

第三章 数据链路层

主要内容

- 设计要点
- 基本的数据链路层协议
- 协议验证
- 例子

数据链路

- 链路：一条无源的点到点的物理线路，中间没有任何交换节点。又称物理链路。
- 数据链路：物理链路加上必要的通信协议，也称逻辑链路。其上传输的是帧。
- 一条链路上可以复用多条数据链路。

数据链路层设计要点

- 数据链路层存在两个不可避免的现实：
- （1）接收方接收数据的速率与发方发送数据的速率不匹配；
- （2）信道不能保证数据不产生错误。
- 解决途径：
- （1）由接收方控制发送方的数据流；
- （2）a.检错； b.收方收到以正确的帧向发方发确认帧，收到错误帧向发方发否认帧； c.若发方的数据丢失？发方对每一帧设置超时计时器； d.若确认帧丢失，会出现重复帧，帧上带有发送序号

数据链路层设计要点

- 数据链路层的主要功能
 - （1）链路管理，连路的建立、维护和释放
 - （2）帧同步，接收方能够从收到的比特流中识别一帧的开始和结束
 - （3）流量控制
 - （4）差错控制
 - （5）将数据和控制信息区分开
 - （6）透明传输，数据中会出现与控制信息相同的比特组合，必须使收方能够识别，不会将这种数据误认为是控制信息。
 - （7）寻址

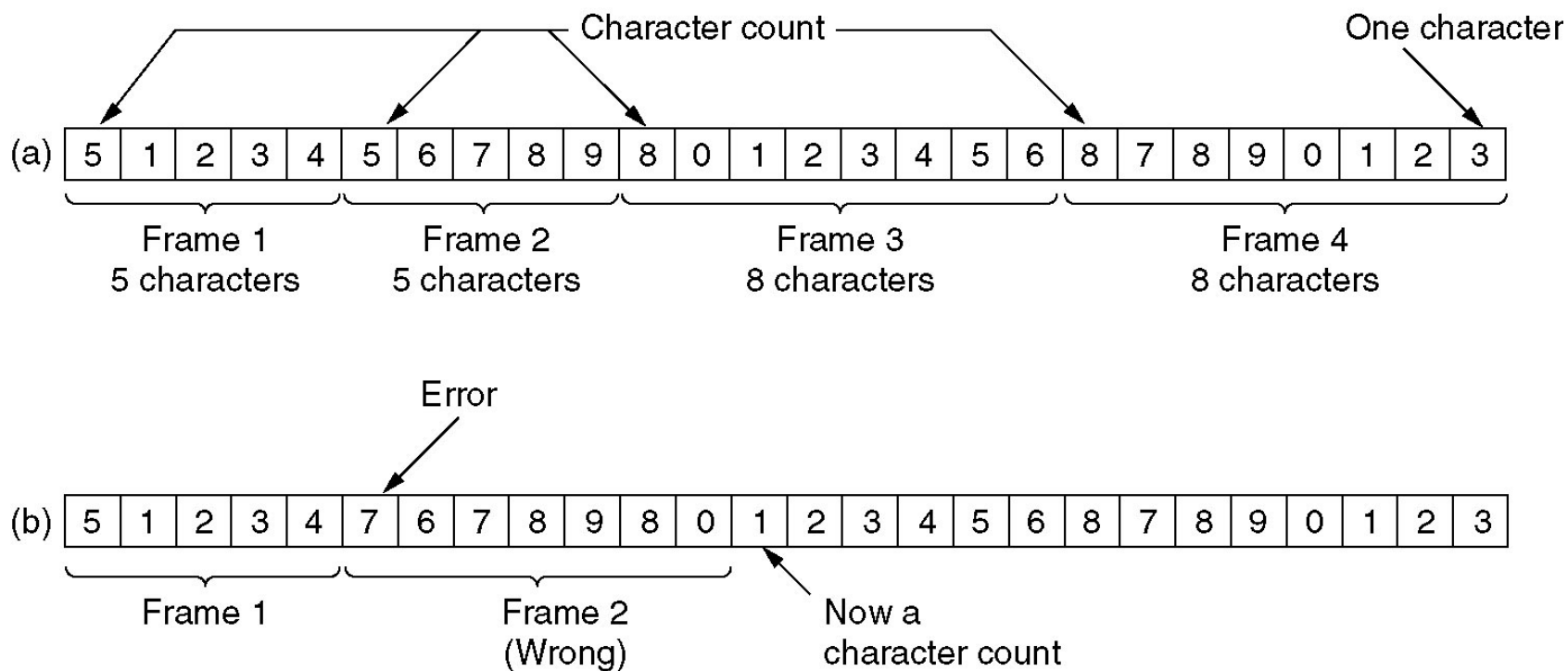
数据链路层设计要点

- 为网络层提供的服务
- （1）无确认的无连接服务
- （2）有确认的无连接服务
- （3）有确认的面向连接服务

帧同步

- 成帧：将原始的位流分解到离散的帧中。
- 方法：
 - (1) 字符计数法
 - (2) 含字节填充的分界符法
 - (3) 含位填充的分界标志法
 - (4) 物理层编码违例法

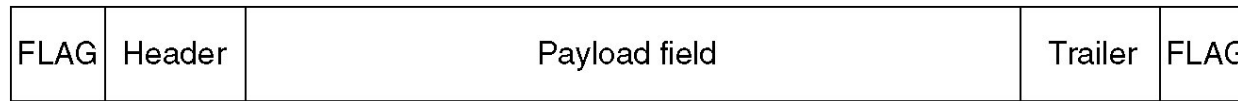
帧同步（字符计数法）



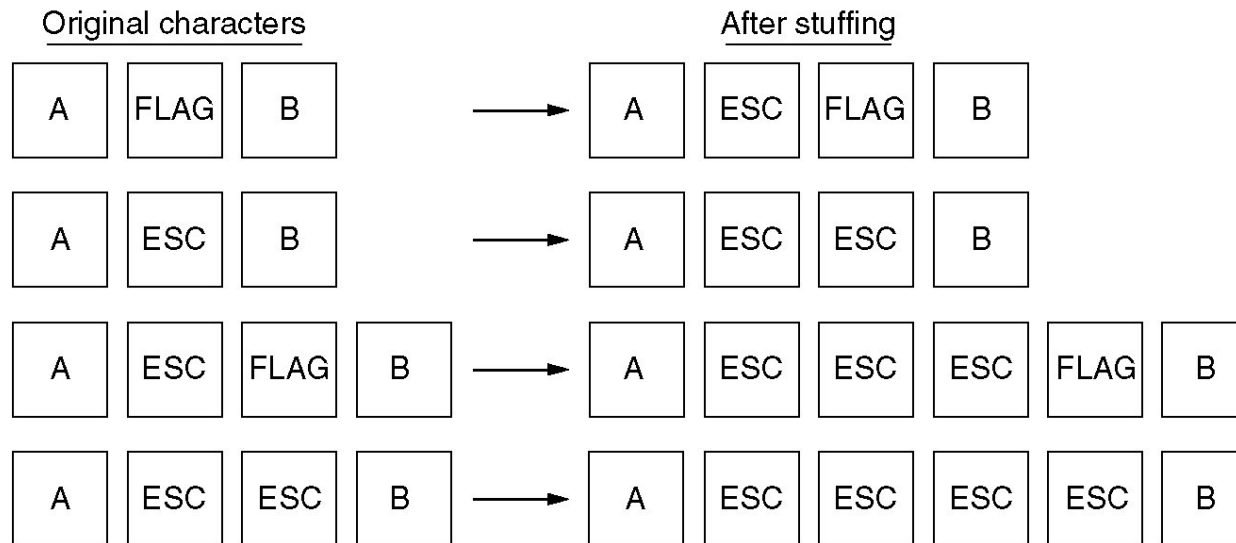
(a) Without errors. (b) With one error.

缺点：错误之后无法重新同步

帧同步（含字节填充的分界符法）



(a)



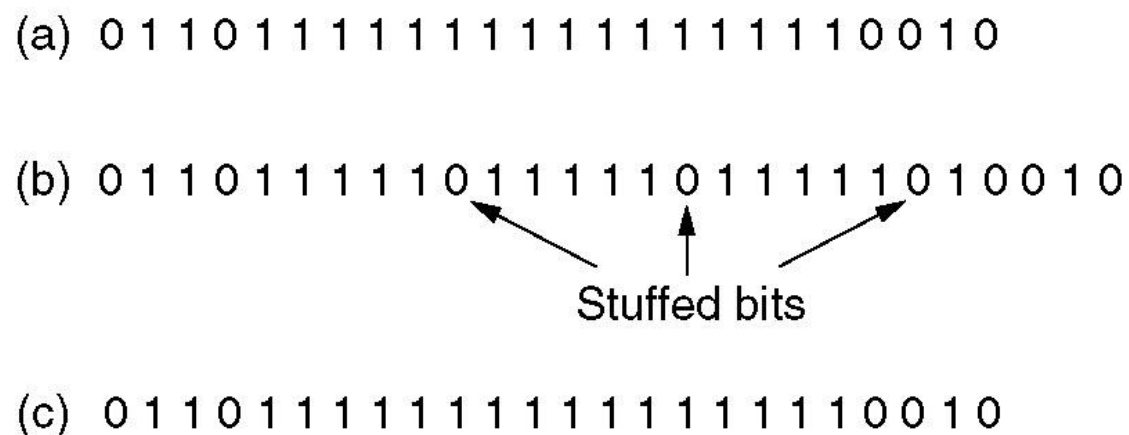
(b)

(a) 有标志字节作为分界的帧

(b) 字节填充后的4个字节序列的例子

缺点：面向字符的，依赖于8位模式

帧同步（含位填充的分界标志法）



比特填充

(a) 原始数据，(b) 线路上的数据，(c) 接收方删除填充之后的数据

帧同步（物理层编码违例法）

- 物理层编码违例法，只适用于“物理介质上的编码方法中包含冗余信息”的网络。

数据链路层设计要点

- 站在数据链路层观察，数据链路存在两个不可避免的现实：
- 信道不能保证所传输的数据不产生错误；
- 收方接收数据的速率与发方发送数据的速率不匹配。
- 而对于可靠的面向连接的服务来说，需要保证帧的正确可靠交付，通常采用**反馈**的方法。

错误控制

- 对于上面的第一个问题采用差错控制机制。
- (1) 纠、检错码；
- (2) 收方收到一帧数据经检验正确，则向发方发确认帧ack，错误，向发方发否认帧nak；
- (3) 若发方的数据帧丢失，怎么办？发方设置超时计时器来解决；
- (4) 若确认帧丢失，会出现重复帧，收方如何区分是重复帧还是新的一帧？给数据帧带上序号；

流量控制

- 对于上面的第二个问题了一采用流量控制。
- （1）基于反馈的流量控制；（由接收方通过反馈信息告诉发送方接收方的情况。）
- （2）基于速率的流量控制。（一种内在机制限制了发送方的发送速率，第五章讨论）

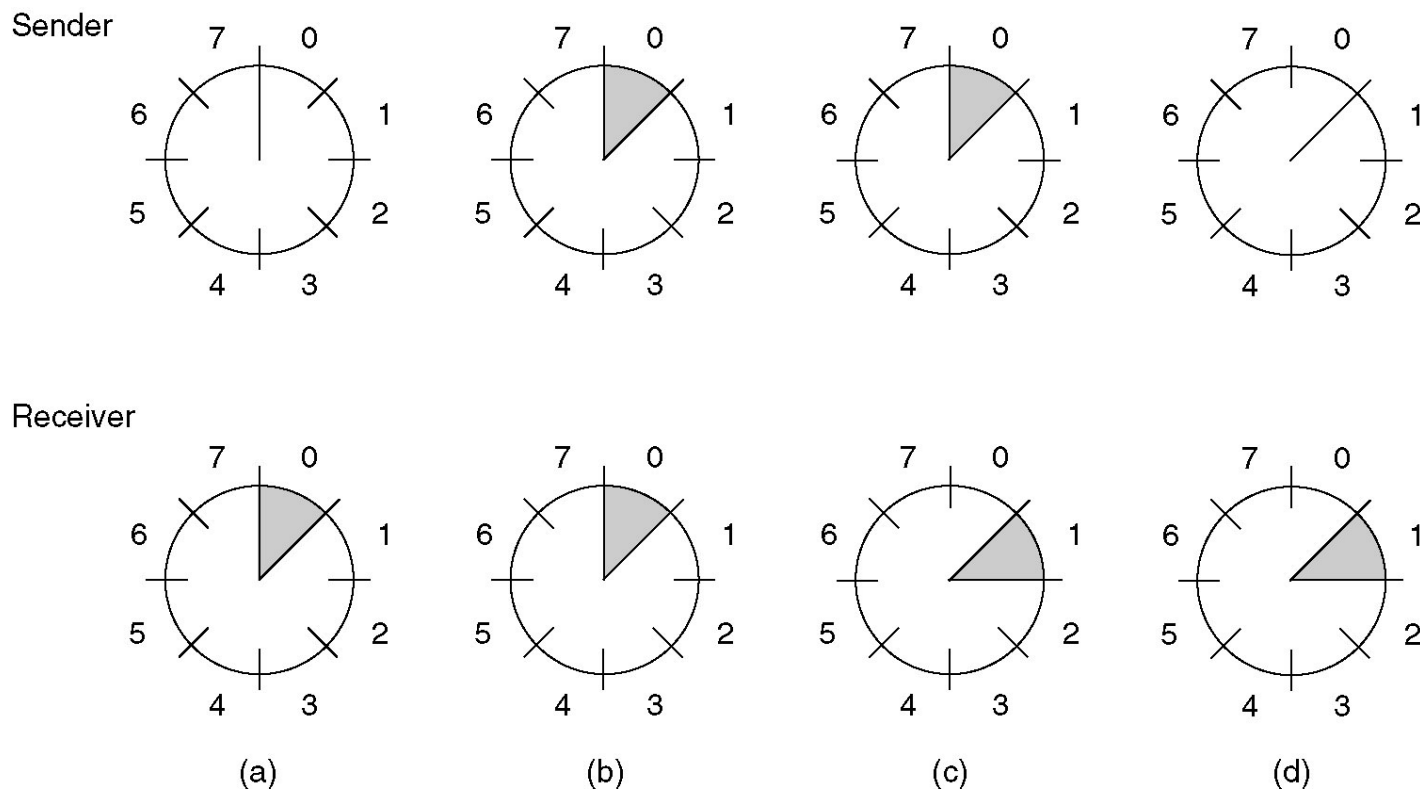
基本数据链路协议

- 滑动窗口协议
- (1) 1位滑动窗口协议;
- (2) 返回n的协议;
- (3) 选择重传的协议。

滑动窗口协议

- 滑动窗口协议：在任何时刻，发送方总维持着一组序列号，分别对应于它允许发送的帧；接收方也维持着一个接收窗口，对应于一组允许它接收的帧。注意，发送方的发送窗口和接收方的接收窗口不必要有相同的上界和下界，也不必要有同样的大小。
- 发送窗口：指发送方在收到接收方的应答之前允许连续发送的帧的最大数目。
- 接收窗口：控制接收方只接收指定编号的数据帧，若收到的帧的序号不在接收窗口之内，则一律将其丢弃。
- 每收到一个序号正确的帧，接收窗口就向前滑动一个位置，同时向发送方返回一确认帧，发送窗口才向前移动一个位置。

滑动窗口协议



一个大小为1、有3位序列号的滑动窗口

(a) 初始时 (b) 第一帧发送完 (c) 第一帧接收完 (d) 第一帧确认后

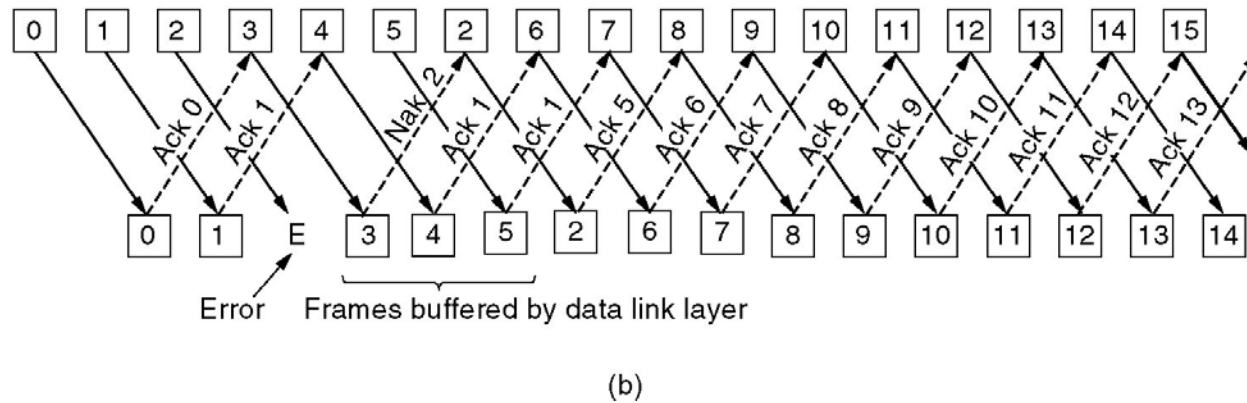
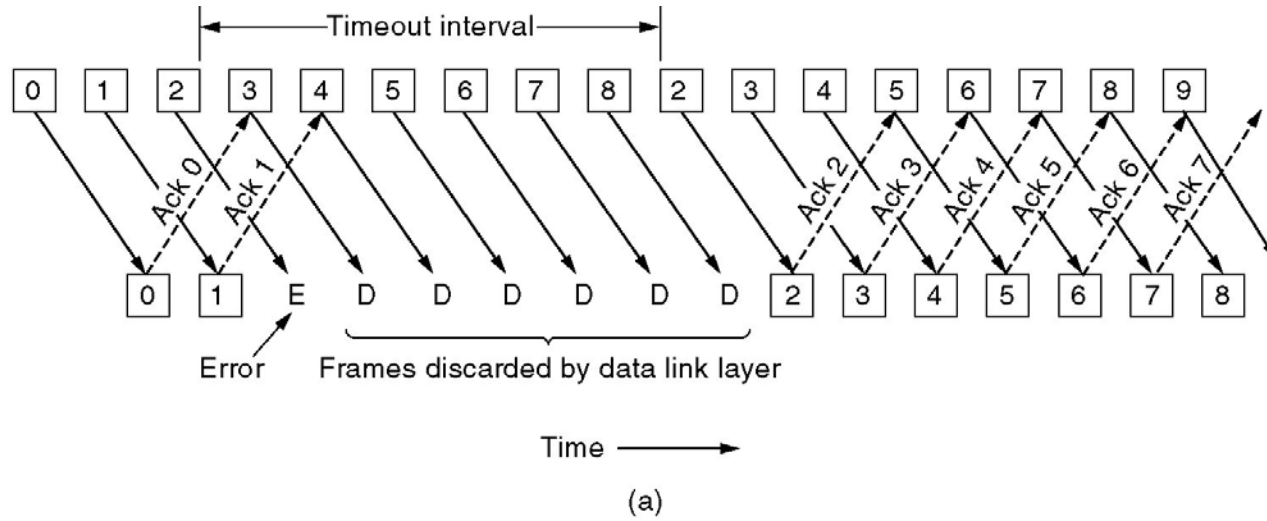
1位滑动窗口协议

- 发送窗口为1的滑动窗口协议。
- 由于发送方在送出一帧以后，在发送下一帧之前要等待前一帧的确认，是一种停-等协议。

返回n的协议

- 发送方在发完一帧数据后，不时停下来等待对方的应答，而是可以继续再发送若干个帧。
- 发送窗口大于1；
- 接收窗口为1，因此只能按序接收。
- 处理错误方法：回退n帧；选择重传。（这两种不同的方法是在带宽和数据链路层缓存空间之间的权衡）

返回 N 的协议



(a) 接收窗口尺寸为1 (b) 接收窗口较大

选择重传的协议

- 只对出错的帧重传。
- 要求就收方有较大的缓存。
- 接收窗口大于1 。

数据链路层协议示例（HDLC）

- HDLC，高级数据链路控制。
- IBM设计BSC规程（面向字符）；
- 1974年IBM推出SNA，在数据链路层采用面向比特的规程SDLC；
- SDLC后经ISO改为HDLC；面向比特。
- CCITT又作了修改后，称为LAP（链路接入规程）；面向比特。
- LAP的新版本LAPB（平衡型的LAP），面向比特。

H D L C

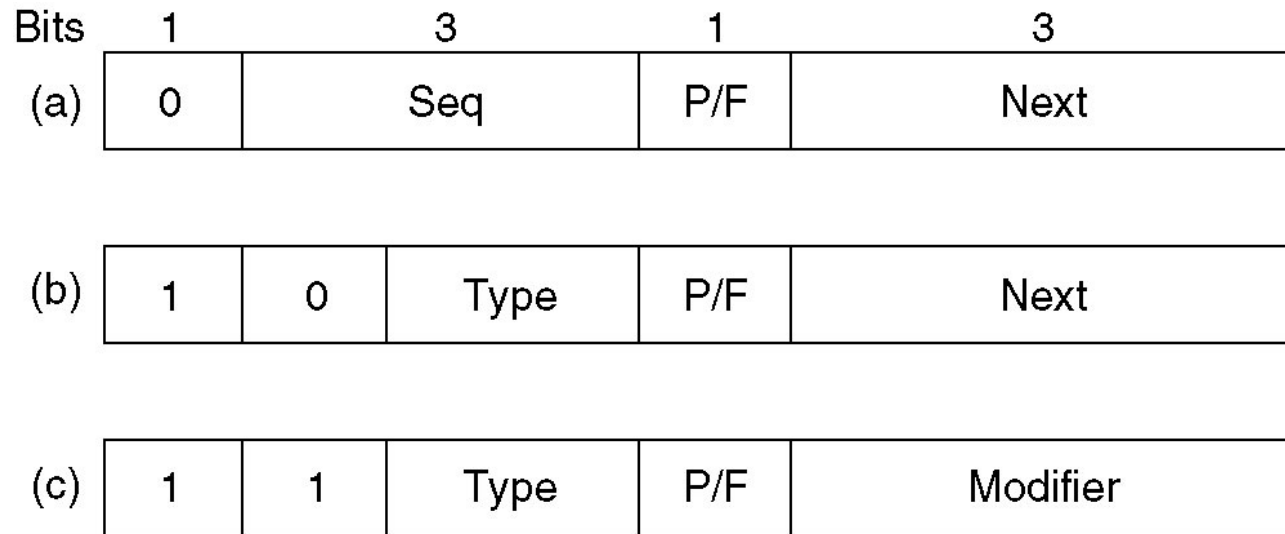
- (1) 平衡配置;
- (2) 非平衡配置。

面向比特协议的帧格式



有三种类型的帧，信息帧、监控帧和无序号帧，
区别在于控制域的前两位

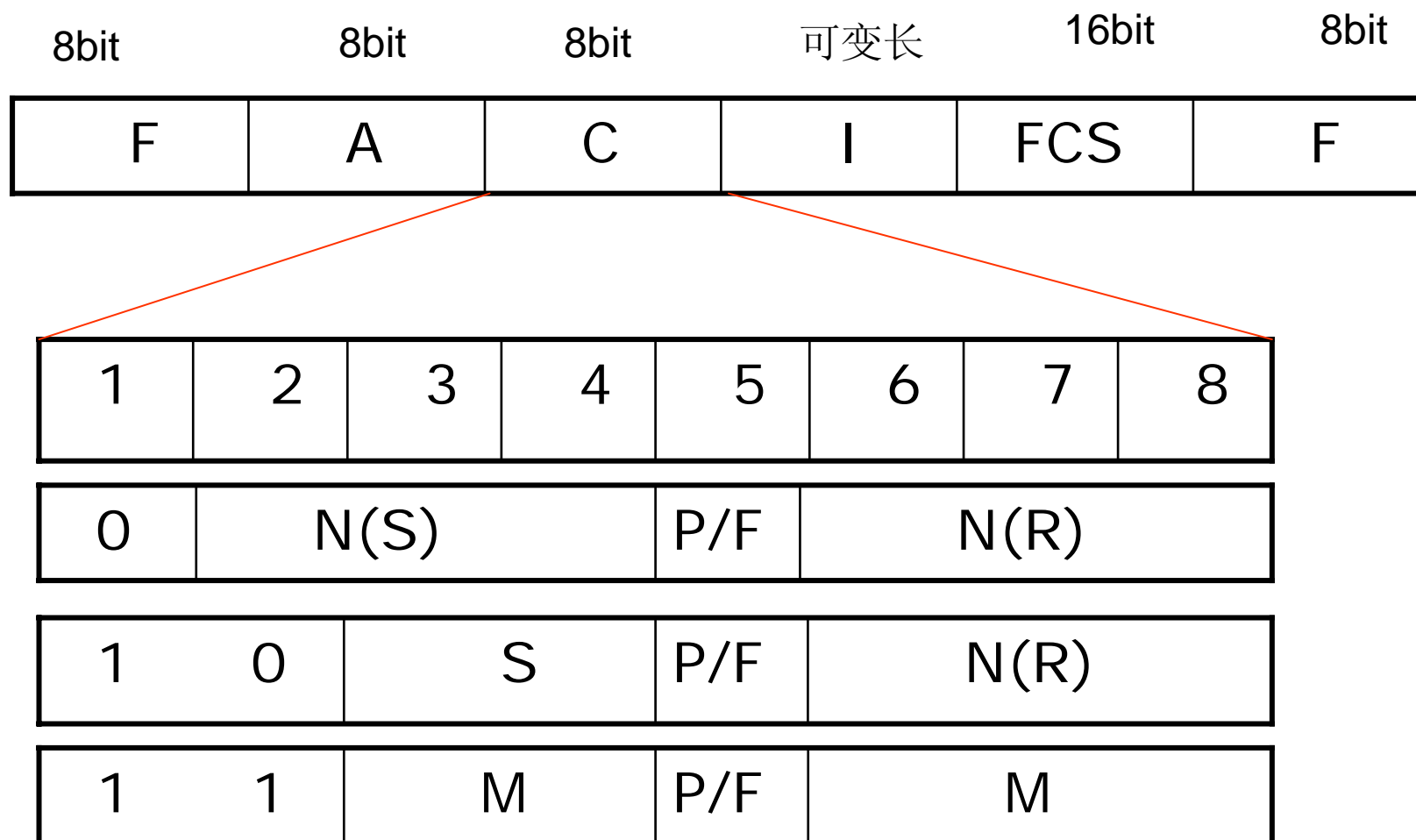
三种帧的控制域



Control field of

- (a) An information frame.
- (b) A supervisory frame.
- (c) An unnumbered frame.

HDL C 的帧格式



HDL C 的帧格式

- 标志序列 (F)

使用01111110比特序列作为一帧的开始 和 结束，进行收发双方同步及帧间信息同步。

- 透明传输，采用“零比特插入删除法”——在两个标志之间若出现连续5个“1”，则在其后插入一个“0”。

H D L C 的帧格式

□ 地址字段（A）

8 比特，指明次站地址。（在命令帧中指明接收该命令的次站的地址，在响应帧中则指明应答响应的次站地址，全“1”为广播地址）

□ 控制字段（C）

8 比特，表示命令和响应的类别及功能。该字段也可扩展为 16 比特，仅 N(S) 和 N(R) 扩展。

根据 C 字段前两位的取值，可将 H D L C 帧划分为三大类：信息帧，监控帧和无编号帧。

H D L C 的帧格式

- 信息字段（I）

任意的二进制组合，长度未作规定，一般为 2 5 5 字节。

- 帧检验序列 F C S

1 6 比特，用于检测误码，标志序列和插入的零不在校验范围之内。

信息帧

- 控制字段的第一个比特为“0”，该帧为信息帧。
- 控制字段 2 至 4 比特为发送序号 $N(S)$ ，表示当前此帧的序号；比特 6 至 8 为接收序号 $N(R)=v$ 表示对方发来的 $(v-1)\bmod 8$ 及以前的帧都正确收到，期望收到 $N(R)=v$ 号帧。
(通信是全双工的，采取了捎带确认)。
- 控制字段第 5 比特是探寻 / 结束位，当一台计算机正在询问一组终端时需要用到该域。当用作 P 时，该计算机正在请求终端发送数据。终端发送的所有帧，除了最后一帧外，都将 P / F 位设置为 P，最后一帧设置为 F。

监控帧

- ❑ 控制字段的前两位为“1 0”。用于执行链路的监控功能，如对帧的确认等。
- ❑ 控制字段的 3 至 4 比特，定义了 4 种应答（具有流控和差错控制的功能）：接收就绪 R R（0 0），接收未就绪 R N R（1 0），接收拒绝 R E J（0 1），选择拒绝接收 S R E J（1 1）。
- ❑ P / F 位和信息帧中的 P / F 含义相同。
- ❑ 所有的监控帧均不包含信息字段，所以长为 4 8 比特。

监控帧

- 控制字段的 6 至 8 比特为接收序号 $N(R)$ ，和 S 一起起到流控和差错控制的功能：
- 当 $s=RR$ 时，表示准备接收 $N(R)$ ，同时对 $N(R)-1$ 及以前的帧作确认（返回 n 的 $A R Q$ ）；
- 当 $s=RNR$ 时，暂停接收，同时对 $N(R)-1$ 及以前的帧作确认（返回 n 的 $A R Q$ ）；
- $s=REJ$ 时，表示从 $N(R)$ 起所有的帧都被否认，同时对 $N(R)-1$ 及以前的帧作确认（返回 n 的 $A R Q$ ）；
- $s=SREJ$ 时，只否认 $N(R)$ 号帧，但确认 $N(R)-1$ 及以前的帧，（选择重传的 $A R Q$ ）。

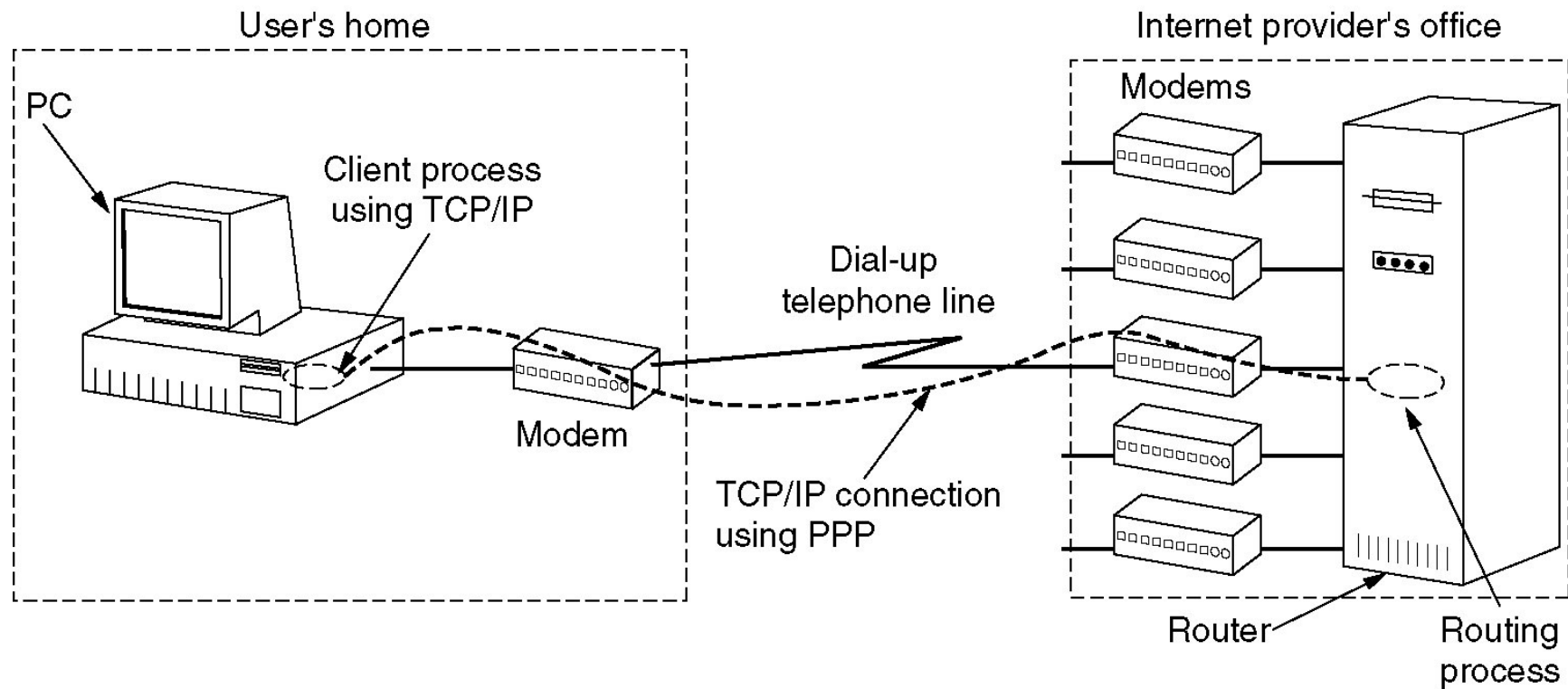
无编号帧

- 控制字段的前两位为 1 1 ，用来提供附加的链路控制功能。
- 无编号帧没有N(S)和N(R)， 5 比特的M用来表示不同功能命令和响应。如置工作方式，或终止已建立的工作方式。

点到点

- 实践中有两种点到点通信：
 - （1）从路由器到路由器的租用线路连接（通信子网情形）；
 - （2）从主机到路由器的拨号连接。
- 都需要点到点的数据链路层协议来完成成帧、错误控制等数据链路层功能。

Internet的数据链路层

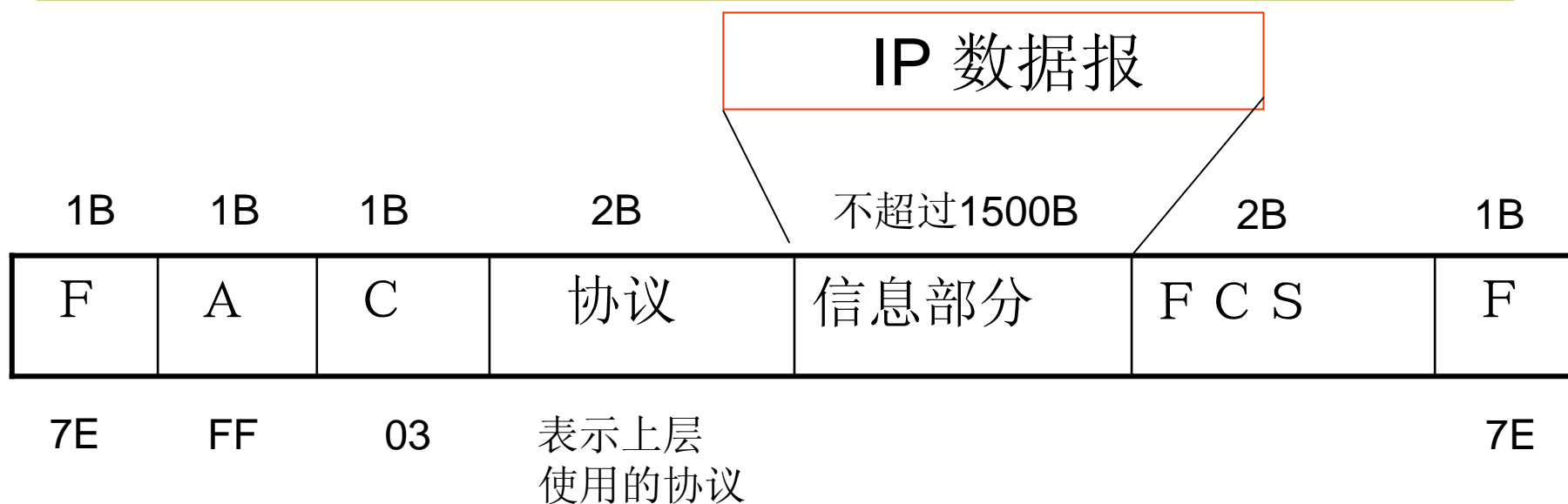


一种接入因特网的方式

P P P 点到点协议

- 用途：传送从路由器到路由器之间的流量以及从家庭用户到 I S P 之间的流量。
- 定义参见 R F C 1 6 6 1 。
- P P P 提供了三类功能：
 - (1) 一种成帧的方法；
 - (2) L C P ， 一个链路控制协议，用于启动链路、测试链路、协商参数以及关闭链路；
 - (3) 一套 N C P ， 一种协商网络层选项的方法。其中每一个协议支持一个不同的网络层协议（ I P ， I P X 等）

PPP的帧格式



P P P 的透明传输

□ P P P 的透明传输使用两种方法：

(1) 对于同步传输，和 H D L C 相同，采用“0”比特插入删除法；

(2) 对于异步传输，采用特殊字符填充法。如将信息字段出现的 O X 7 E 转换成 O X 7 D O X 5 E。

协议验证

- 形式化的、数学的方法描述和验证协议：
 - (1) 有限状态机模型
 - (2) Petri网模型

有限状态机模型

- 协议机，即发送方或接收方，在任何时刻总是处于一种特定状态。
- 通常总是选择协议机在等待下一事件发生的那些时刻作为协议机的状态，所有其它的状态都可以看作是瞬时态，只是为达到其中一个主状态的中间步骤而已。
- 整个系统的状态是两个协议机和信道的所有状态的组合。
- 转换，对于每一种状态，有 0 个或多个可能的转换到达其它状态。对于一个协议机，以下几种情况会发生转换：一帧被送出、一帧到达、一个定时器到期、产生一个中断等；对于信道：协议机将一个帧插入到信道上、一帧被递交给一个协议机、由于噪声丢失了帧等等。

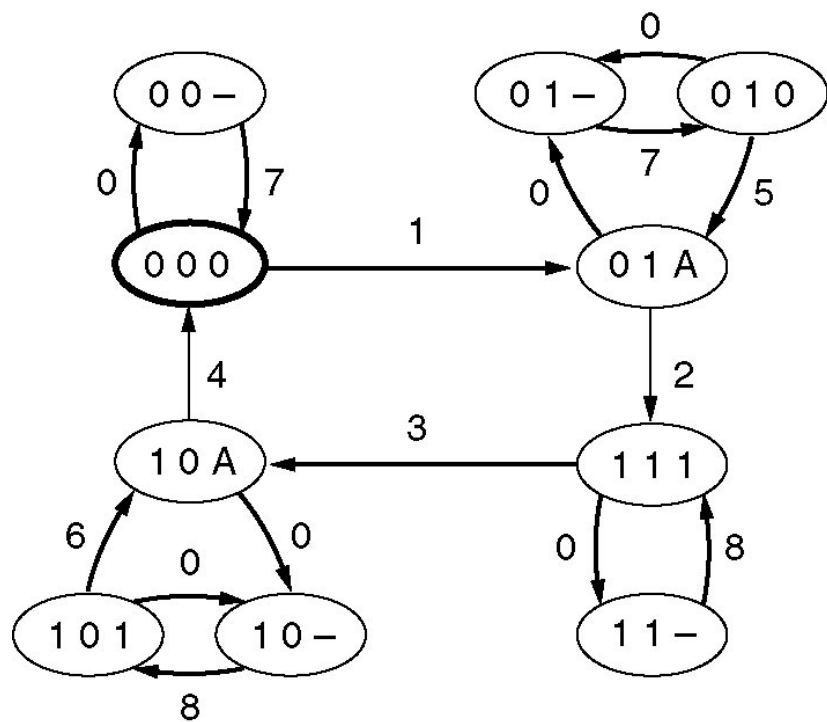
有限状态机模型（2）

- 初始状态，是一种特殊状态，他对应于系统开始运行时的状态。
- 给出了协议机和信道特征的完整描述，就可以画出一个有向图，其中，所有的状态显示为节点，而所有的转换显示为有向弧。这样，利用图论的技术和理论就可以确定那些状态是可达的，那些状态是不可达的，以及协议正确性的验证。

有限状态机模型（3）

- 从形式上看，一个协议的有限状态机可以看作一个四元组 (S, M, I, T) ，其中：
- S 是指进程和信道可能的状态集合。
- M 是指能在信道上进行交换的帧的集合。
- I 是指进程的初始状态的集合。
- T 是指状态之间转换的集合。

有限状态机模型 (3)



(a)

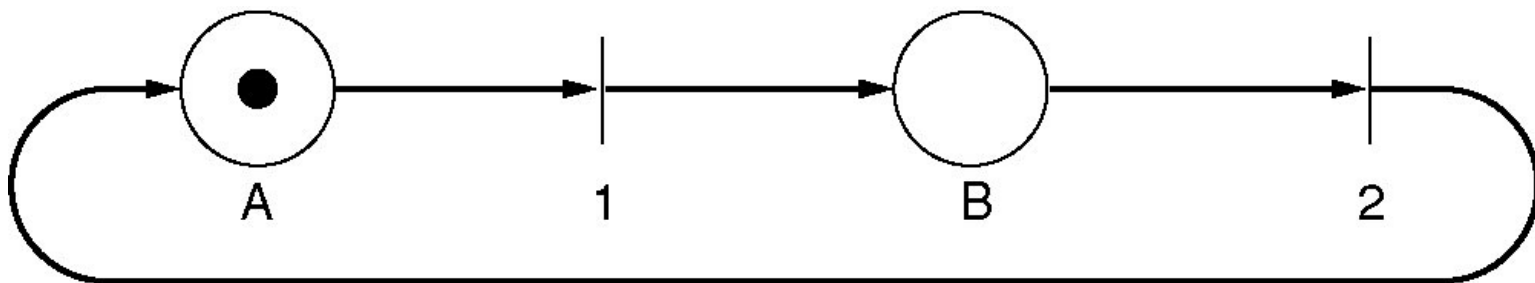
Transition	Who runs?	Frame accepted	Frame emitted	To network layer
0	-	(frame lost)		-
1	R	0	A	Yes
2	S	A	1	-
3	R	1	A	Yes
4	S	A	0	-
5	R	0	A	No
6	R	1	A	No
7	S	(timeout)	0	-
8	S	(timeout)	1	-

(b)

(a) 协议3的状态图 (b) 转换

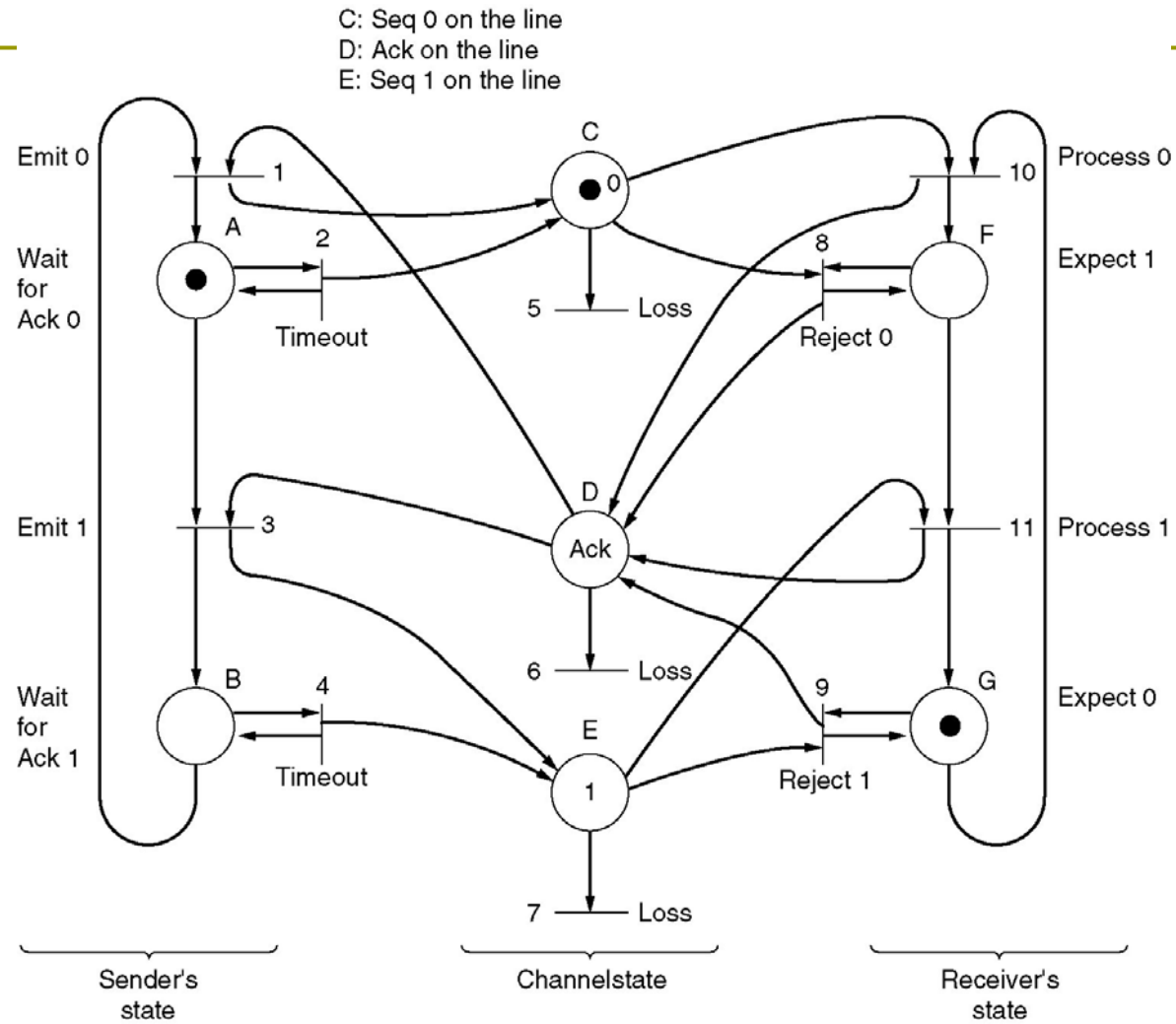
Petri网模型

- 一个Petri网有 4 个基本元素：库所，转换，弧和标记。其中库所，代表了该系统（或部分系统）可能处的状态。



包含两个库所和两个转换的Petri网

Petri网模型(2)



协议 3的Petri网模型