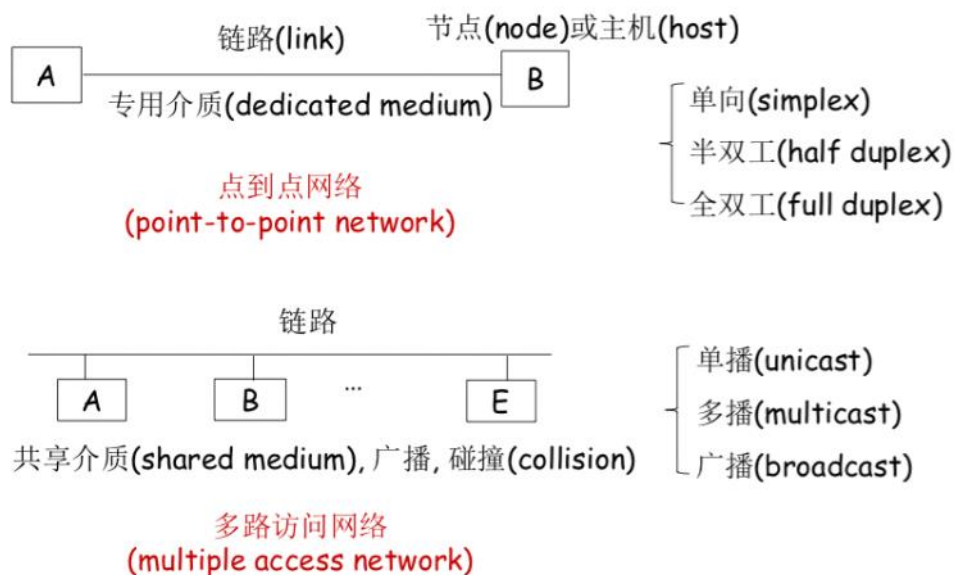


概述

2019年3月1日 星期五 上午12:13

直接连接的网络



多播：分组内一对多传播

广播：一对所有传播

间接连接的网络



有多少个直连网？ 4个

区分：

internet：用路由器或网关连接起来构成的网络称为互连网络

Internet：互联网，是互连网的一种

不同范围的网络：

系统域网(System Area Network) SAN

局域网(Local Area Network) LAN

城域网(Metropolitan Area Network) MAN

广域网(Wide Area Network) WAN

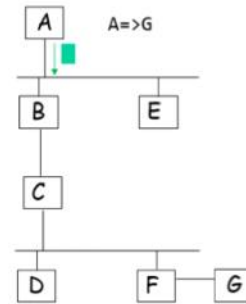
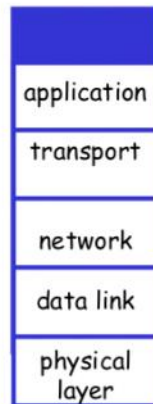
网络提供的服务：

- ❑ 可靠的服务
不可靠的服务
- ❑ 面向连接的服务
无连接的服务
- ❑ 有确认的服务
无确认的服务
- ❑ 数据报服务：无连接无确认
- ❑ 请求响应和消息流服务

- 主机、端系统：所有连接互联网的设备
- ISP：因特网服务提供商
- 协议：协议(protocol)是在网络实体(entities)之间传送消息的规则，例如， 消息的格式、收发消息的次序
- 分组：在端系统传输数据的时候，发送端将数据分段并为每一段加上首部字节，由此形成信息包。。这些分 组送到目的端后再被装配成初始数
- 分组交换机：从入通信链路接受分组再从出通信链路转发该分组，路由器 (router) 就是其中一种
- 路径：从发送到接受端系统，一个分组所经过的通信链路和分组交换机
- 路由器：路由器 (Router) 又称网关设备 (Gateway) 是用于连接多个逻辑上分开的网络，所谓逻辑网络是代表一个单独的网络或者一个子网。当数据从一个子网传输到另一个子网时，可通过路由器的路由功能来完成。因此，路由器具有判断网络地址和选择IP路径的功能，它能在多网络互联环境中，建立灵活的连接，可用完全不同的数据分组和介质访问方法连接各种子网，路由器只接受源站或其他路由器的信息，属网络层的一种互联设备。
- 全双工 full-duplex：通信允许数据在两个方向上同时传输
- 半双工 half-duplex：数据传输指数据可以在一个信号载体的两个方向上传输，但是不能同时传输
- 单工 simplex：数据传输只支持数据在一个方向上传输

分层的体系结构

- **应用层:** 提供对某些专门应用的支持
 - ❖ 文件服务(FTP), 邮件(SMTP), 网页(HTTP)
- **传输层:** 进程之间的数据传送 (端到端)
 - ❖ TCP, UDP
- **网络层:** 路由选择, 实现在互连网中的数据传送 (主机到主机)
 - ❖ IP, routing protocols
- **数据链路层:** 在物理网络中传送包 (跳到跳, 节点到节点)
 - ❖ PPP, Ethernet
- **物理层:** 线上的比特 (传送原始比特流)



每一层都可以多个协议, 每个协议都可以使用下层提供的服务并为上层提供服务。

- 协议栈: 各层中的所有协议
- 物理层:

虽然链路层的任务是将整个帧从一个网络元素移动到邻近的网络元素, 而物理层的任务是将该帧中的一个一个比特从一个结点移动到下一个结点。在这层中的协议仍然是链路相关的, 并且进一步与该链路 (例如, 双绞铜线、单模光纤) 的实际传输媒体相关。例如, 以太网具有许多物理层协议: 一个是关于双绞铜线的, 另一个是关于同轴电缆的, 还有一个是关于光纤的, 等等。在每种场合中, 跨越这些链路移动一个比特是以不同的方式进行的。

- 数据链路层:

因特网的网络层通过源和目的地之间的一系列路由器路由数据报。为了将分组从一个结点 (主机或路由器) 移动到路径上的下一个结点, 网络层必须依靠该链路层的服务。特别是在每个结点, 网络层将数据报下传给链路层, 链路层沿着路径将数据报传递给下一个结点。在下个结点, 链路层将数据报上传给网络层。

由链路层提供的服务取决于应用于该链路的特定链路层协议。例如, 某些协议基于链路提供可靠传递, 从传输结点跨越一条链路到接收结点。值得注意的是, 这种可靠的传递服务不同于 TCP 的可靠传递服务, TCP 提供从一个端系统到另一个端系统的可靠交付。链路层的例子包括以太网、WiFi 和电缆接入网的 DOCSIS 协议。因为数据报从源到目的地传送通常需要经过几条链路, 一个数据报可能被沿途不同链路上的不同链路层协议处理。例如, 一个数据报可能被一段链路上的以太网和下一段链路上的 PPP 所处理。网络层将受到来自每个不同的链路层协议的不同服务。在本书中, 我们把链路层分组称为帧 (frame)。

在连接层, 信息以**帧(frame)**为单位传输。所谓的帧, 是一段有限的0/1序列。连接层协议的功能就是识别0/1序列中所包含的帧。比如说, 根据一定的0/1组合识别出帧的起始和结束。在帧中, 有**收信地址(Source, SRC)**和**送信地址(Destination, DST)**, 还有能够探测错误的**校验序列(Frame Check Sequence)**。当然, 帧中最重要的最主要是所要传输的数据 (payload)。这些数据往往符合更高层协议, 供网络的上层使用。与数据相配套, 帧中也有数据的类型(Type)信息。连接层协议不关心数据中到底包含什么。帧就像是一个信封, 把数据包裹起来。

以太网(Ethernet)和WiFi是现在最常见的连接层协议。通过连接层协议, 我们可以建立局域的以太网或者WiFi局域网, 并让位于同一局域网络中的两台计算机通信。连接层就像是一个社区的**邮差**, 他认识社区中的每一户人。社区中的每个人都可以将一封信(帧)交给他, 让他送给同一社区的另一户人家。

- 网络层:

因特网的网络层负责将称为数据报（datagram）的网络层分组从一台主机移动到另一台主机。在一台源主机中的因特网运输层协议（TCP 或 UDP）向网络层递交运输层报文段和目的地址，就像你通过邮政服务寄信件时提供一个目的地址一样。

因特网的网络层包括著名的 IP 协议，该协议定义了数据报中的各个字段以及端系统和路由器如何作用于这些字段。仅有一个 IP 协议，所有具有网络层的因特网组件必须运行 IP 协议。因特网的网络层也包括决定路由的路由选择协议，它使得数据报根据该路由从源传输到目的地。因特网具有许多路由选择协议。如我们在 1.3 节所见，因特网是一个网络的网络，在一个网络中，其网络管理者能够运行所希望的任何路由选择协议。尽管网络层包括了 IP 协议和一些路由选择协议，但通常把它简单地称为 IP 层，这反映了 IP 是将因特网连接在一起的粘合剂这样的事实。

计算机1，路由器和计算机2都要懂得IP协议。当计算机1写信的时候，会在信纸的开头（网络层协议）写上这封信的出发地址和最终到达地址（而不是在信封上），而在信封上写上要送往邮局（链路层协议）。WiFi网的邮差将信送往邮局。在邮局，信被打开，邮局工作人员看到最终地址，于是将信包装在一个新的信封中，写上出发地为邮局，到达地为计算机2，并交给以太网的邮差，由以太网的邮差送往计算机2

• 传输层：

因特网的运输层在应用程序端点之间传送应用层报文。在因特网中，有两个运输协议，即 TCP 和 UDP，利用其中的任一个都能运输应用层报文。TCP 向它的应用程序提供了面向连接的服务。这种服务包括了应用层报文向目的地确保传递和流量控制（即发送方/接收方速率匹配）。TCP 也将长报文划分为短报文，并提供拥塞控制机制，因此当网络拥塞时，源抑制其传输速率。UDP 协议向它的应用程序提供无连接服务。这是一种不提供不必要服务的服务，没有可靠性，没有流量控制，也没有拥塞控制。在本书中，我们把运输层分组称为报文段（segment）。

上面的三层协议让不同的计算机之间可以通信。但计算机中实际上有许多个进程，每个进程都可能都有通信的需求。这就好像一所房子里住了好几个人(进程)，如何让信精确的送到某个人手里呢？遵照之前相同的逻辑，我们需要在信纸上写上新的信息，比如收信人的姓名，才可能让信送到。所以，传输层就是在信纸的空白上写上新的“收信人”信息。每一所房子会配备一个管理员(传输层协议)。管理员从邮差手中接过信，会根据“收信人”，将信送给房子中的某个人。传输层协议，比如TCP和UDP，使用端口号(port number)来识别收信人(某个进程)。在写信的时候，我们写上目的地的端口。当信到达目的地的管理员手中，他会根据传输层协议，识别端口号，将信送给不同的人。

• 应用层：

应用层是网络应用程序及它们的应用层协议存留的地方。因特网的应用层包括许多协议，例如 HTTP（它提供了 Web 文档的请求和传送），SMTP（它提供了电子邮件报文的传输）和 FTP（它提供两个端系统之间的文件传送）。我们将看到，某些网络功能，如将像 www.ietf.org 这样对人友好的端系统名字转换为 32 比特网络地址，也是借助于特定的应用层协议即域名系统（DNS）完成的。我们将在第 2 章中看到，创建并部署我们自己的新应用层协议是非常容易的。

应用层协议分布在多个端系统上，一个端系统中的应用程序使用协议与另一个端系统中的应用程序交换信息的分组。我们把这种位于应用层的信息分组称为报文（message）。

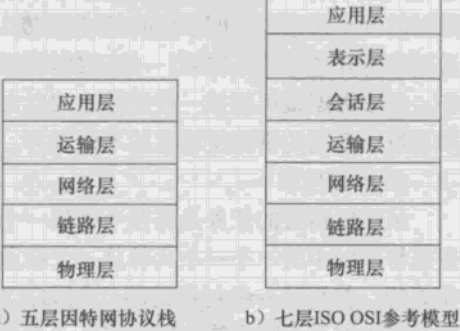
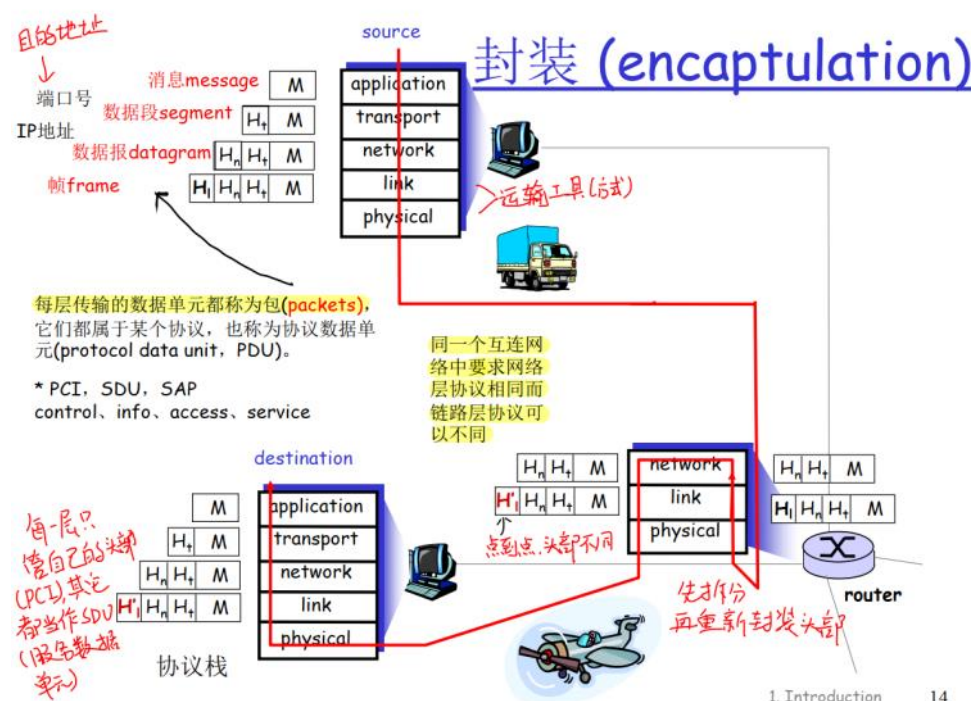


图 1-23 因特网协议栈和 OSI 参考模型

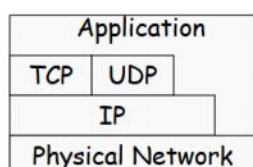
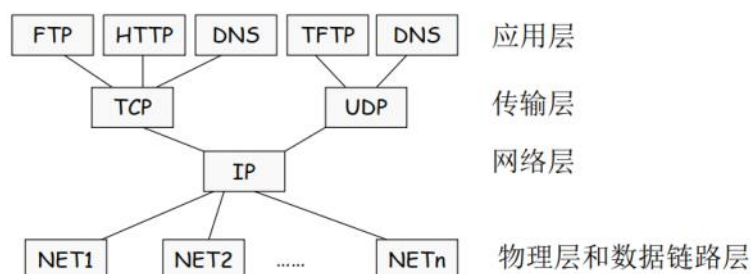
通过上面的几层协议，我们已经可以在任意两个人(进程)之间进行通信。然而每个人实际上从事的是不同的行业。有的人是律师，有的人外交官。比如说律师之间的通信，会用严格的律师术语，以免产生纠纷。再比如外交官之间的通信，必须符合一定的外交格式，以免发生外交误会。再比如间谍通过暗号来传递加密信息。应用层协议是对信件内容进一步的用语规范。应用层的协议包括用于Web浏览的HTTP协议，用于传输文件的FTP协议，用于Email的IMAP等等。

封装



- 分组交换技术 (包交换技术 circuit-switching) :
是将用户传送的数据划分成一定的长度，每个部分叫做一个分组，通过传输分组的方式传输信息的一种技术。每个分组的前面有一个分组头，用以指明该分组发往何地址，然后由交换机根据每个分组的地址标志，将他们转发至目的地，这一过程称为分组交换。
- 线路交换技术 packet-switching:
线路交换技术又称电路交换技术。电路交换是一种直接的交换方式，它为一对需要进行通信的装置（站）之间提供一条临时的专用通道，即提供一条专用的传输通道，既可是物理通道又可是逻辑通道（使用时分或频分复用技术） 例如：电话系统

协议簇(protocol family)



TCP 提供可靠的面向连接的服务

UDP 提供无连接不可靠的服务

协议簇是指互相关联的一组协议
比较出名的是TCP/IP协议簇

ISO/OSI参考模型

- 表示层(presentation): 提供数据转换服务, 例如, 加密解密, 压缩解压缩, 数据格式变换
- 会话层(session): 简化会话实现机制, 例如, 数据流的检查点设置和回滚以及多数据流同步。
- 因特网没有这两层, 如果需要, 可以在应用程序中实现。
例如: 开网络会议、保证直播不同数据流 (声音、画面) 同步



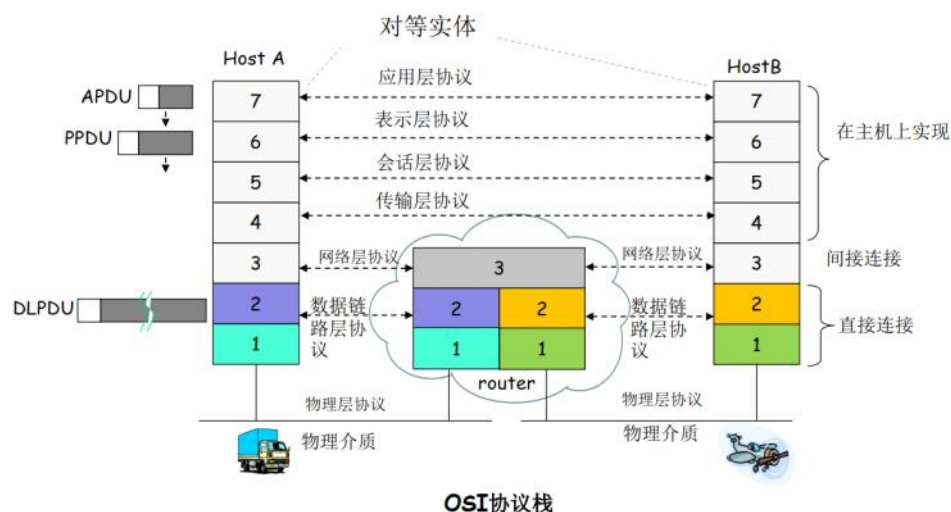
ISO — 国际标准化组织(International Organization for Standardization)
OSI — 开放系统互连(Open System Interconnection)

表示层位于OSI分层结构的第六层, 它的主要作用之一是为异种机通信提供一种公共语言, 以便能进行互操作。这种类型的服务之所以需要, 是因为不同的计算机体系结构使用的数据表示法不同。与第五层提供透明的数据运输不同, 表示层是处理所有与数据表示及运输有关的问题, 包括转换、加密和压缩。每台计算机可能有它自己的表示数据的内部方法, 例如, ASCII码与EBCDIC码, 所以需要表示层协定来保证不同的计算机可以彼此理解。

例如, IBM主机使用EBCDIC编码, 而大部分PC机使用的是ASCII码。在这种情况下, 便需要表示层来完成这种转换。如果您想要用尽量少的词语来记住这第6层, 那就是“一种通用的数据格式”。

会话层, 表示层, 应用层构成开放系统的高3层, 面对应用进程提供分布处理, 对话管理, 信息表示, 恢复最后的差错等。会话层同样要担负应用进程服务要求, 而运输层不能完成的那部分工作, 给运输层功能差距以弥补。主要的功能是对话管理, 数据流同步和重新同步。要完成这些功能, 需要由大量的服务单元功能组合, 已经制定的功能单元已有几十种。如果您想要用尽量少的词来记住这第5层, 那就是“对话和交谈”。

对等实体 (Peer Entity)



对等实体之间传输的数据包是: PDU

对等实体指的是实现了同一个协议的软件和硬件

对等实体(peer entity)指实现了同一个协议的软件或硬件。PCI (Protocol Control Information) 是 PDU (Protocol Data Unit) 的头部, 包含协议有关的信息。下层把上层通过SAP (Service Access Point) 传来的

SDU (Service Data Unit) 用PCI封装为PDU后传给对等实体。

延迟和丢包:

当一个包到达时如果有空闲缓存则排队等待转发, 产生延迟(delay)
如果没有空闲缓存, 则丢弃该包, 造成丢失(loss)

延迟不可太久,

包交换技术: Packet-Switching

1. 处理延迟(processing)

- ❖ 检查比特错
- ❖ 确定输出链路

2. 排队延迟(queueing)

- ❖ 在输出链路队列等待发送
- ❖ 依赖于路由器的拥塞程度

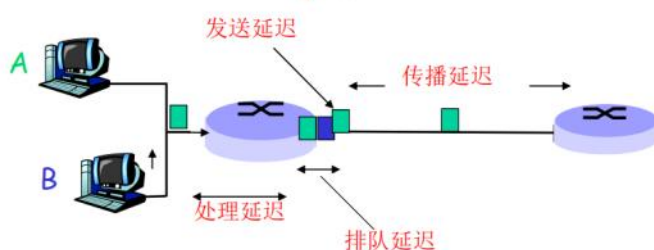
3. 传输延迟(Transmission delay):

包长(bits)/链路带宽 (bps)
也称为发送延迟

bits per second
每秒多少比特

4. 传播延迟(Propagation delay):

物理链路长度/信号传播速度
($\sim 2 \times 10^8$ m/sec)



节点延迟

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- ❑ d_{proc} = 处理延迟
- ❑ d_{queue} = 排队延迟
- ❑ d_{trans} = 发送延迟
- ❑ d_{prop} = 传播延迟

往返时间

- ❑ 从源主机到目的主机再返回源主机所花的时间称为往返时间(round trip time, RTT)。



带宽和吞吐量:

- **带宽(bandwidth)**为一条链路或通道可达到的最大数据传送速率(bits per second, bps)。
- **吞吐量(throughput)**: 一条链路或通道的实际数据传送速率(bps)
 - ❖ **瞬时吞吐量**: 某个时刻的数据速率
 - ❖ **平均吞吐量**: 在某个较长时期的数据速率