

Consensus Algorithms in Wireless Blockchain System

1 Consensus Algorithm in Each Round

Algorithm 1 Stable Wireless Blockchain Protocol

```
1:  $\triangleright$  Initialization:
2:  $Sortition(PKs^r, S^r)$ 
3:  $Rds^r = GenerateRandomness(r, B_{hash}^{r-1}, sig_{final}^r)$ 
4:  $\triangleright$  Leader Election and Block Proposal:
5:  $result = BlockProposerSelection(sk, Rds^r)$ 
6: if  $result == True$  then  $\triangleright$  As Block Proposer
7:    $B^r = GenerateBlock(B^{r-1}, Tx)$ 
8:    $sig_{partial}^r = Sign(B_{hash}^r)$ 
9:    $broadcast(B^r, sig_{partial}^r)$  with probability  $p$ 
10: else  $\triangleright$  As Ordinary Nodes
11:   Waiting to receive new Block
12: end if
13:  $\triangleright$  Block Verification and Finalization:
14: while  $!finalized$  do  $\triangleright$  All Consensus Nodes
15:    $(B^r, Signs^r, sig_{full}^r, Tx) = RcvMSG()$ 
16:   /**Check the validation of new block
17:   if  $isValid(B^r)$  and  $VerifyBlockProposer(pk_{BP}, Rds^r)$  then
18:      $sig_v^r = GenerateSignature(B_{hash}^r, sk_v)$ 
19:   end if
20:   if  $isValid(sig_{full}^r)$  then
21:      $\sigma_F^r = sig_{full}^r$ 
22:      $broadcast(\sigma_F^r)$  with probability  $p$ 
23:      $Append(B^r, \sigma_F^r)$ 
24:      $finalized = True$ 
25:   else if  $Count(Signs^r) \geq \lceil \frac{N}{2} \rceil$  then
26:      $\sigma_F^r = RecoverFullSignature(Signs^r)$ 
27:      $broadcast(\sigma_F^r)$  with probability  $p$ 
28:      $Append(B^r, \sigma_F^r)$ 
29:      $finalized = True$ 
30:   else if  $sig_u^r \notin Signs^r$  then
31:      $Signs^r = AppendSignature(sig_u^r)$ 
32:   else if  $v$  did not broadcast its partial signature then
33:      $broadcast(sig_v^r)$  with probability  $p$ 
34:   else
35:      $broadcast(Tx)$  with probability  $p$ 
36:   end if
37:    $count = count + 1$ 
38:   if  $count > T$  then
39:      $count = 1$ 
40:     if Received  $T$  consecutive transactions in the past  $T$  rounds then
41:        $p = p * (1 + \delta)^{-1}$ 
42:        $T = T + 2$ 
43:     end if
44:   end if
45: end while
```

```

46: function RECNEWBLOCK( $m_B, \sigma_v$ )
47:   if  $\sigma_v \notin sigShares$  then
48:      $sigShares = AppendSignature(\sigma_v)$ 
49:   end if
50:   if  $Count(sigShares) > K$  then
51:      $FinalSig = RecoverFinalSig(sigShares)$ 
52:   else
53:      $FinalSig = null$ 
54:   end if
55:   return  $sigShares, FinalSig, B_v^{new}$ 
56: end function
57: function APPENDSIGNATURE( $\sigma_v$ )
58:   if  $\sigma_v \notin sigShares$  then
59:      $sigShares \leftarrow sigShares + \sigma_v$ 
60:   end if
61:   return  $sigShares$ 
62: end function

```
