

端口号的完整列表可以在IANA网站上查看：

<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

对于网络层和传输层的描述在书上有一个很好的比喻，我将他汉化后的内容如下。考虑有两个家庭，一家位于辽宁沈阳，一家位于浙江宁波，每家有4口人。每个人每星期要互相写一封信，每封信都用单独的信封通过中国邮政传送。因此每个家庭每星期向另一家发送16封信。每一个家庭有个孩子负责收发邮件，浙江宁波家庭是小明，辽宁沈阳家庭是小红。每星期小红去她的家人那里收集邮件，并将这些信件投到社区的邮筒里。当新建到达沈阳的家庭时，小红也负责将信件分发给家里人。在宁波的小明家里也负责类似的工作。

在这个例子中，邮政服务为两个家庭间提供逻辑通信，邮政服务将信件从一家送到另一家，而不是从一个人送到另一个人。在另一方面，小明和小红为家人之间提供了逻辑通信，他们从家人那里收取信件或到家人那里交付信件。注意到从家人们的角度来看，小明和小红就是邮件服务，尽管他们只是端到端交付的一部分（即端系统部分）。类比关系如下：

进程 = 家人

主机（又称为端系统） = 家庭

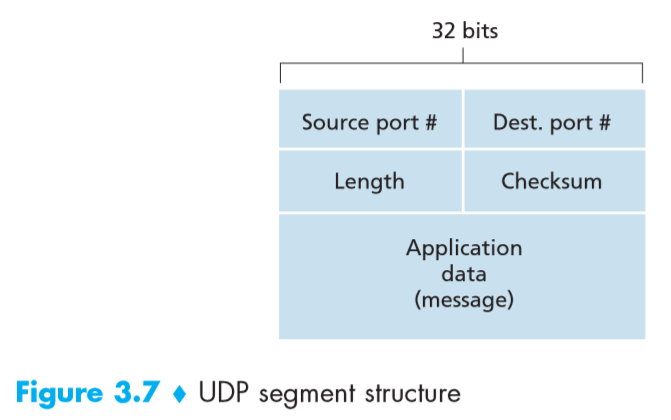
传输层协议 = 小明和小红

网络层协议 = 邮政服务（包括邮筒）

小明和小红都是在自己家里进行工作的，他们没有参与任何一个中间邮件中心对邮件进行分拣，或者将邮件从一个邮件中心送到另一个邮件中心之类的工作。类似地，运输船协议只工作在端系统中。在端系统中，运输船协议将来自应用进程的报文移动到网络边缘（即网络层），反过来也是一样，但对有关这些报文在网络核心如何移动并不作任何规定。

多路复用和多路分解的原理可以同样用这个家庭的例子来作类比。每一个家庭成员通过他们的名字来标识（相当于套接字）。当小明从邮递员出收到一批信件，并通过查看收信人名字而将信件交付给他的家人时，他执行的就是一个多路分解操作。当小红从家人那里收集信件并将它们投到邮筒中时，她执行的就是一个多路复用操作。

UDP报文段结构



UDP校验和（checksum）计算。发送方的UDP对报文端中的所有16比特字的和进行反码运算，求和时遇到的任何溢出都被回卷。得到的结果被放在UDP报文段的校验和字段。举例来说，假定我们有下面3个16比特的字：

0110011001100000

0101010101010101

1000111100001100

前两个之和是

0110011001100000

0101010101010101

1011101110110101

再将上面的和和第三个字相加，得出：

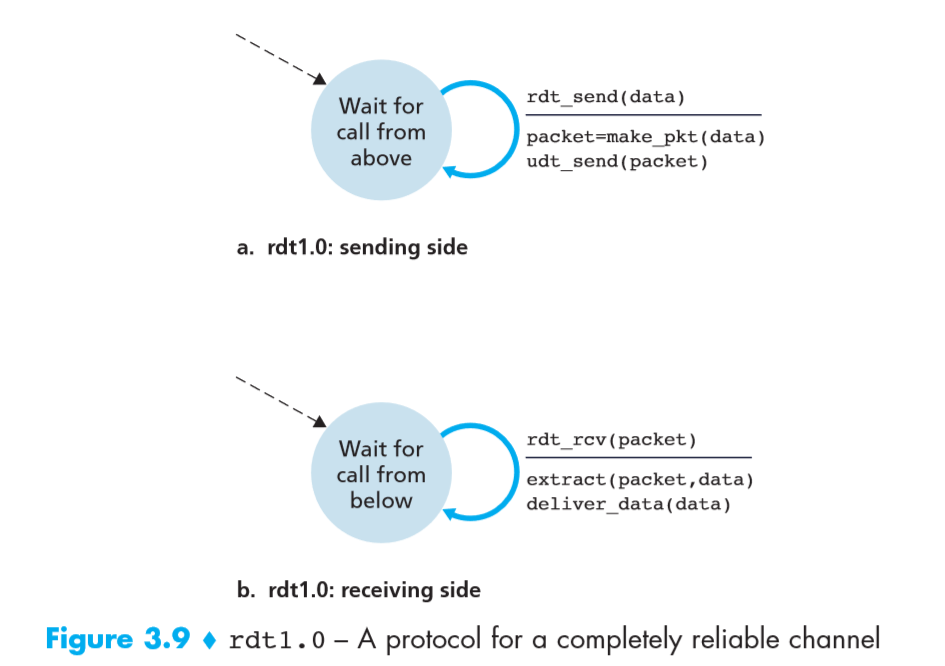
1011101110110101

1000111100001100

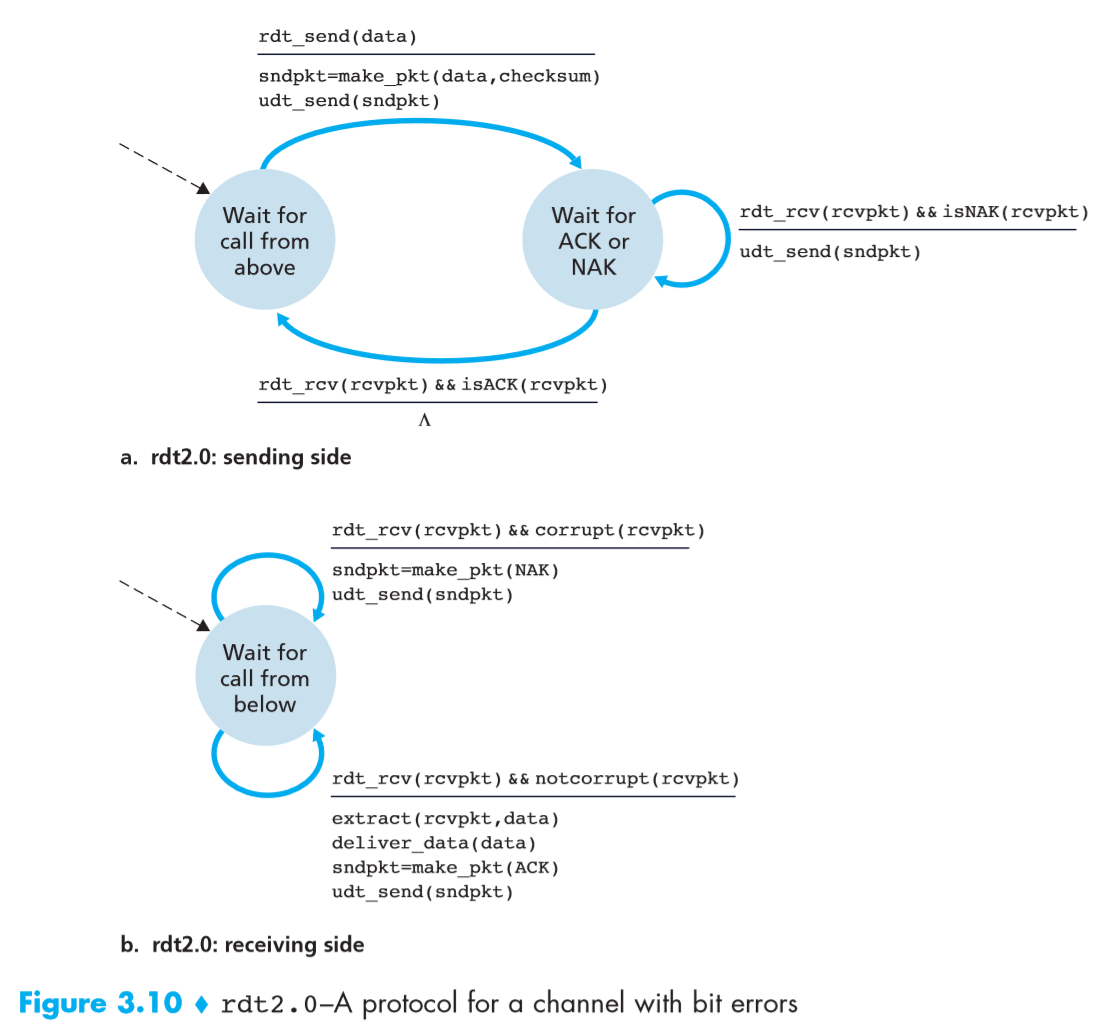
0100101011000010

注意到最后一次加法有溢出，它要被回卷。反码运算就是将所有的0换成1，所有的1换成0.因此，该和0100101011000010的反码运算结果是1011010100111101，这就是校验和。在接收方，全部的4个16比特字（包括校验和）加在一起，如果该分组中没有引入差错，则显然在接收方处该和将是1111111111111111。如果有一个是0，就代表分组中出了差错。

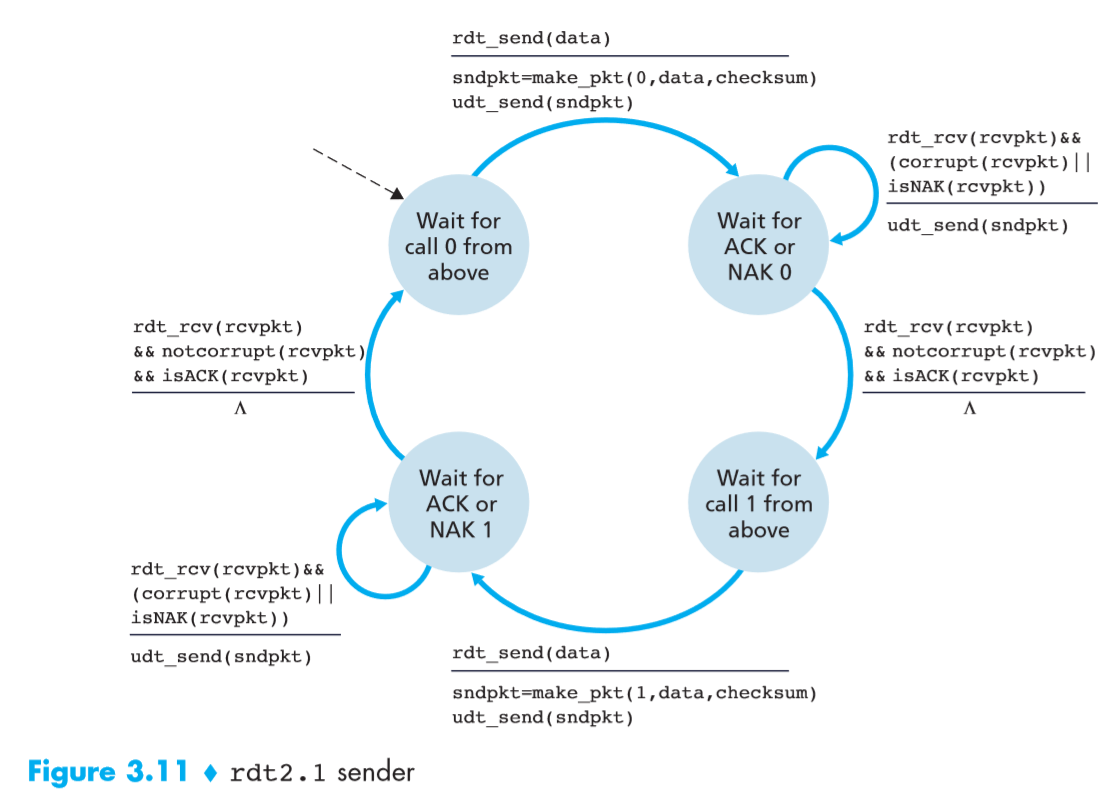
rdt1.0有限状态机（FSM）示意图

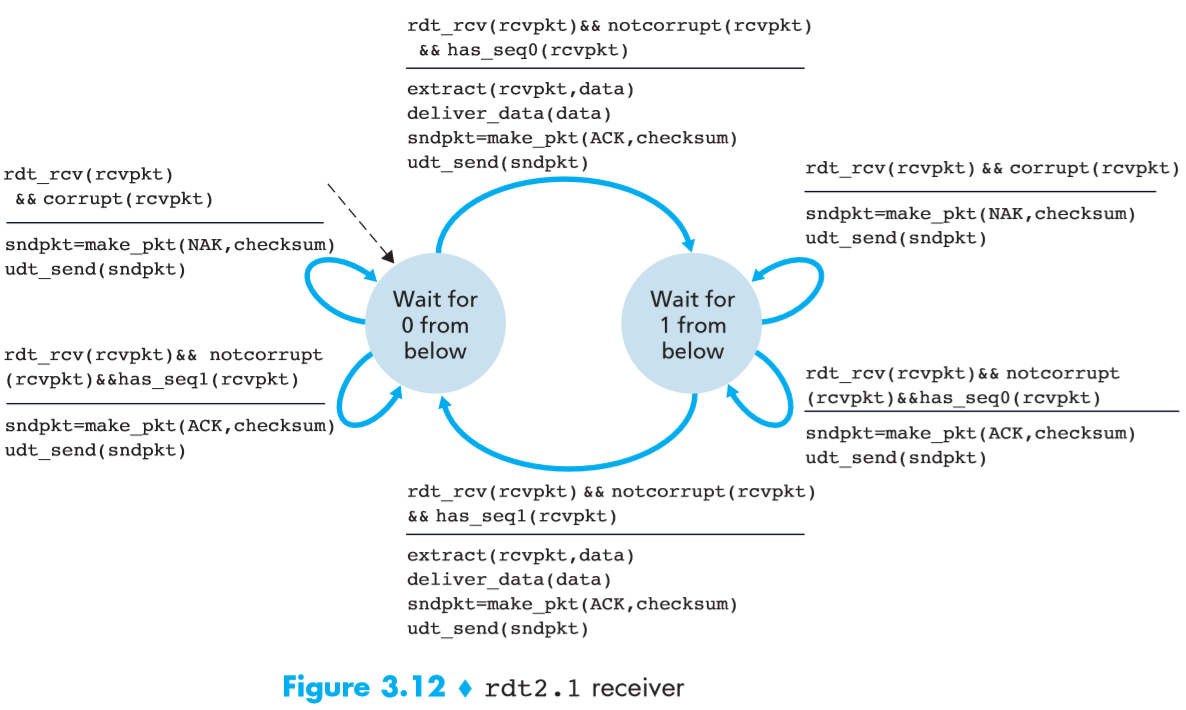


rdt2.0的FSM

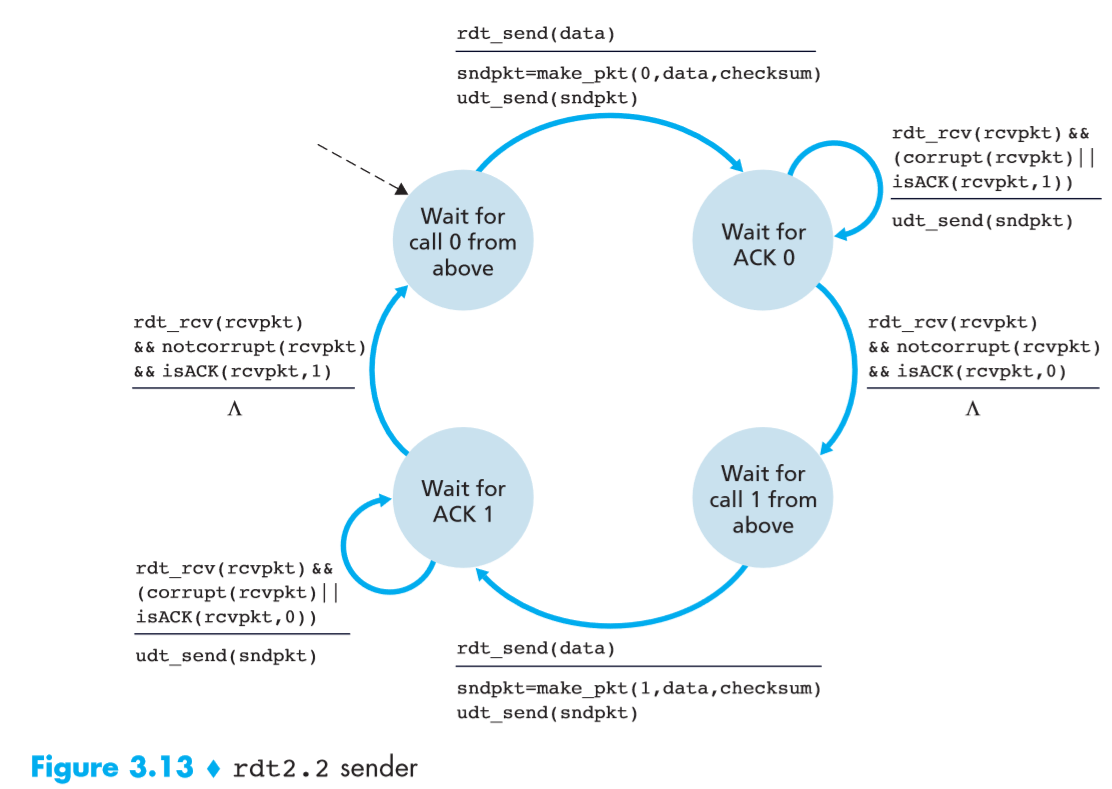


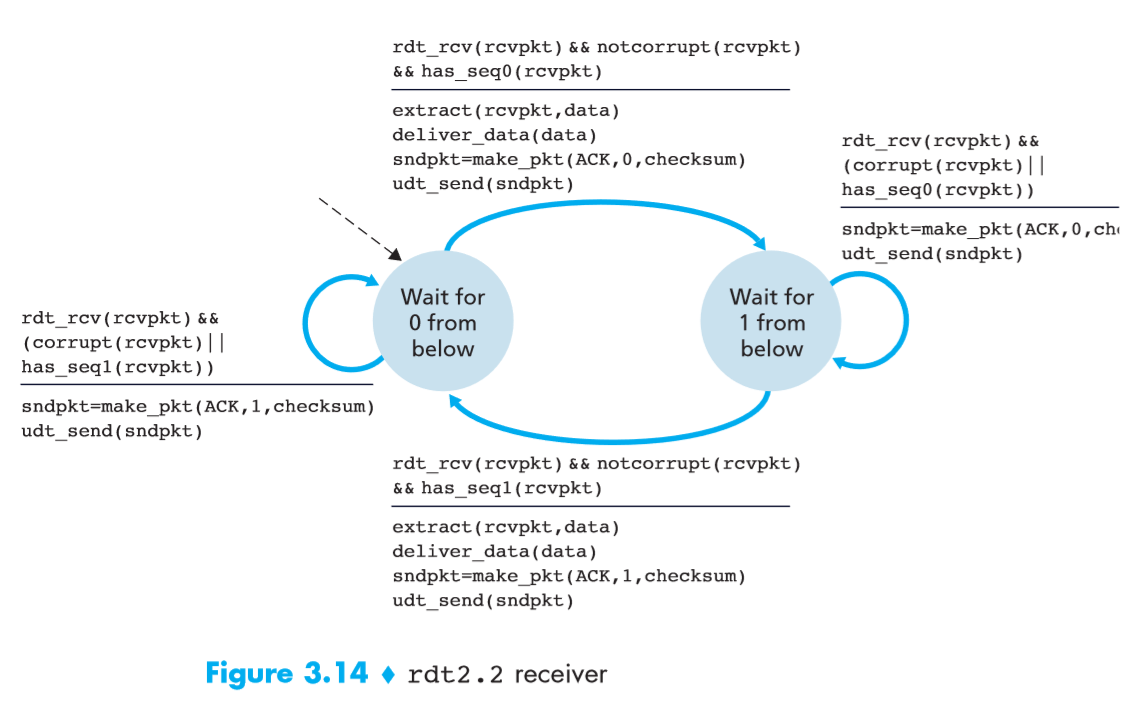
rdt2.1的FSM





rdt2.2的FSM





rdt3.0发送端的FSM，它的接收端的FSM和rdt2.2一样

