

Network Programming Part I

周新凯

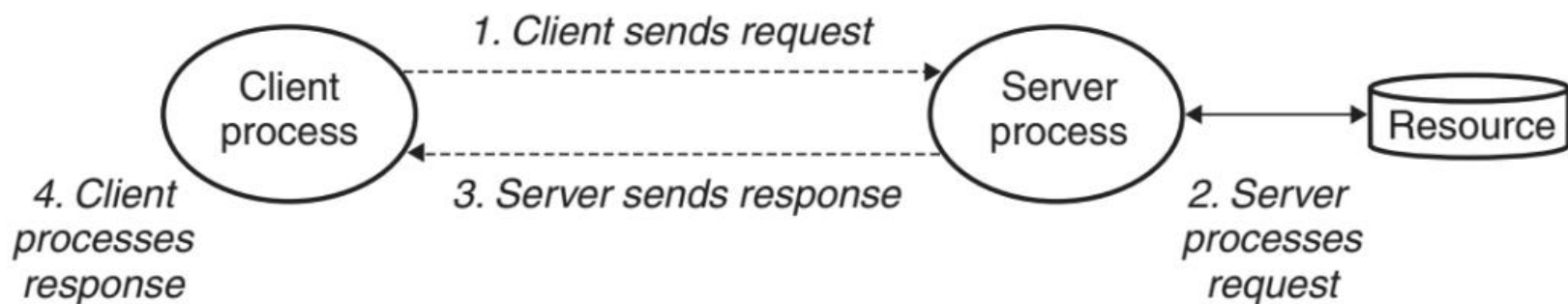
2024/12/11

1.The Client-Server Programming Model

每个网络应用都是依据**客户端-服务器**模型：一个应用由一个**服务器进程**和一个或多个**客户端进程**组成。

Tips: 客户端和服务端是**进程**而不是机器

一个**服务器**管理某种**资源**，并通过操作这种**资源**为**客户端**提供某种**服务**。



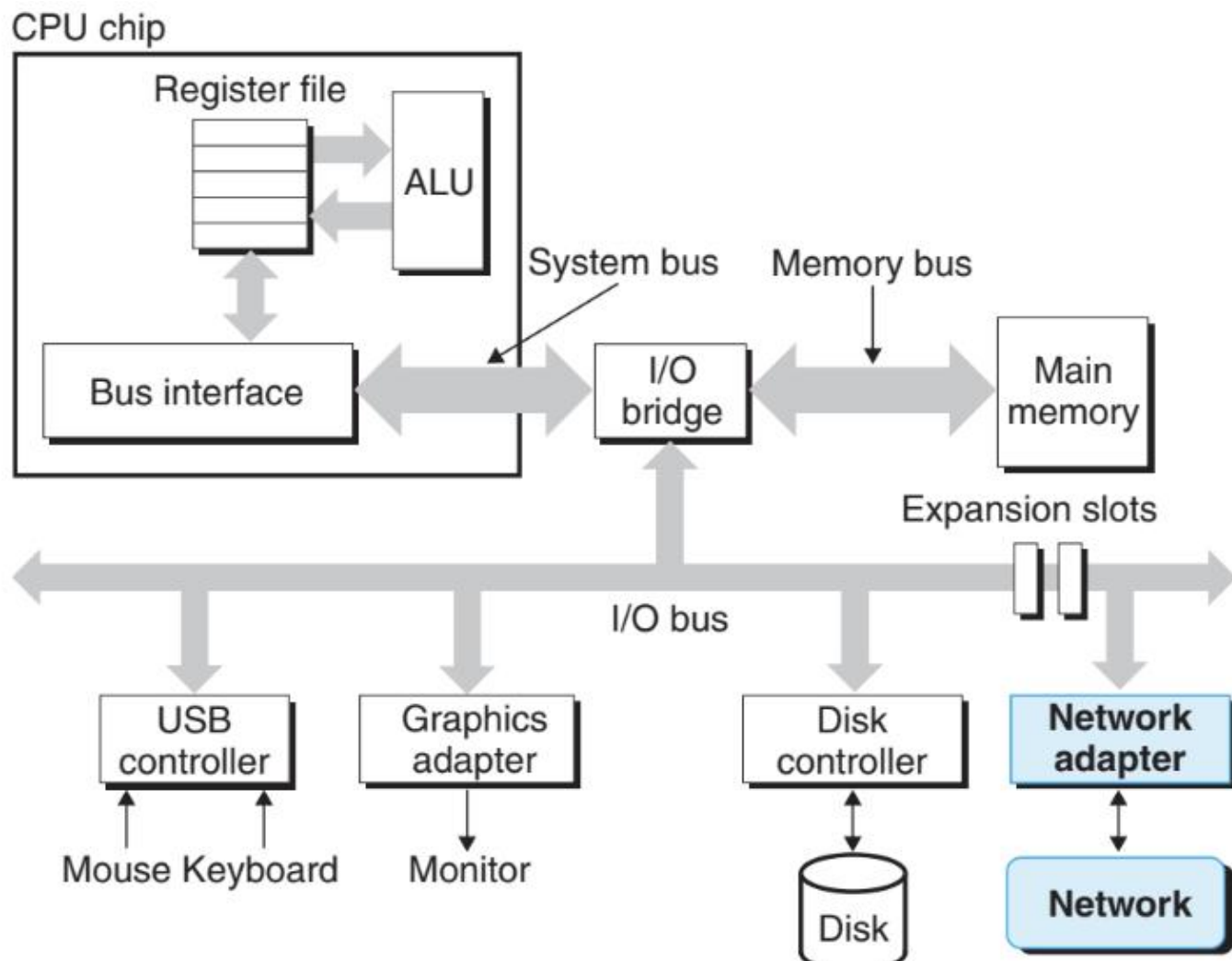
客户端-服务器模型的基本操作是事务 (transaction) :

1. 客户端需要服务时，通过向服务器发送请求来初始化一个事务
2. 服务器接收请求，解释它，然后以合适方式操作其资源
3. 服务器向客户端发送响应 (response) ，等待下一个请求
4. 客户端接收响应并处理

2.Networks

客户端和服务端通常运行在分开的主机（host）上，通过计算机网络（computer network）的硬件和软件资源来交流。

对于**主机**而言，其就是一种I/O设备，充当数据源和数据接收方。



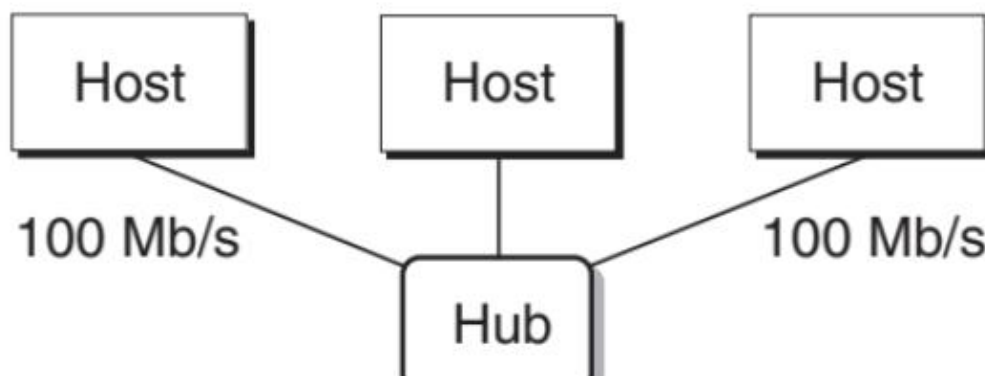
- 适配器 (adapter) 插在 I/O 总线的拓展插槽 (expansion slots) 中, 为网络 (network) 提供物理接口。
- 从网络接收的数据经过网络适配器, I/O 总线和内存总线最终复制到主存中, 即通过 DMA 传输方式。
- 数据也能从内存复制到网络。

从**物理**层面上说，**网络**是依据地理远近组成的**层次系统**：

- SAN： System Area Network 系统域网： 主机及外设
- LAN： Local Area Network 局域网： 以太网
- WAN： Wide Area Network 广域网： 因特网

2.1.Ethernet Segment

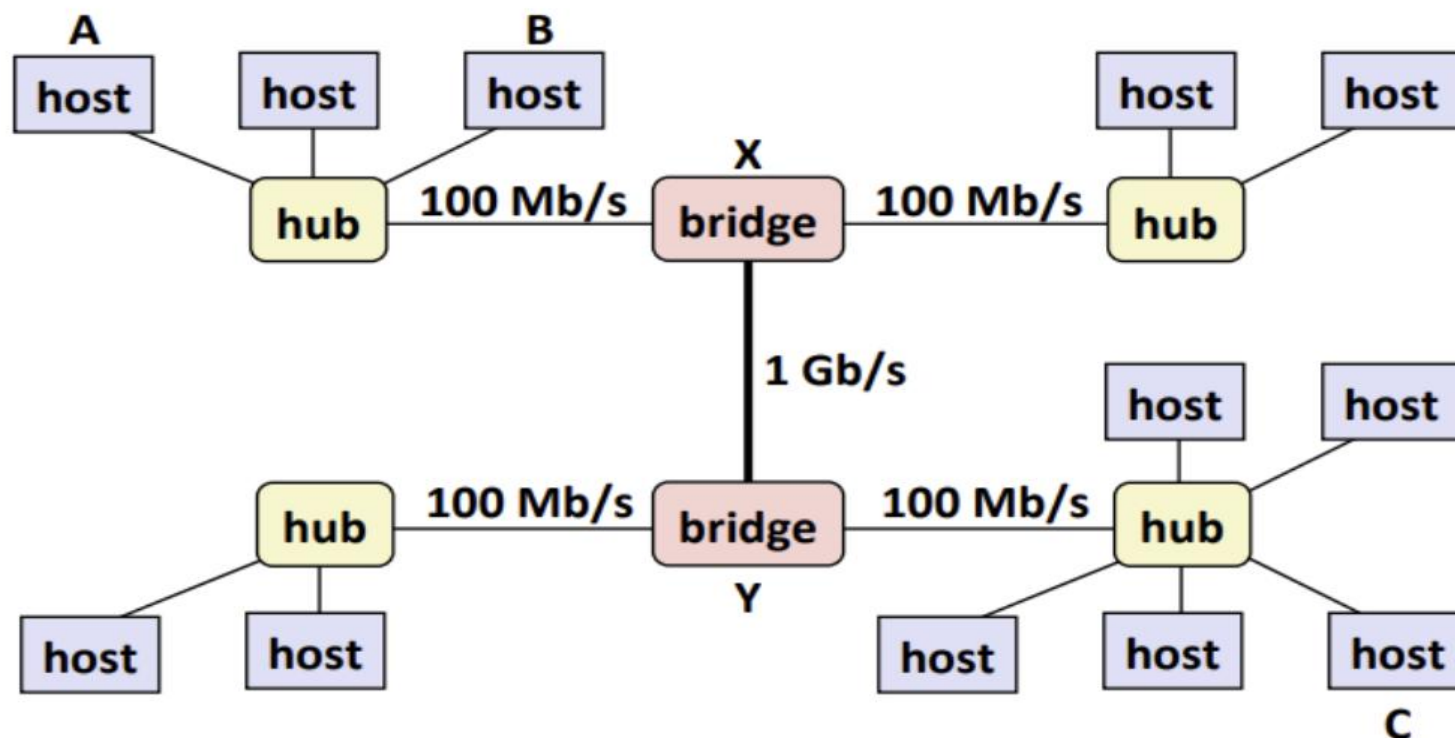
以太网段由一些**电缆**（通常为双绞线（twisted pairs of wires））和一个被称为**集线器**（hub）的小盒子组成，见下图所示：

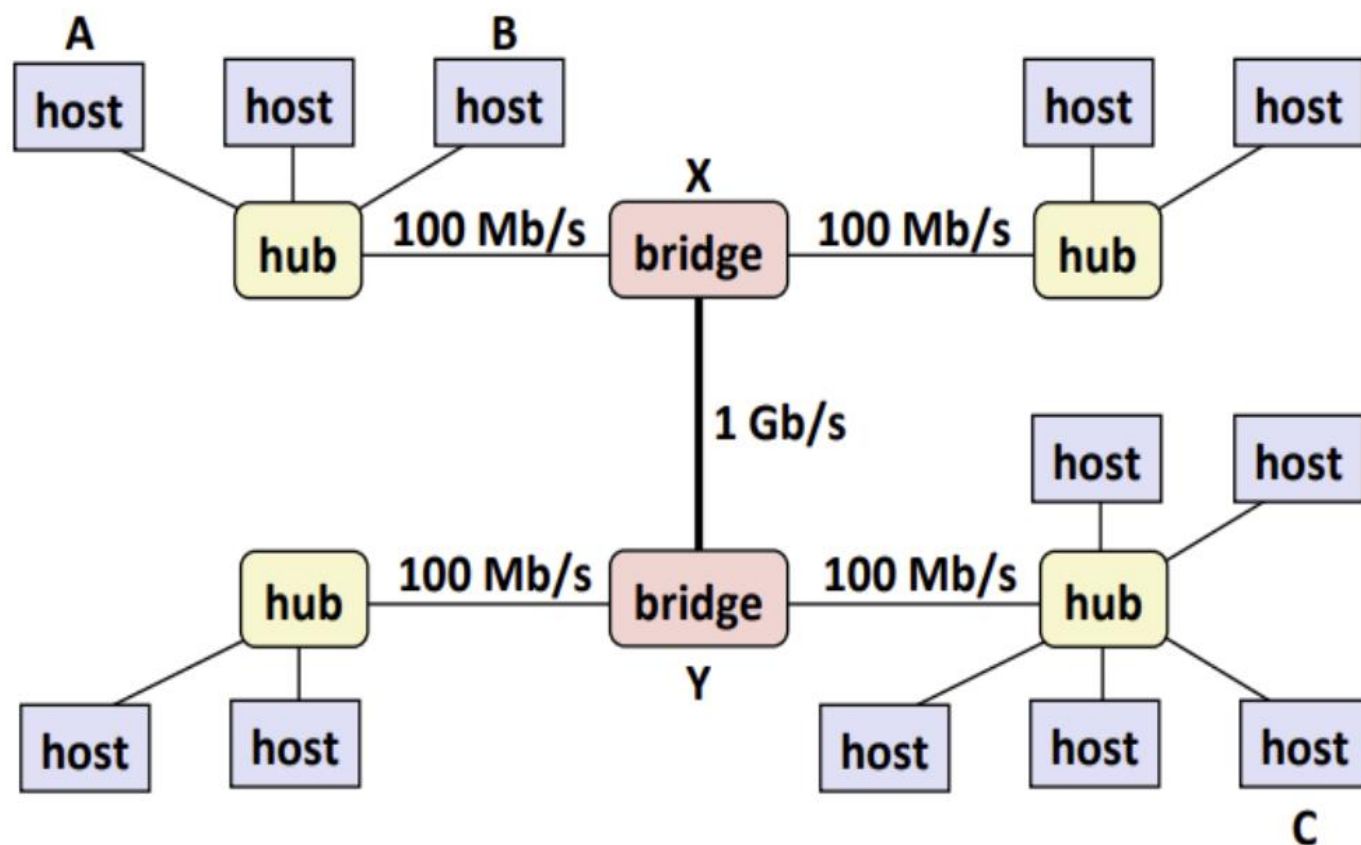


- **以太网段**通常覆盖的面积的范围很小，如一个房间或一栋建筑的一层。
- 每个**电缆**有相同大小的带宽（bandwidth），通常为 100 Mb/s 或 1 Gb/s。
- **电缆**的一端连接到一个**主机**的**适配器**上，另一端连到**集线器**的一个端口上。
- **集线器**会将每个**端口**接收到的**数据**会复制给**其他的所有端口**，因此与**集线器**相连的**每个主机**能看到集线器上传输的所有数据。
- 每个**以太网适配器**有一个全球唯一的 48 位地址（MAC address），该地址存在**适配器的非易失性存储器**上。

2.2.Bridged Ethernet Segment

通过一些**电缆**和被称为**网桥** (bridges) 的小盒子, 多个**以太网段**能连接成更大的**局域网** (LAN), 称为**桥接以太网** (bridged Ethernets) 。





网桥比集线器能更好的利用带宽，它**不会无差别的复制数据**给所有的集线器，而是通过**自学习**后有选择的复制数据到对应的端口。例如主机 A 发送帧给主机 B，则网桥 X 会丢掉该帧的数据；如果主机 A 发送帧给主机 C，则网桥 X 复制帧给网桥 Y，然后网桥 Y 只会将帧复制给主机 C 所在的集线器端口。

2.3.Internet

多个不兼容的局域网可以通过一种被称为**路由器**（router）的计算机连接起来形成**互联网络**（internet）。

- 每个**路由器**对于它所连接的每个网络都有一个适配器（端口）。
- 路由器**也能连接高速点到点电话连接，这是**广域网**（wide area networks）的例子。

2.4.Protocol Software

“互联网络能由采用完全不同和不兼容技术的各种局域网和广域网组成——如何发送数据?? ”

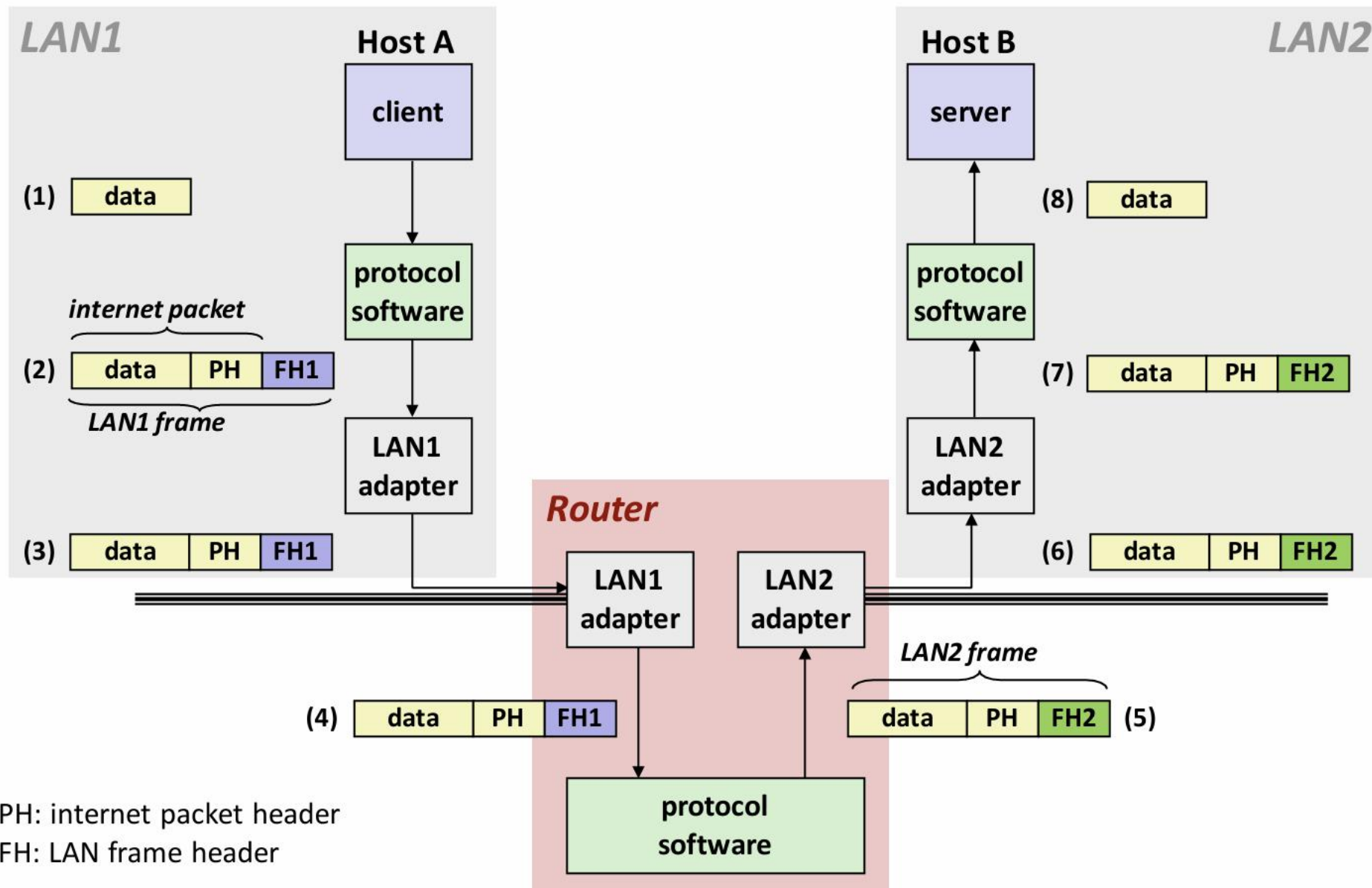
协议软件运行在每个**主机**和**路由器**上，通过一种**协议**（protocol）来处理**不同网络之间的差异**。

协议（protocol）：一系列规则，用来管理不同的**主机**和**路由器**在**网络之间如何传输数据**。

- 命名机制——Provides a naming scheme:
 - **互联网络协议**定义了一种一致的**主机地址格式**，以消除不同局域网技术在为主机分配地址上的差异。
 - 每台**主机**至少被分配一个**互联网络地址**（internet address），这个地址唯一地标识了这台**主机**。
- 传送机制——Provides a delivery mechanism:
 - **互联网络协议**定义了一种把数据位捆扎成不连续的片（称为**包**（packet））的统一的传送方式。
 - **包**由**包头**（header）和**有效载荷**（payload）组成：
 - **包头**：**包**的大小，**源主机**和**目的主机**的地址。
 - **有效载荷**：从**源主机**发出的**数据位**。

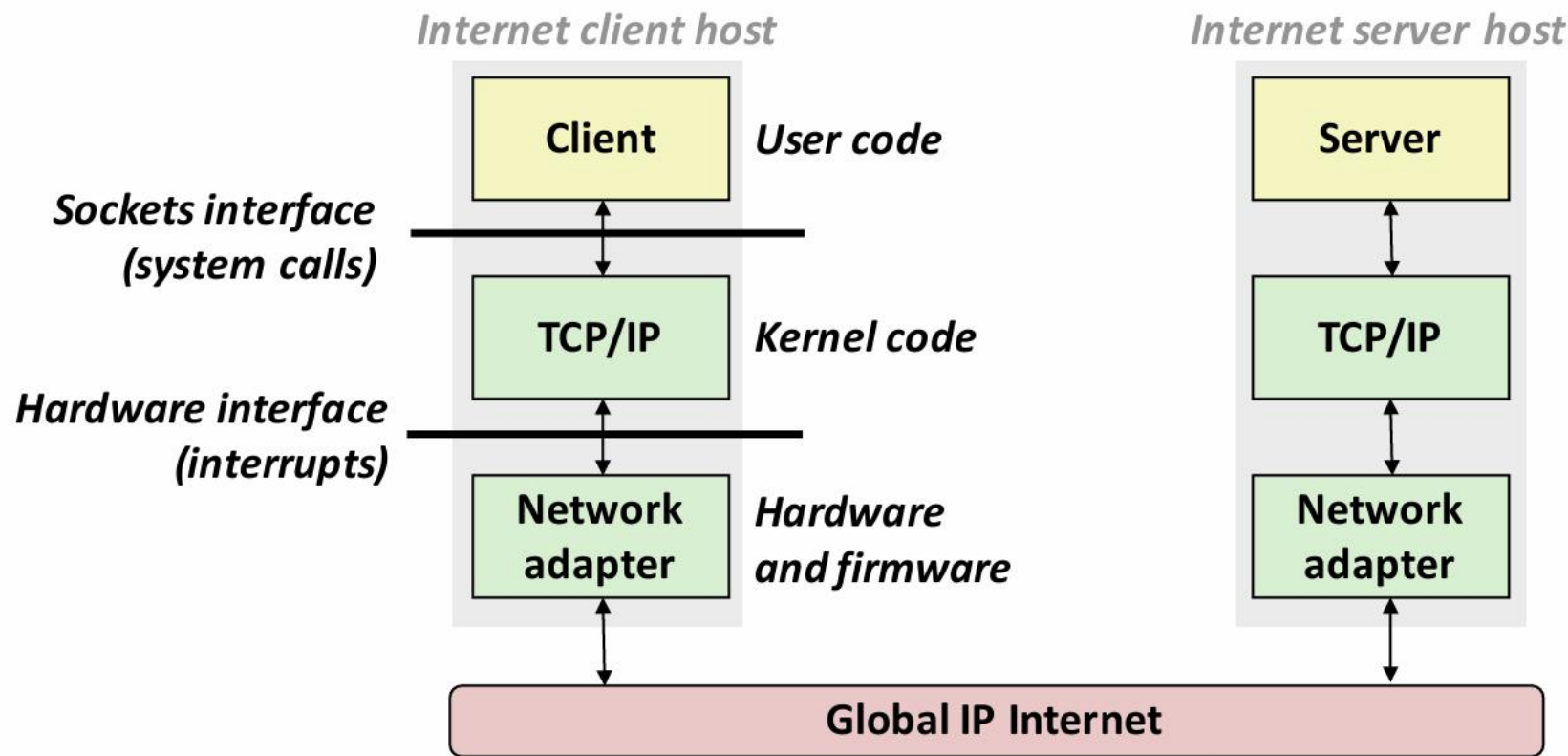
下图展示了**主机**和**路由器**怎么通过**互联网络协议**在不兼容的**局域网**之间传输数据：

PH: 记录B的ip地址
FH1: 记录路由器的
MAC地址
FH2: 记录主机B的
MAC地址



3.The Global IP Internet

如图为一个因特网客户端-服务器应用程序的基本硬件和软件组织：



3.1.IP Address

- IP 地址是无符号 32 位整数。
- TCP/IP 定义了统一的网络字节序，对于 IP 地址等整型数据采用**大端字节序**，即使主机字节序是小端法。
转换如下：

```
#include <arpa/inet.h>
```

```
uint32_t htonl(uint32_t hostlong);  
uint16_t htons(uint16_t hostshort);
```

主机字节序转换为网络字节序

Returns: value in network byte order

```
uint32_t ntohl(uint32_t netlong);  
uint16_t ntohs(uint16_t netshort);
```

网络字节序转换为主机字节序

Returns: value in host byte order

- IP 地址通常用**点分十进制** (dotted-decimal notation) 表示

- IP 地址**点分十进制**字符串和**二进制**网络字节序之间的转换函数

```
#include <arpa/inet.h>
```

```
int inet_pton(AF_INET, const char *src, void *dst);
```

Returns: 1 if OK, 0 if src is invalid dotted decimal, -1 on error

```
const char *inet_ntop(AF_INET, const void *src, char *dst,  
                      socklen_t size);
```

Returns: pointer to a dotted-decimal string if OK, NULL on error

函数名 _ 后面的 n 表示 network , p 表示 presentation , 这两个函数能操作 32 位的 IPv4 地址 (AF_INET) 或者 128 位的 IPv6 地址 (AF_INET6)

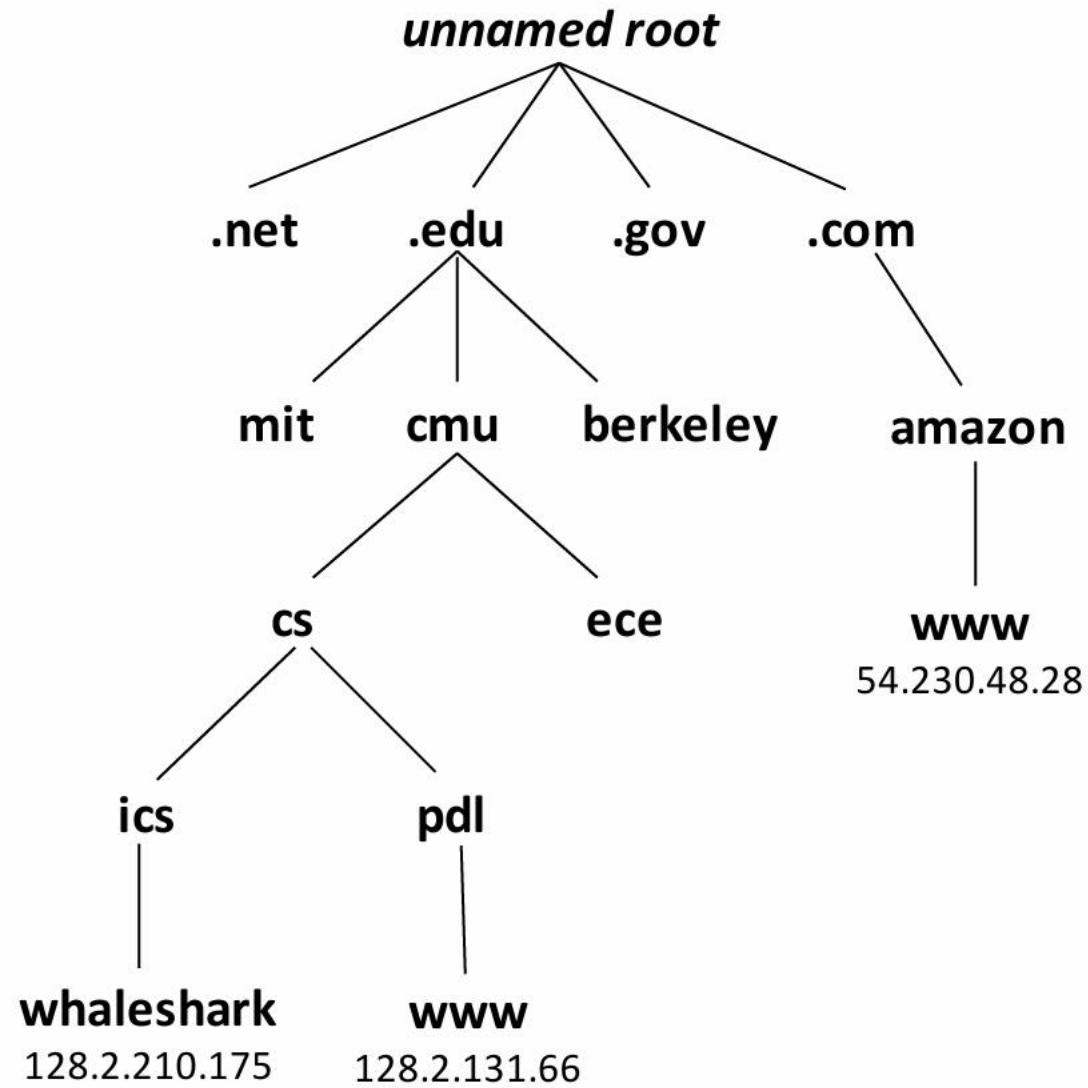
- inet_pton 将**点分十进制**的字符串 (src) 转换为**二进制**的网络字节序 (dst)
- inet_ntop 将**二进制**的网络字节序 (src) 转换为**点分十进制**的字符串 (dst)

3.2.Internet Domain Names

互联网定义便于记忆的**域名**（domain name）来代替难记忆的 IP 地址，并通过域名系统 DNS（domain name system）来管理域名和 IP 地址之间的映射，DNS 是一个**分布式的数据库系统**。

域名采用层次树状结构来命名，不区分大小写

每个主机都有一个本地定义的域名 localhost，总是映射为**回送地址**（lookback address）127.0.0.1。



First-level domain names

Second-level domain names

Third-level domain names

3.3.Internet Connections

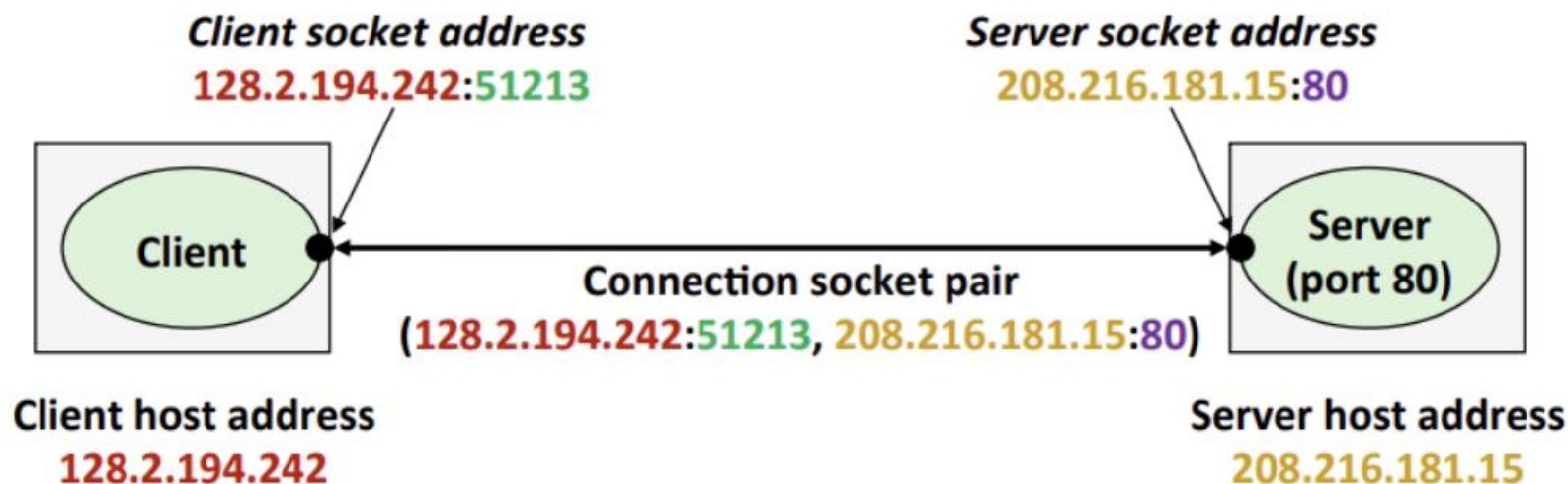
因特网链接从不同角度来说有三个特点：

- **点对点** (Point-to-point) ： 连接一对进程
- **全双工** (Full-duplex) ： 数据可以同时双向流动，同时发送和接收数据。
- **可靠** (Reliable) ： 源进程发送的数据基本能被目的进程按照相同的字节序接收。

连接的末端就是**套接字**（socket）。

- 套接字的地址格式为：**IP 地址：端口**。

- 连接由其两端的套接字地址唯一的确定，这一对套接字称为**套接字对**（socket pair），格式为（cliaddr:cliport, servaddr:servport），cliaddr 为客户端 IP 地址。



谢谢