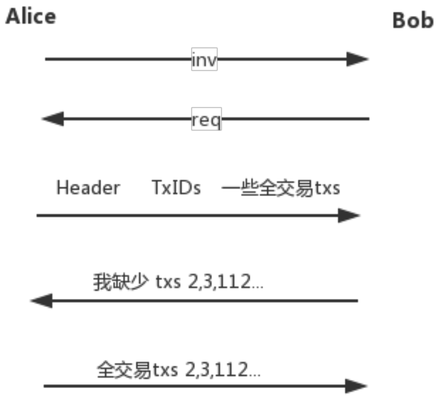
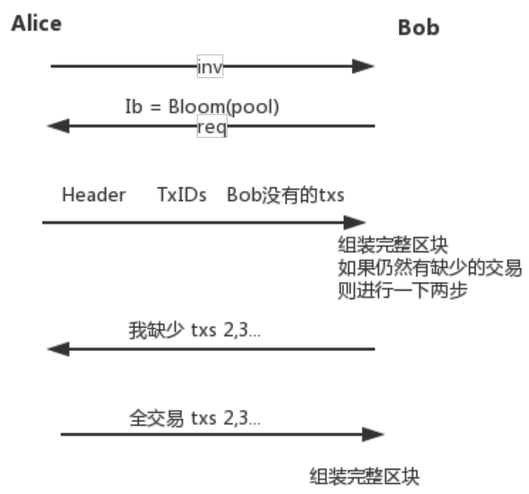
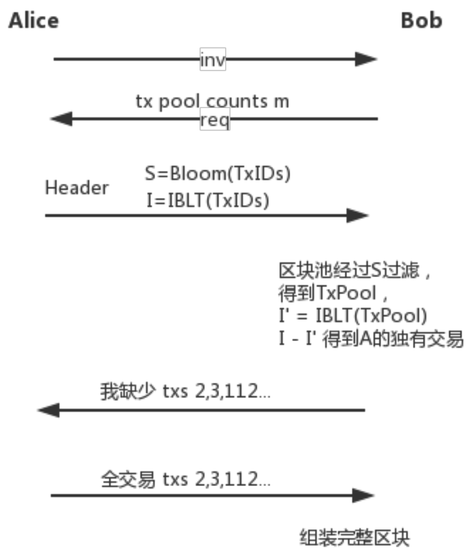
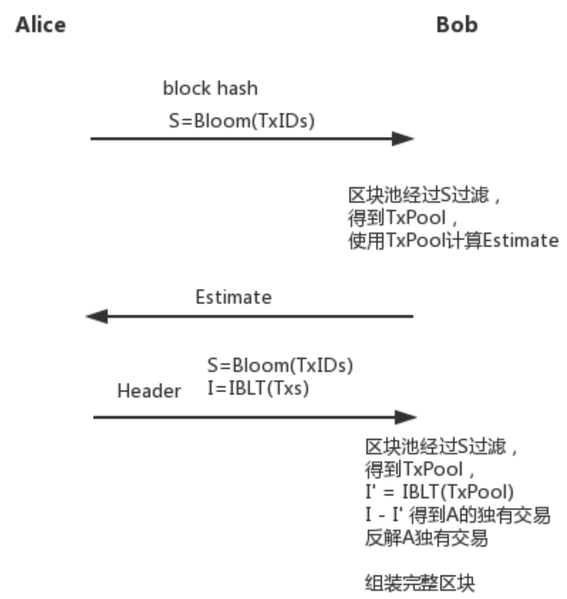
Compact Block 方案	Xthin 方案	石墨烯方案	ChainStack方案
			
数据传输量 （区块中有2000笔交易，Bob的交易池有4000笔交易，区块中有1800个交易与Bob的交易池重合， 每笔交易长100B, 交易ID截取5B就够了） (我们设定Bloom过滤器的容错率f = 0.01577, 套用公式，布隆过滤器的大小为x = y × (-ln(f))/ln(2)^2 × 1/8) = y * 1.079618			
Header （忽略） 所有TxID （2000 × 5 = 10KB） 缺少的交易ID （200 × 5 = 1KB） 缺少的全交易 （200 × 100 = 20KB） **总量：31KB （区块200KB）**	Header （忽略） 布隆Ib (4000 × 1.079 = 4.3KB) 所有TxID （2000 × 5 = 10KB） 缺少的全交易 （200 × 100 = 20KB） 忽略最后的缺少交易ID传输 **总量：34.3KB**	Header （忽略） 布隆S（2000× 1.079 = 2.1KB) IBLT I （1.5（2+4+5）× 预期差异 = 16.5 × 200 = 3.3KB） 缺少的交易ID （200 × 5 = 1KB） 缺少的全交易 （200 × 100 = 20KB） **总量：26.4KB**	Header （忽略） 布隆S两次（2000× 1.079 × 2 = 4.2KB) Estimate （80×7×32+1800/128×32=18.3K） IBLT（Txs）（1.5（2+4+100）× 预期差异 = 159 × 200 = 31.8KB/47.7KB） **总量：54.2KB**
Hash次数 **总Hash次数：0**	Bob 产生区块Bloom,Ib （4000） Alice 区块交易过滤Ib （2000） **总Hash次数：6,000 次**	Alice 产生区块Bloom, S （2000） Alice 产生区块IBLT, I （2000） Bob 区块池过滤S（4000 ） Bob 产生剩余交易的IBLT （1800） **总Hash次数：9,800 次**	Alice 产生区块Bloom, S （2000） Bob 区块池过滤1次S （4000） Bob Estimate （1800 × 2 = 3600） Alice 计算差异 （2000 × 2 = 4000） Alice 产生区块IBLT （2000 × 3 = 6000） Bob 区块池过滤1次S （4000） Bob 产生剩余交易的IBLT （1800 ×3 = 5400） **总Hash次数：29,000 次**

总结：

推测慢的原因在于：1. 两次遍历交易池，交易池有锁，遍历和等待锁释放都需要时间。

2. Hash计算量大

建议：1. 考虑只遍历一次交易池，优化协议

2. 改用石墨烯的方法，或者区块压缩的方法。