Internet Protocol

- Provides network addressing
  - Ensures data packets quickly reach the correct destination
- Versions
  - **Internet Protocol Version 4 (IPv4)**
    - Used on most networks
  - **Internet Protocol Version 6 (IPv6)**
- **Router**
  - Connects networks

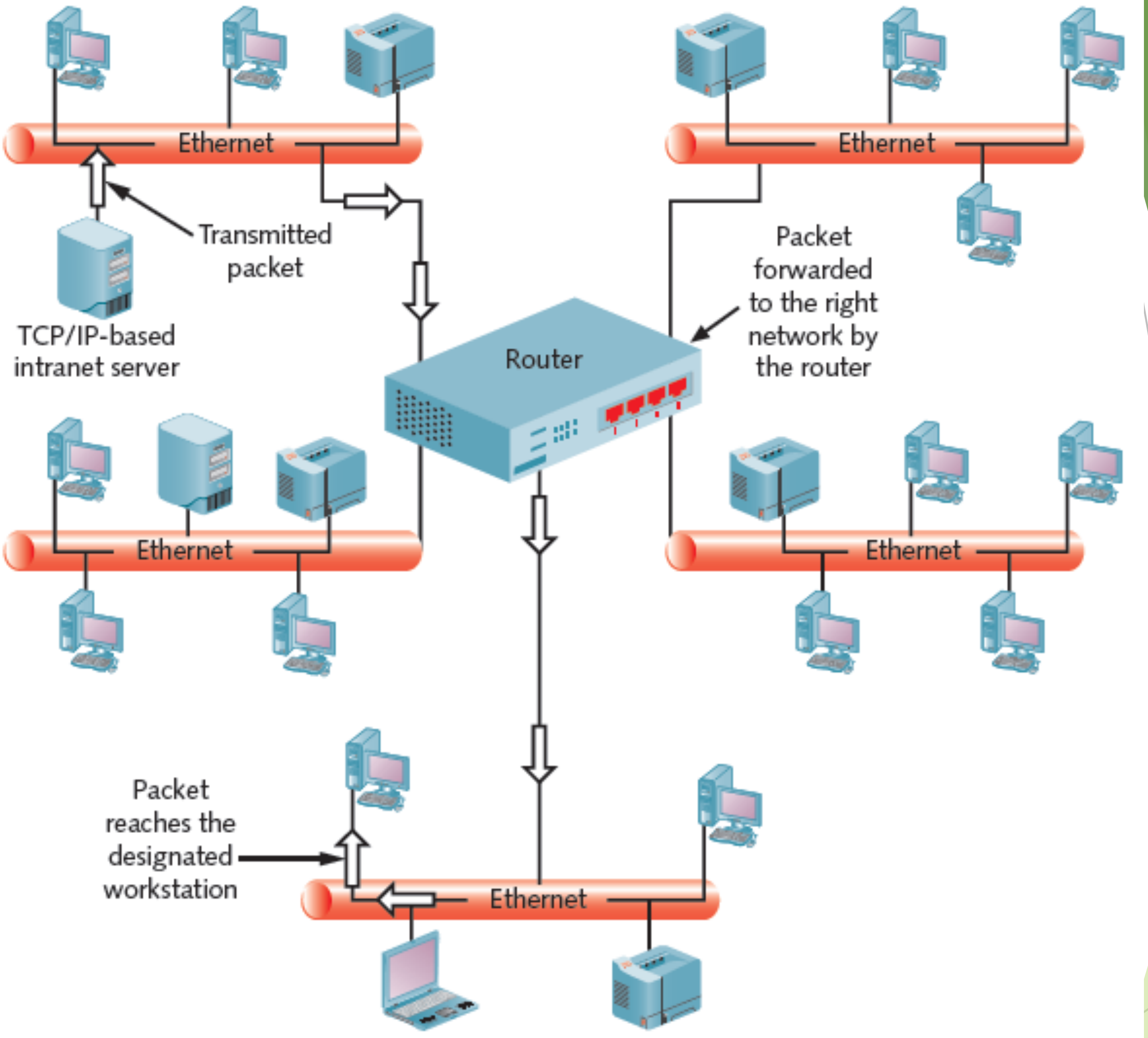

Ethernet
Transmitted packet
TCP/IP-based intranet server
Ethernet
Packet forwarded to the right network by the router
Router
Ethernet
Ethernet
Packet reaches the designated workstation
Ethernet

Figure 1-8 A router forwarding packets to a designated network
Courtesy of Course Technology/Cengage Learning

Internet Protocol (cont’d.)

- **IP addressing**
  - Dotted decimal notation
  - 32 bits long
  - Four fields
  - Example: 10000001.00000101.00001010.01100100 or 129.5.10.100
- **Unicast**
  - One packet is sent from a server to each client on request
- **Multicast**
  - Packet is sent to all clients as a group

Internet Protocol (cont’d.)

- **Broadcast**
  - Sends communication to all points on network
- **Subnet mask**
  - Used to show class of addressing and to divide network into subnets
- IP address considerations
  - Network number 127.0.0.0 cannot be assigned to any network
  - Private addresses reserved for **Network Address Translation (NAT)**
  - Cannot assign highest network number to a host

NY SYSTEMS

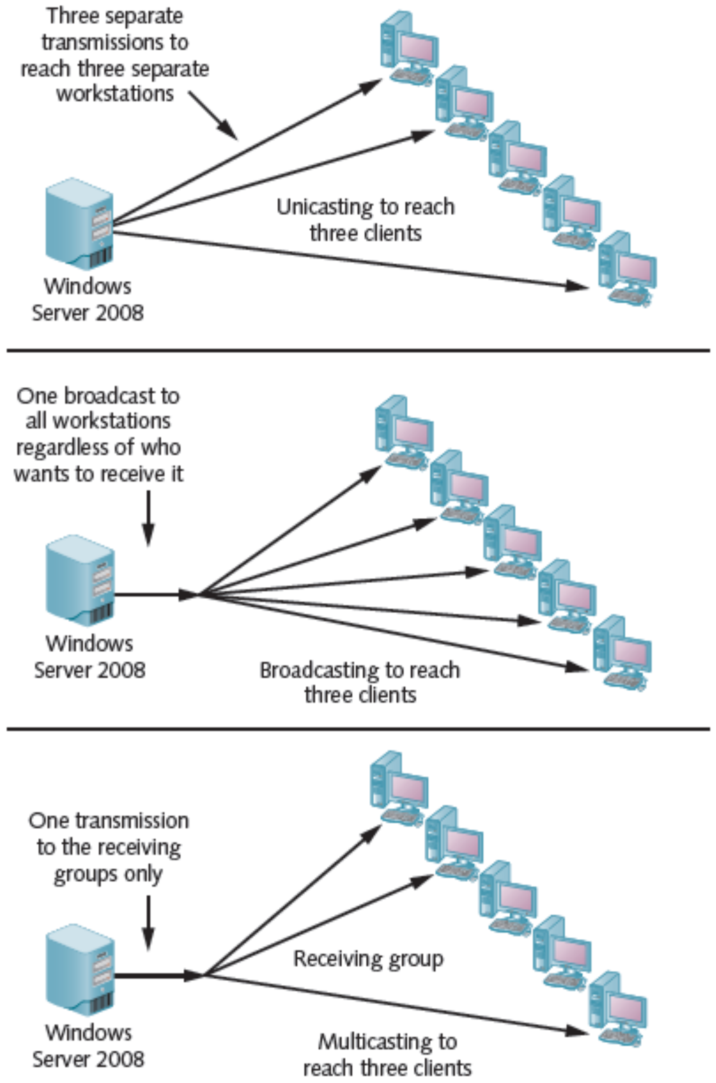


Figure 1-9 Unicasting, broadcasting, and multicasting
Courtesy Course Technology/Cengage Learning

Internet Protocol (cont'd.)

- Activity 1-4: Testing for IP Address and Connectivity
  - Objective: Practice using the Windows Server 2008 Command Prompt window with the *pathping* and *tracert* commands
- Internet Protocol version 6
  - Overcomes limitations of IPv4
  - 128-bit address capability
  - Single address associated with multiple network interfaces
  - IP extension headers

NY SYSTEMS

Internet Protocol (cont'd.)

- **Static addressing**
  - Assign permanent IP address
  - Gives consistency for monitoring
  - Can be laborious for large networks
- **Dynamic addressing**
  - IP address assigned during logon
  - Uses the **Dynamic Host Configuration Protocol (DHCP)**

NY SYSTEMS

Internet Protocol (cont’d.)

- **Default gateway**
  - IP address of the router that has a connection to other networks
- **Name resolution**
  - **Domain Name System (DNS)** translates domain and computer names to IP addresses
- **NetBIOS names**
  - **Windows Internet Naming Service (WINS)** server resolves NetBIOS names to IP addresses
- Host names
  - **Dynamic Domain Name System (DDNS)**

Physical Addresses and the Address Resolution Protocol

- **Address Resolution Protocol (ARP)**
  - Acquire the physical addresses associated with a computer’s network interface card (NIC)
- **Media access control (MAC) address**
  - Physical address of NIC
- TCP/IP relies on both IP addresses and MAC addresses
- Activity 1-5: Using Sample Utilities for IP Address and Connectivity Testing
  - Objective: Practice using the Windows Server 2008 Command Prompt window and ARP command

NY SYSTEMS

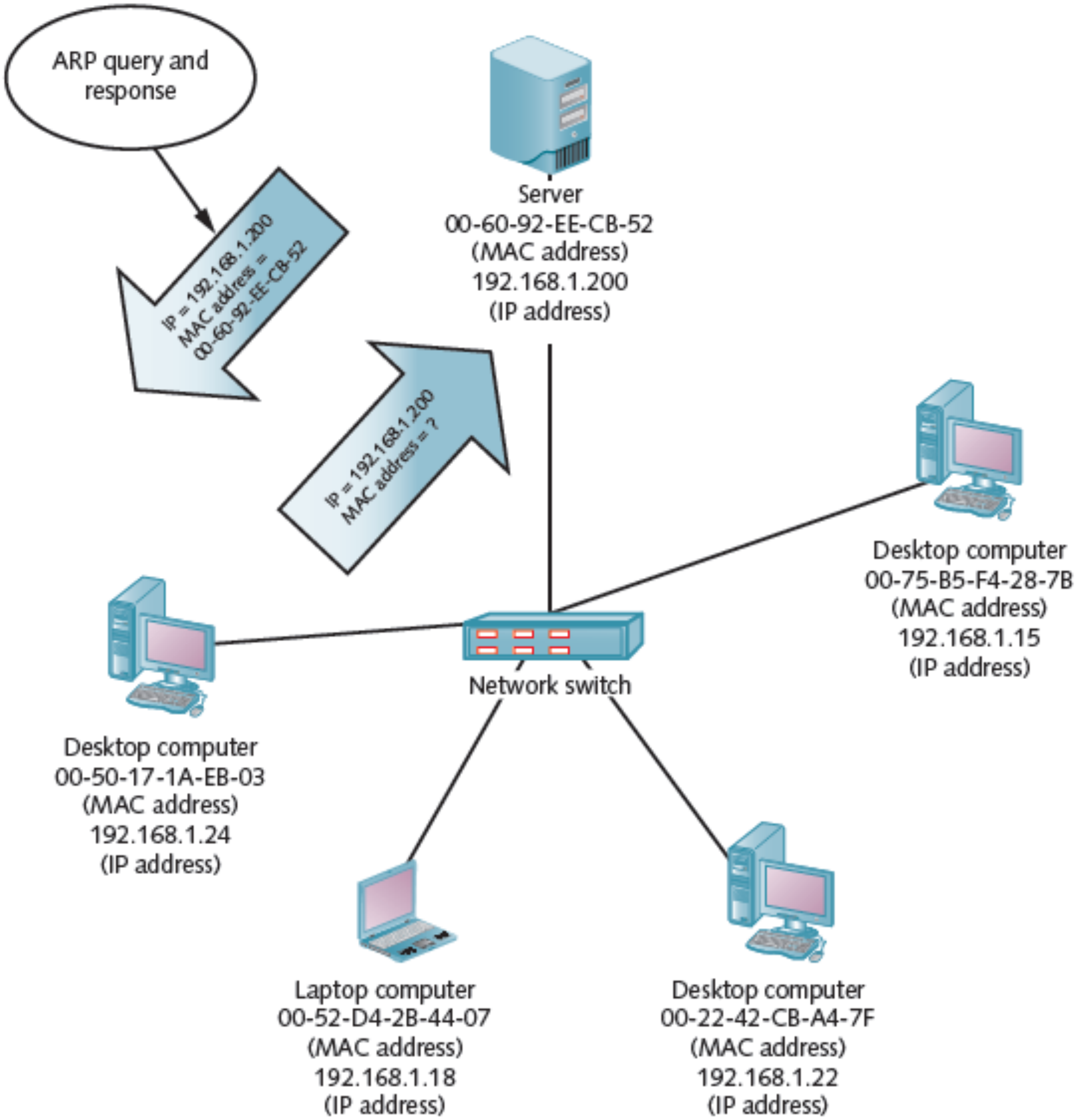


Figure 1-11 Using ARP to query the MAC address

Courtesy Course Technology/Cengage Learning

Implementing TCP/IP in Windows Server 2008

- Tasks
  - Verify TCP/IP enabled
  - Configure TCP/IP

Enabling TCP/IP

- TCP/IP
  - Only protocol installed by default when you install Windows Server 2008
- Activity 1-6: Verifying TCP/IP Is Enabled
  - Objective: Ensure that TCP/IP is enabled in Windows Server 2008

Configuring TCP/IP

- Choose static or dynamic addressing
- Activity 1-7: Configuring TCP/IP for Static Addressing
  - Objective: Learn how to manually configure TCP/IP for situations in which static addressing is used
- **Automatic Private IP Addressing (APIPA)**
  - Automated addressing through automatic private IP addressing
- Dynamic addressing through a DHCP server

Summary

- Eight editions of Windows Server 2008
- Features for security, clustering, virtualization, reliability, multitasking, and multithreading
- Peer-to-peer and server-based networking models
- TCP/IP
  - Default protocol for Windows Server 2008
  - IP addressing and versions
  - IP addresses can be statically or dynamically assigned

NY SYSTEMS

Microsoft .NET

Agenda

- .NET
- C#
- .NET vs. J2EE (C# vs. Java)
- Any .NET or C# programmers here?

Definition…

- “Microsoft .NET is a set of Microsoft software technologies for connecting information, people, systems and devices.”
- Microsoft’s explanation of .NET:

  http://www.microsoft.com/net/basics/whatis.asp

  - More of an emphasis on web services (self-describing self modules wrapped in Internet protocols (XML and SOAP)
- In real terms to the developer:
  - A new platform for building applications that run in stand-alone mode or over the Internet

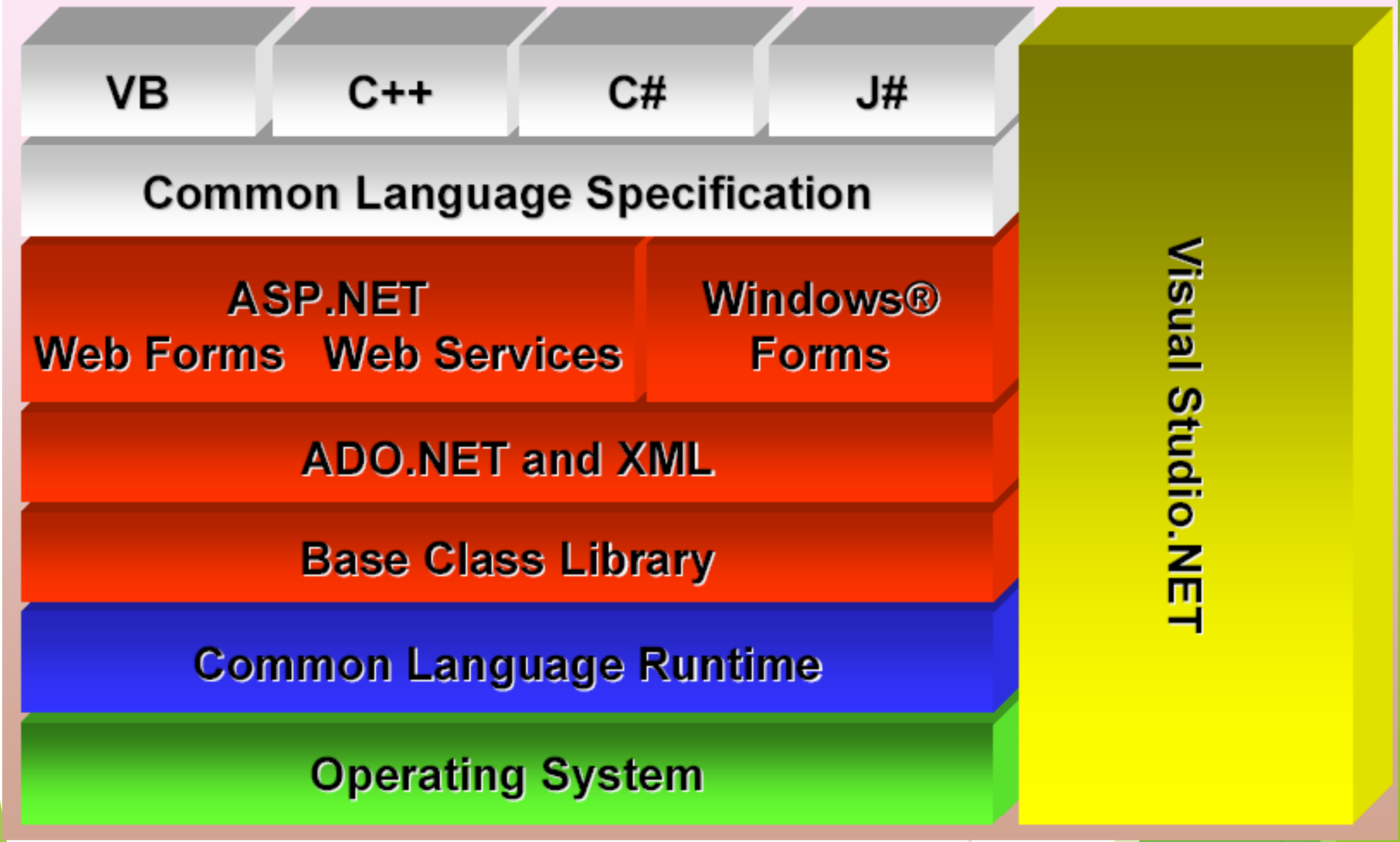
NY SYSTEMS

Evolution

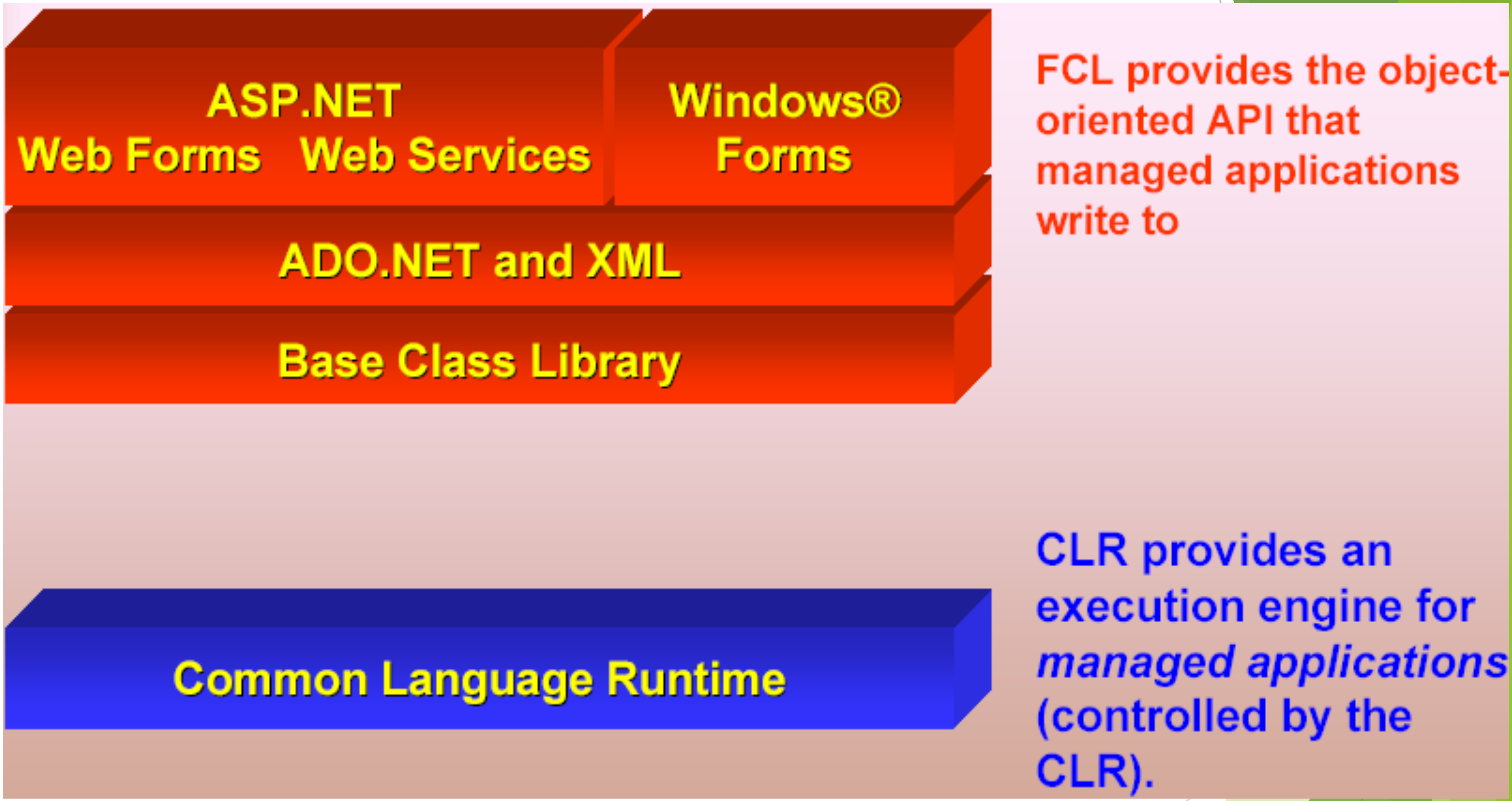
- Next Generation of COM:
  - Component oriented software:
    - Win32/C-style APIs are outdated
    - COM was step in right direction, but painful to program with
    - COM was restricted to VB, C++
    - Binary compatibility/portability an issue: x86 version of COM component needed to be compiled for e.g. PowerPC
    - Memory management also a pain
- Common Object Runtime:
  - An execution environment for components written in any language:
    - Eventually became .NET with incorporation of Web Services
    - Standardised API
- Web Services:
  - Interoperability is key in the connected world:
    - Require open standards for interoperability and leveraging legacy code

NY SYSTEMS

Architecture



.NET Core Components



FCL provides the object-oriented API that managed applications write to

CLR provides an execution engine for *managed applications* (controlled by the CLR).

- **FCL is Framework Class Library, comparable to JDK's library**

NY SYSTEMS

Java and .NET: Runtime environments

- Java
  - Intermediate language is *bytecode*
  - Original design targeted interpretation
  - Java VMs with JIT compilation are now also used
- .NET Framework
  - Intermediate language is *MSIL*
  - Provides JIT compilation
  - What is JIT?
  - Just-In-Time compilation: translates a bytecode method into a native method on the fly, so as to remove the overhead of interpretation

NY SYSTEMS

Common Language Runtime

- CLR sits on top of OS to provide a *virtual environment* for hosting managed applications
  - What is CLR similar to in Java?
  - Java Virtual Machine (JVM)
- CLR loads modules containing executable and executes their code
- Code might be managed or unmanaged
  - In either case the CLR determines what to do with it
- Managed Code consists of instructions written in a pseudo-machine language called common intermediate language, or IL.
- IL instructions are just-in-time (JIT) compiled into native machine code at run time