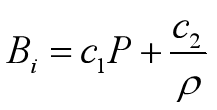
2.*隐私数据的上链存储过程中，在IOV场景下，参与共识的边缘计算节点的数量往往十分庞大，这使得经典的pbft共识协议的通信效率变得极其低下。li等人提出了一种双层的pbft共识协议，将原有的节点分组并分别进行上下两层共识。然而现有的双层共识pbft的上下层之间的共识是串行的，每一个新区块需要分别经过下层和上层的两次共识才能被最终确认，假设下层各个组每1s完成一次共识，上层每1s可以处理一次共识，那么上层每处理一次下层的共识，下层就会新产生n-1个新共识，这样下层的总共识产生速度大于上层，就会造成下层积压的共识越来越多，最终共识阻塞。*

本方案将下层一个组内的多轮共识结果批量提交到上层进行二次共识，并动态调整批量的数额。

具体介绍：

* 组内共识：组内共识相较原始pbft主要增加了主节点选举的过程，并在确认阶段将共识结果缓存在本地后直接开始新一轮共识。
* 选主节点：子CA计算组内节点的资产值B，从组内共识节点中选取最高的n个节点作为候选主节点。通知第一个候选节点担任本轮共识主节点。主节点回复确认后CA广播至组内节点主节点ID。
* 预准备：主节点从消息队列中取出k条车辆请求，验证后打包成区块m/\*可以计算k个车的聚合签名\*/，生成预准备消息并广播。从节点验证主节点签名和车辆签名以及v和n，生成准备消息并广播。同时缓存m和D(m)。
* 准备：从节点验证收到的prepare消息中v和n以及D(m)，当收到超过2f+1prepare后广播确认消息
* 确认：从节点验证收到的commit消息中v,n,D(m),当收到超过2f+1commit后，将2f+1个签名聚合为一个签名C，生成一个证书Q作为区块已达成共识的证据，并将区块m和证书Q缓存在本地map中。然后开始下一轮共识。
* 组间共识：当当前视图的组内主节点发现本地缓存中已有N个已完成的共识结果时，会触发组间共识，上层CA按照与组内相同的方式选出组间主节点并通知各组主节点。

1.预准备：主节点a向上层主节点P发送含有N次共识结果的请求，其中R为N次的区块集合，R=(m1,m2...mn)。P收到后验证每个证书中的聚合签名Ci以及每个区块m的摘要和对应的D(m)是否一致。验证完成后，生成预准备消息并广播，从节点验证v和自身v是否一致，n是否在低水位和高水位之间以及在当前v中没有其他的编号n。

2.准备：从节点计算h=H(v,n,R)并缓存，并回复prepare消息给主节点P。P收到并验证v是否与自己一致，n是否在水位范围内。

3.确认：若收到超过2f+1个一致的prepare后，P将所有节点的签名聚合成聚合签名S，生成commit消息并广播。收到commit消息的从节点验证S和v，n以及h。

4.重确认：为了防止确认阶段由于组间主节点发送了少于一半commit后宕机导致各节点状态不一致，在确认后加入重确认阶段。各从节点在确认主节点的commit有效后，广播confirm消息。节点收到confirm并验证h。完成后各组长将S，R打包成reply消息广播至组内，组内各节点验证S以及R并依次执行上链。

5.组间同步：当完成一次组间共识时，各个组立即将组内缓存的头区块header中的前区块hash更新以防止出现分叉。

安全性和活性分析：1.组内共识的视图切换协议和原有pbft相同。可以保证安全性和活性不受损坏。由于组内节点最后会缓存相同的信息，当上层完成组间共识后本组内会同时验证并执行，所以下层满足安全性要求。2.组间共识中，无论主节点在哪个阶段宕机都会触发视图切换，在视图切换中，从节点与CA通信，由CA验证viewchange请求并生成聚合签名，并将下一轮视图主节点ID放在Newview消息中。从节点验证S，并进入下一视图重新处理。若主节点在commit阶段发生宕机且已发送的commit消息少于一半，这时由于已有节点收到了commit消息，其中验证了2f+1个prepare消息，所以无需进行视图更换，由该节点在重确认阶段广播confirm消息，其他节点只要收到一个confirm消息就说明R已经过共识。3.系统CA节点为一定安全的节点。所以检查点协议不再采用广播形式，而改变成所有节点向CA发送checkpoint，CA收到超过一半后生成聚合签名并一一回复。4.如果组间共识开始时，请求发起节点a未能向组间主节点P发送共识结果，则上层CA会在计时器超时后通知该组CA更换请求节点以保持系统的活性。安全性的证明。。。。。