# Consensus Algorithms in Wireless Blockchain System

## 1 Consensus Algorithm in Each Round

---
**Algorithm 1** The SWIB Blockchain Consensus Protocol for each node $v$

---
1: **while** true **do**
2:     //** Iteration for round $r$
3:     $\triangleright$ Initialization:
4:     $Rds^r = GenerateRandomness(r, B_H^{r-1}, sig_{full}^{r-1})$
5:     $\triangleright$ Consensus Process:
6:     Block Proposer Election();
7:     Block Verification();
8:     Block Finalization();
9:     $r = r + 1$
10: **end while**

---

---
**Algorithm 2** Block Proposer Election for each node $v$

---
1: $result, proof = $ Block Proposer Election$(sk, Rds^r)$
2: **if** $result == True$ **then**                   $\triangleright$ As Block Proposer
3:     $B^r = $ Generate Block$(B^{r-1}, Txs)$
4:     $sig_v^r = Sign(B_H^r)$
5:     $broadcast(B^r, proof, sig_v^r)$ with probability $p$
6: **else**                               $\triangleright$ As Ordinary Nodes
7:     receiving messages or $broadcast(Tx)$ with probability $p$
8: **end if**

---

**Algorithm 3** Block Verification and Finalization for each node $v$

1: **while** $!finalized$ **do**
2:     $B^r, Sig_u^r, proof, sig_{full}^r = RcvMSG()$
3:     //**Check the validation of new block
4:     $result = $ Verify Block Proposer$(pk_{BP}, proof, Rds^r)$
5:     **if** $isValid(B^r)$ and result **then**
6:         $sig_v^r = $ Generate Signature$(B_H^r, sk_v)$
7:     **end if**
8:     **if** $isValid(sig_{full}^r)$ **then**
9:         $AddSig(B^r, sig_{full}^r)$
10:         $Append(BC, B^r)$
11:         $finalized = True$
12:     **else if** $Count(Sigs^r) \geq \lceil \frac{N}{2} \rceil$ **then**
13:         $sig_{full}^r = $ Recover Full Signature$(Sigs^r)$
14:         $broadcast(sig_{full}^r)$ with probability $p$
15:         $AddSig(B^r, sig_{full}^r)$
16:         $Append(BC, B^r)$
17:         $finalized = True$
18:     **else if** $sig_u^r \notin Signs^r$ **then**
19:         Append Signature$(Sigs^r, sig_u^r)$
20:     **else if** $v$ did not broadcast its partial signature **then**
21:         $broadcast(sig_v^r)$ with probability $p$
22:     **else**
23:         $broadcast(Tx)$ with probability $p$
24:     **end if**
25:     $count = count + 1$
26:     **if** $count > T$ **then**
27:         $count = 1$
28:         **if** Received $T$ consecutive transactions in the past $T$ rounds **then**
29:             $p = p * (1 + \delta)^{-1}$
30:             $T = T + 2$
31:         **end if**
32:     **end if**
33: **end while**

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