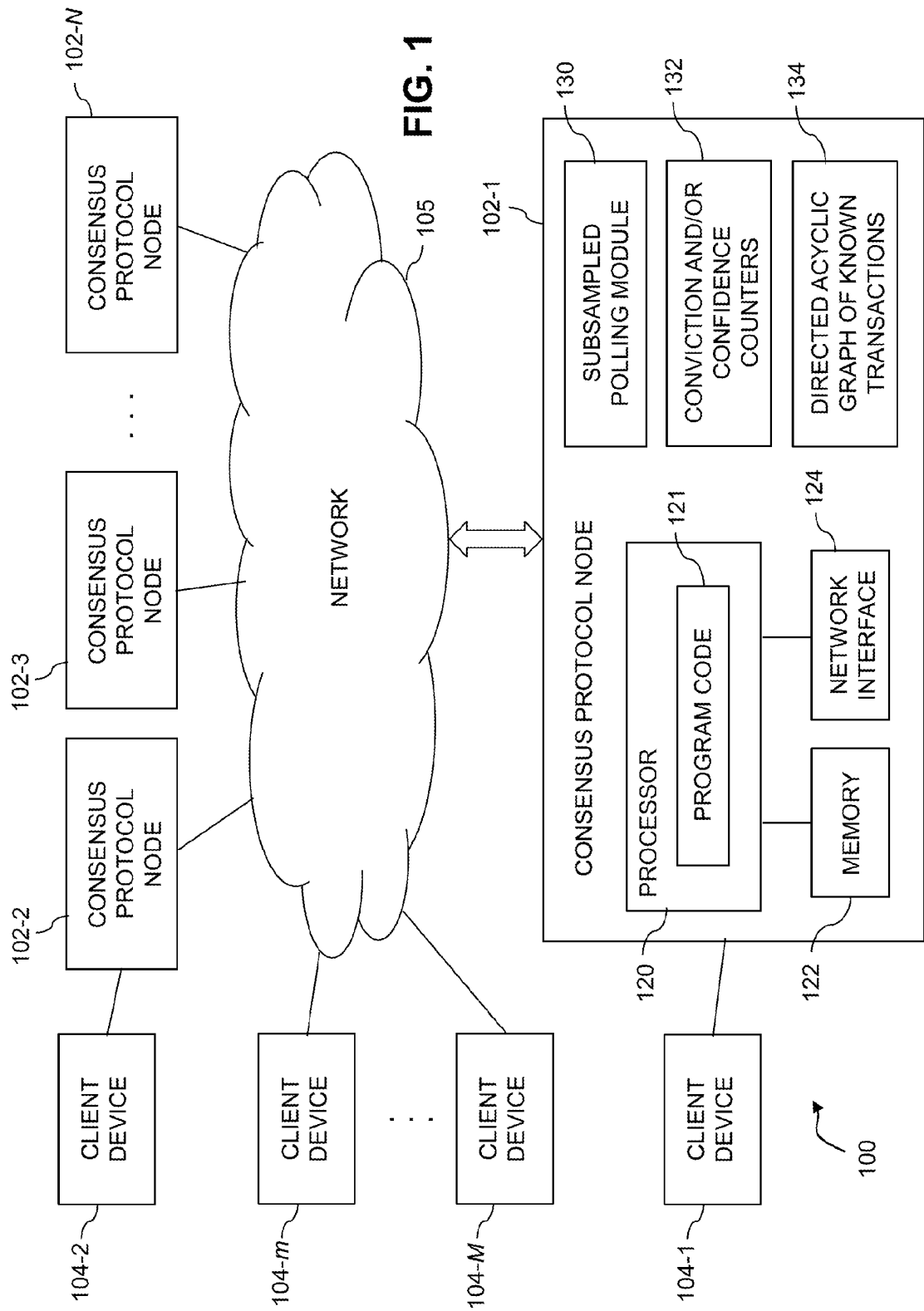




```

1: procedure ONQUERY( $v, \text{col}'$ )
2:   if  $\text{col} = \perp$  then  $\text{col} := \text{col}'$ 
3:   RESPOND( $v, \text{col}$ )
4: procedure SLUSHLOOP( $u, \text{col}_0 \in \{\mathbf{R}, \mathbf{B}, \perp\}$ )
5:    $\text{col} := \text{col}_0$  // initialize with a color
6:   for  $r \in \{1 \dots m\}$  do
7:     // if  $\perp$ , skip until ONQUERY sets the color
8:     if  $\text{col} = \perp$  then continue
9:     // randomly sample from the known nodes
10:     $\mathcal{K} := \text{SAMPLE}(\mathcal{N} \setminus u, k)$ 
11:     $P := [\text{QUERY}(v, \text{col}) \quad \text{for } v \in \mathcal{K}]$ 
12:    for  $\text{col}' \in \{\mathbf{R}, \mathbf{B}\}$  do
13:      if  $P.\text{COUNT}(\text{col}') \geq \alpha \cdot k$  then
14:         $\text{col} := \text{col}'$ 
15:  ACCEPT( $\text{col}$ )

```

FIG. 2

```

1: procedure SNOWFLAKELOOP( $u, \text{col}_0 \in \{\mathbf{R}, \mathbf{B}, \perp\}$ )
2:    $\text{col} := \text{col}_0, \text{cnt} := 0$ 
3:   while undecided do
4:     if  $\text{col} = \perp$  then continue
5:      $\mathcal{K} := \text{SAMPLE}(\mathcal{N} \setminus u, k)$ 
6:      $P := [\text{QUERY}(v, \text{col}) \text{ for } v \in \mathcal{K}]$ 
7:     for  $\text{col}' \in \{\mathbf{R}, \mathbf{B}\}$  do
8:       if  $P.\text{COUNT}(\text{col}') \geq \alpha \cdot k$  then
9:         if  $\text{col}' \neq \text{col}$  then
10:            $\text{col} := \text{col}', \text{cnt} := 0$ 
11:         else
12:           if  $++\text{cnt} > \beta$  then ACCEPT( $\text{col}$ )

```

FIG. 3

```

1: procedure SNOWBALLLOOP( $u, \text{col}_0 \in \{\mathbf{R}, \mathbf{B}, \perp\}$ )
2:    $\text{col} := \text{col}_0, \text{lastcol} := \text{col}_0, \text{cnt} := 0$ 
3:    $d[\mathbf{R}] := 0, d[\mathbf{B}] := 0$ 
4:   while undecided do
5:     if  $\text{col} = \perp$  then continue
6:      $\mathcal{K} := \text{SAMPLE}(\mathcal{N} \setminus u, k)$ 
7:      $P := [\text{QUERY}(v, \text{col}) \text{ for } v \in \mathcal{K}]$ 
8:     for  $\text{col}' \in \{\mathbf{R}, \mathbf{B}\}$  do
9:       if  $P.\text{COUNT}(\text{col}') \geq \alpha \cdot k$  then
10:         $d[\text{col}']++$ 
11:        if  $d[\text{col}'] > d[\text{col}]$  then
12:           $\text{col} := \text{col}'$ 
13:        if  $\text{col}' \neq \text{lastcol}$  then
14:           $\text{lastcol} := \text{col}', \text{cnt} := 0$ 
15:        else
16:          if  $++\text{cnt} > \beta$  then ACCEPT( $\text{col}$ )

```

FIG. 4

```

1: procedure AVALANCHELOOP
2:   while true do
3:     find  $T$  that satisfies  $T \in \mathcal{T} \wedge T \notin \mathcal{Q}$ 
4:      $\mathcal{K} := \text{SAMPLE}(\mathcal{N} \setminus u, k)$ 
5:      $P := \sum_{v \in \mathcal{K}} \text{QUERY}(v, T)$ 
6:     if  $P \geq \alpha \cdot k$  then
7:        $c_T := 1$ 
8:       // update the preference for ancestors
9:       for  $T' \in \mathcal{T} : T' \leftarrow^* T$  do
10:        if  $d(T') > d(\mathcal{P}_{T'}.pref)$  then
11:           $\mathcal{P}_{T'}.pref := T'$ 
12:        if  $T' \neq \mathcal{P}_{T'}.last$  then
13:           $\mathcal{P}_{T'}.last := T', \mathcal{P}_{T'}.cnt := 0$ 
14:        else
15:           $++\mathcal{P}_{T'}.cnt$ 
16:       // otherwise,  $c_T$  remains 0 forever
17:        $\mathcal{Q} := \mathcal{Q} \cup \{T\}$  // mark T as queried

```

FIG. 5

```

1: procedure INIT
2:    $\mathcal{T} := \emptyset$  // the set of known transactions
3:    $\mathcal{Q} := \emptyset$  // the set of queried transactions
4: procedure ONGENERATETx(data)
5:   edges :=  $\{T' \leftarrow T : T' \in \text{PARENTSELECTION}(\mathcal{T})\}$ 
6:    $T := \text{Tx}(\text{data}, \text{edges})$ 
7:   ONRECEIVETx( $T$ )
8: procedure ONRECEIVETx( $T$ )
9:   if  $T \notin \mathcal{T}$  then
10:    if  $\mathcal{P}_T = \emptyset$  then
11:       $\mathcal{P}_T := \{T\}$ ,  $\mathcal{P}_T.\text{pref} := T$ 
12:       $\mathcal{P}_T.\text{last} := T$ ,  $\mathcal{P}_T.\text{cnt} := 0$ 
13:    else  $\mathcal{P}_T := \mathcal{P}_T \cup \{T\}$ 
14:     $\mathcal{T} := \mathcal{T} \cup \{T\}$ ,  $c_T := 0$ .

```

FIG. 6

```

1: function ISPREFERRED( $T$ )
2:   return  $T = \mathcal{P}_T.\text{pref}$ 
3: function ISSTRONGLYPREFERRED( $T$ )
4:   return  $\forall T' \in \mathcal{T}, T' \xleftarrow{*} T : \text{ISPREFERRED}(T')$ 
5: function ISACCEPTED( $T$ )
6:   return
      $((\forall T' \in \mathcal{T}, T' \leftarrow T : \text{ISACCEPTED}(T'))$ 
        $\wedge |\mathcal{P}_T| = 1 \wedge d(T) > \beta_1)$  // safe early commitment
      $\vee (\mathcal{P}_T.\text{cnt} > \beta_2)$  // consecutive counter
7: procedure ONQUERY( $j, T$ )
8:   ONRECEIVETX( $T$ )
9:   RESPOND( $j, \text{ISSTRONGLYPREFERRED}(T)$ )

```

FIG. 7

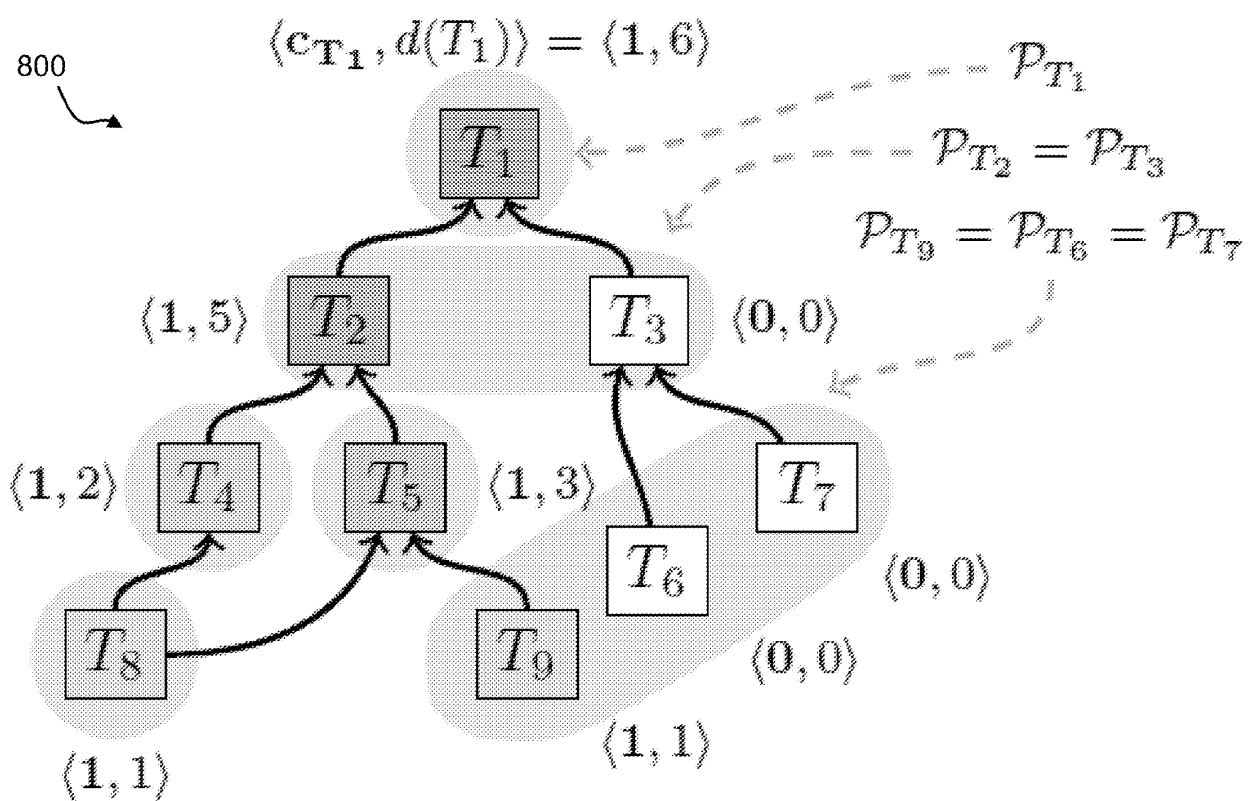


FIG. 8

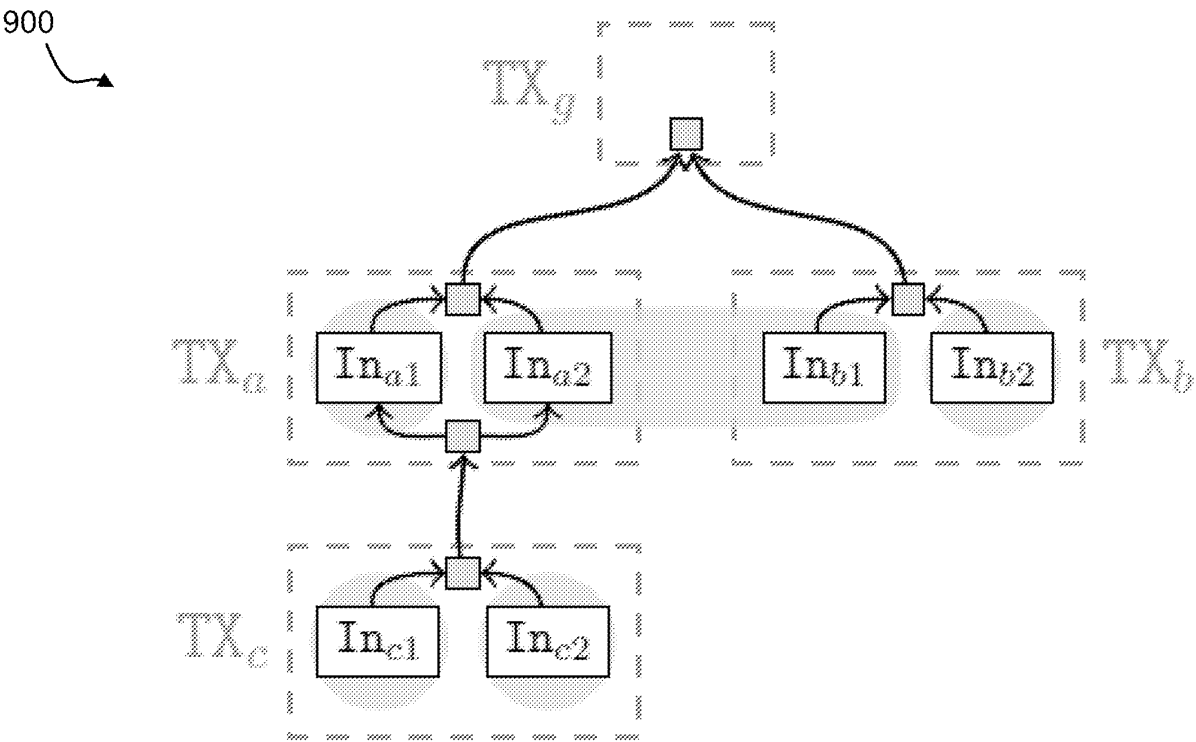


FIG. 9