state of the PIC® MCU's RAM.
  - See lab handout for required SMTP server

Internet Bootloader

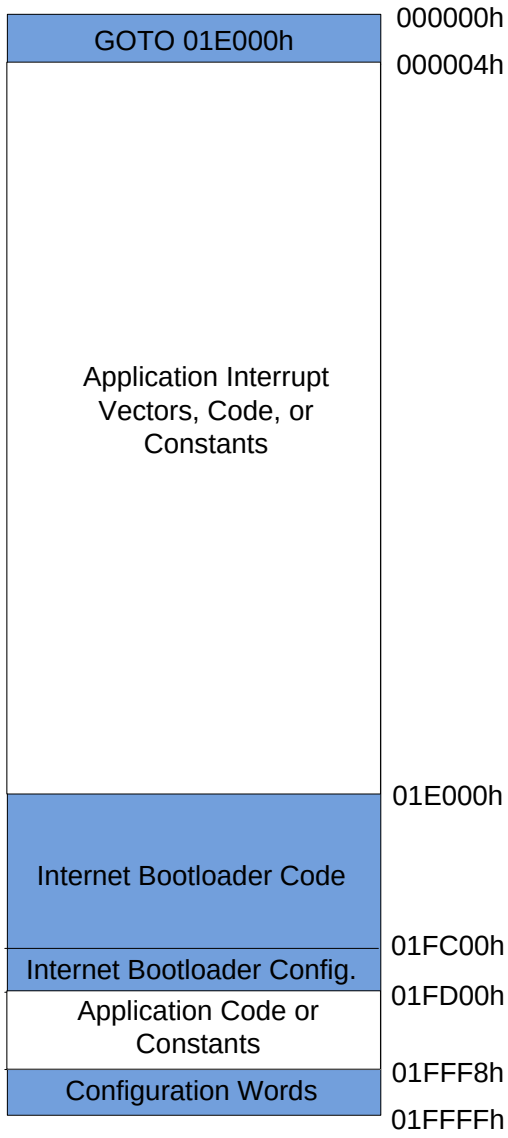
Bootloader

- **Implements its own private UDP/IP stack**
  - 8 KB of Flash on PIC18F97J60 family
- **Executes on Power-on Reset**
  - Waits 3 seconds before starting main application
  - Bootloader packets arriving within the 3 second window can begin reading and writing to program memory
- **Updates configuration fuses**

Bootloader

- **Does not modify interrupt vectors**
- **Executes outside application**
  - Effectively uses no RAM
- **Very easy to link application**
- **PC GUI interface with source code provided**
  - DLL also available for raw reading and writing to addresses without a GUI

Bootloader Memory Map



Bootloader

- **Limitations**
  - Bootloader cannot update itself
  - Cannot use watchdog timer postscalars of 1:1 and 1:2 unless enabled via firmware
  - Bootloader forces HS or HS+PLL clock mode to prevent loss of bootloader clock

Dynamic Host Configuration Protocol (DHCP)

What is DHCP?

- **Provides nodes with IP address, subnet mask, gateway address, DNS server addresses, and other configuration parameters**
- **Supports full DHCP client**
  - Updates `AppConfig` structure
- **Supports single node DHCP server**

DHCP Server

- **Allows one client**
  - Useful for direct connection to PC or second embedded device
  - Operates simultaneously with DHCP client
- **Automatically disables itself if the DHCP client successfully contacts a different DHCP server**
- **Gives a lease of only 15 seconds**
  - Forces client to constantly renew
  - Prevents IP address from staying in use if authoritative DHCP server is present

NetBIOS Name Service (NBNS)

What is NBNS?

- **Provides host name to IP address translation**
  - Ex: “mchpboard” to 192.168.2.100
- **Limitations**
  - Only works on a single broadcast domain (same subnet)
  - Supports up to 15 characters

NBNS Functions

- **NBNSTask**
  - Operates as a passive server, listening for broadcasts
  - Replies to name queries only
  - Does not generate name queries
  - `MY_DEFAULT_HOST_NAME` is the only configuration option

Simple Network Time Protocol (SNTP)

What is SNTP?

- **Automatically find current date/time from Internet**
- **Many free servers available**
  - No configuration needed (**pool.ntp.org**)
- **Returns time as 64-bit integer**
  - High order 32 bits specify seconds since midnight January 1st 1970 (or other epoch)
  - Low order 32 bits are fractional seconds (233ps resolution)

SNTP Functions

- **SNTPGetUTCSeconds**
  - Warning: do not use for time difference calculations as date could go backwards on next SNTP update
  - Use **TickGet*()** API for higher than 1 second resolution and difference calculations
- **SNTPClient**

Other Modules

Tick

- **Provides accurate time keeping for non-blocking wait loops**
  - Instead of:
    - **while(i++ < 10000);**
  - Use:
    - **if(TickGet() – i < TICK_SECOND)**
      return;
- **Can be used as a real time clock**
  - Ticks increment on integer multiple of timer hardware input clocks

Telnet Server

- **Text based console interface**
  - Like web browsers, has ubiquitous implementations on PCs
  - Smaller than HTTP2 with MPFS2 for simple command and control
  - Optional authentication is done in clear text
- **Try executing URL:**
  - telnet://mchpboard/

SNMP Agent (Server)

- **Simple Network Management Protocol (SNMP) version 1 supported**
  - Implements traps
- **See AN870**
- **Example SNMP applications:**
  - Cable Modem
  - Network Printer
  - Routers
  - Uninterruptible Power Supply

UART 2 TCP Bridge (Server)

- **Copies data from a hardware UART module to TCP socket and visa versa**
- **UART interface implemented as high priority Interrupt Service Routine**
  - Full duplex at maximum performance
- **TCP interface implemented as cooperative state machine**

Announce (Server)

- Listens for UDP discovery requests from the Microchip Ethernet Discoverer PC tool
- Useful for finding the IP address of your board with DHCP
- Full Microsoft C# source code to PC tool provided
- Demonstrates UDP broadcasting

Generic TCP Client

- **Acts as an HTTP client**
  - Resolves IP address for www.google.com
  - Resolves MAC address for gateway
  - Opens a TCP client socket
  - Downloads Google Search page of "Microchip"
  - Outputs results on UART

Generic TCP Server

- **Acts as a ToUpper() server**
  - Converts each ‘a-z’ character to upprcase ‘A-Z’ and echos it back
  - Other characters echoed back as is

Lab 5 – misc.

- **Program using Bootloader**
- **Try using Telnet, SNTP, ICMP client and server, and DHCP server modules**
  - See handout

Frequently Asked Questions

Notes on security

- **SSL/TLS is coming**
  - Secures HTTP and SMTP modules
  - Will take a lot of RAM and ROM
  - Will take several seconds per connection on PIC18 devices
- **Custom applications with symmetric-only encryption may be more appropriate**
  - See AN953 and AN1044 for strong AES encryption code
- **Secure authentication can be done without encryption**
  - Use secure one-way hash (ex: MD5, SHA1) algorithms to perform a challenge-response
- **Keep your devices behind firewalls whenever possible**
  - If access over Internet needed, use a VPN

Questions

- **How fast is the stack?**
  - See TCPIP Stack Version.txt release notes
  - TCP max TX rate, ~1.4Mbps on PIC24H @ 40 MIPS
    - **Performance drops over Internet**
  - UDP max TX rate, ~3.1Mbps on PIC24H @ 40 MIPS

Questions

- **How big is the stack?**
  - >90 kbytes for all modules, debug optimization
  - >64 kbytes for all modules, full optimization
  - <20 kbytes for minimal UDP only stack, debug optimization
  - Stack is very modular so it depends on application
  - See compiler linker map file for your specific configuration

Wrap Up

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Summary

- Overview of Ethernet products and status
- Wireshark and TCP/IP Stack protocols
- Editing web pages to suit your application
- Experienced using SMTP, HTTP2, MPFS2, Telnet, Bootloader, SNTP, and other modules

Resources

- AN833 – The Microchip TCP/IP stack
- Microchip TCP/IP and Ethernet forum – http://forum.microchip.com/tt.aspx?forumid=173
- Direct Questions – http://support.microchip.com/

Tools Used In This Class

- **Microchip TCP/IP stack**
  - Download from http://www.microchip.com/tcpip
- **MPLAB® IDE and MPLAB C18 compiler**
  - Download from http://www.microchip.com
- **Wireshark network packet sniffer and protocol analyzer**
  - Download from http://www.wireshark.org
- **Microsoft Visual Web Developer Express Edition**
  - Download from http://msdn.microsoft.com/vstudio/express
- **Hardware**
  - Purchase from http://www.microchipdirect.com or Development Tools store at discount
    - **PICDEM.net™ 2 development board (DM163024)**
    - **MPLAB ICD 2 debugger (DV164005)**
    - **9V power supply (AC162039)**

Thank You!
Questions?

References

- AN833 – The Microchip TCP/IP stack
- AN1044 – Data Encryption Routines for PIC24 and dsPIC® DSC Devices
- AN953 – Data Encryption Routines for the PIC18
- PICDEM.net™ 2 Users' Guide
- Microsoft Visual Studio Express Edition tools – http://msdn.microsoft.com/vstudio/express/
- Internet RFCs – http://www.faqs.org/rfcs/ or alternate sources

Appendix

University of Microchip UofM MASTERs 2007

TCP Functions

- TCPInit
- TCPListen(…)
- TCPConnect(…)
- TCPIsConnected(…)
- TCPDisconnect(…)
- TCPIsPutReady(…)
- TCPPut(…)
  TCPPut[ROM]Array(…)
  TCPPut[ROM]String(…)
- TCPFlush(…)
- TCPIsGetReady(…)
- TCPGet(…)
  TCPGetArray(…)
- TCPProcess
- TCPTick

TCP Application Examples

- **World wide web (HTTP/HTTPS)**
- **Email (SMTP/POP3)**
- **File and print servers**
- **Remote control (Telnet/SSH)**
- **Turn based multiplayer games**
  - Board games
  - Card games

UDP Functions

- UDPInit
- UDPOpen
- UDPClose
- UDPIsPutReady(…)
- UDPPut(…)
  UDPPut[ROM]Array(…)
  UDPPut[ROM]String(…)
- UDPFlush
- UDPIsGetReady(…)
- UDPGet(…)
  UDPGetArray(…)
- UDPDiscard
- UDPProcess

UDP Application Examples

- **Real time multiplayer games**
  - First person shooters
  - Real time strategy
  - Action/adventure
- **Voice over IP (VoIP) telephones**
- **Industrial automation**
- **VPN tunnels**
- **Some streaming audio/media protocols**
- **DHCP, DNS, SNMP, TFTP, NBNS, SNTP**

Blocking LCD

- **Interfaces to 16x2 character LCD**
- **Available API Functions**
  - LCDText[33] byte array
    - User accessible memory to write to
  - LCDInit
  - LCDUpdate
    - Copies LCDText[] into the LCD
  - LCDErase
    - Clears all characters in the LCD

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.