## 滑动窗口协议

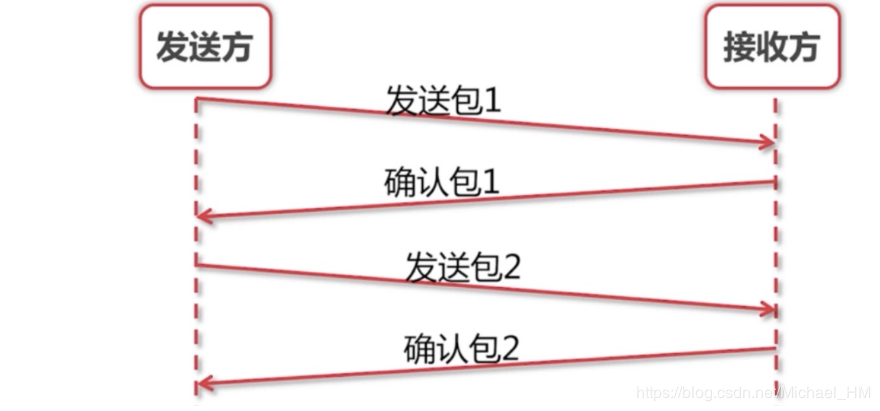
1. TCP协议的使用
2. 维持发送方/接收方缓冲区

缓冲区是用来解决网络之间数据数据不可靠的问题，例如丢包，重复等

在TCP协议中，发送方和接收方各位维护自己的缓冲区，通过商定包的传送机制来解决不可靠的问题

### 问题一：如果保证次序

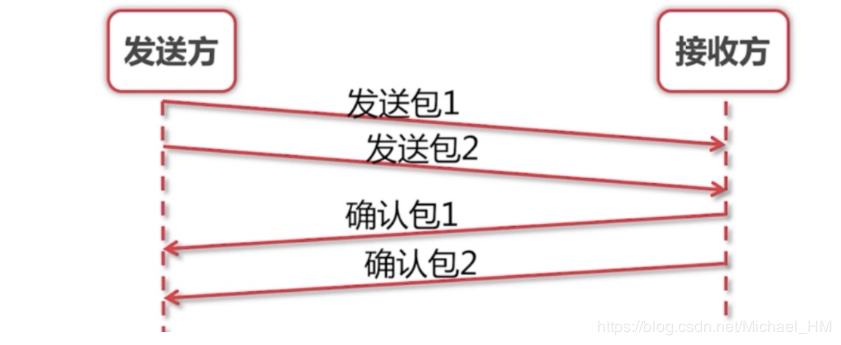
问题：在滑动窗口协议前，如何保证包的数据发送与被接受之间每个包斗牛被接受，并且是按次序发送的呢



发送方发送一个包1，这个是接受方发送确认包1，发送包2，确认包2，如此循环，直到数据发送完毕，这样就可以解决丢包的问题了，但是会有一个问题就是吞吐量非常低，要一个个来发送包

### 问题2,：如何提高吞吐量呢

问题：我们不能一次发送多个包吗，这样速度会不会更快，吞吐量更高

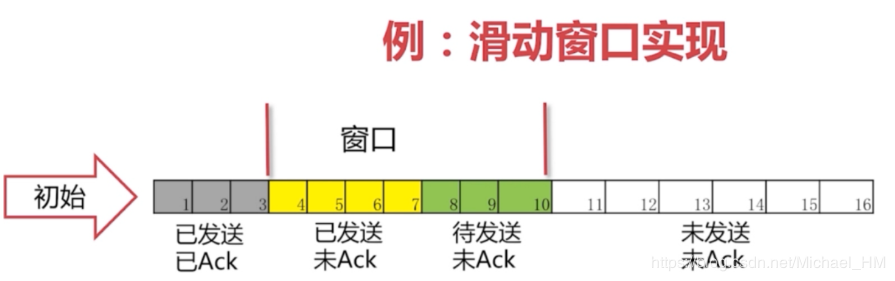


如图：这个就是我们把两个一起发送过去，然后一起确认，可以看出改进的方案比之前的要好，改善了吞吐量

### 问题3：如何实现最优解

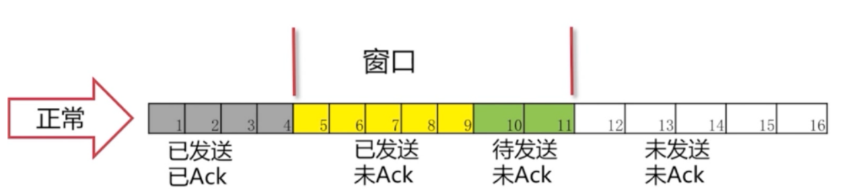
问题：我们每次需要发送多少个包过去呢，发送多少包是最优的呢

我们能不能把第一个和第二个包发送过去之后，收到第一包的确认就发送第三个包过去呢，而不是要等第二个包确认了才去发送两个包，就这样就产生了我们所说的滑动窗口：



在图中，我们可以看出灰色123是已经发送成功的数据，并且已经收到ack了，4567包是已发送但是还没有收到ack回复的包，所以不知道接收方有没有收到数据。8910是待发送的数据，后面的11-16是还没有装进内场的，要等4-10号有动作了才进行操作

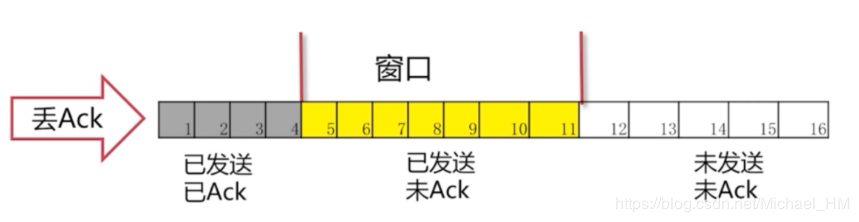
正常情况：



可以看到4号包已经被接收到，所以涂成了灰色，窗口就往右移一格，这里只有保证窗口是7格，所以我们就把11号包读取进来我们的缓存，进入待发送的状态，89号包已经变成了黄石，表示已经发送出去了，接下来的操作都是一样的了，确认包后，窗口就继续往后移动继续去将未发送的包读取进缓存，把待发送的状态转为已发送的状态

#### 丢包情况

有可能我们包发过去了，对方的ack丢了，也有可能我们的包并没有发送过去，从发送方的角度就是我们没有收到ack，发生的情况就是一直在等ack，我们也会把读进缓存的待发送的包也一起发送过去，但是这个时候我们的窗口已经满了，所以并不能把12号包读进来，而是始终在等5号包的ack，如果这个ack一直都没有怎么办



#### 超时重发

鉴于上面的一直等ack的情况发生，我们就有了个解决方案，就是超时重传

这里有一点要说明，这个ack是要按顺序的，必须等到5的ack收到，才会把6-11的ack发送过去，这样就保证了滑动窗口的一个顺序，这个时候可以看出5号包已经收到ack包了

后面的6.7.8号也已经发送过ack了，窗口变继续往前走

