Protocol Π_{Chan}

Initialisation:
    $State \leftarrow$ INIT

On (OPEN, c, `tx_out`, sk_A, pk_B) by $\mathcal{E}$:
    ensure $State =$ INIT
    $State \leftarrow$ OPENING BASE CHANNEL
    do LN (other box)

On (CHECK) by $\mathcal{E}$:
    ensure $State =$ WAITING FOR LEDGER
    send (READ) to $\mathcal{G}_{\text{Ledger}}$ and assign reply to Σ
    ensure $F \in \Sigma$
    $c_A \leftarrow c$; $c_B \leftarrow 0$ // c received in OPEN
    $State \leftarrow$ OPEN BASE
    output (OPEN SUCCESS) to $\mathcal{E}$

On (PAY, x) by $\mathcal{E}$:
    ensure $State \in \{$OPEN BASE, OPEN VIRTUAL$\}$
    ensure $c_A \geq x$
    do LN payment (these channels won't be async) (balance change here)
    output (PAY SUCCESS) to $\mathcal{E}$

On (CLOSE) by $\mathcal{E}$:
    if $State =$ OPEN BASE **then**
        prepare C TODO
        send (SUBMIT, C) to $\mathcal{G}_{\text{Ledger}}$
    else if $State =$ OPEN VIRTUAL **then**
        TODO
    end if

// notification to funder
// trust that *Alice* has c in her channel
On (FUND VIRTUAL, c, *Bob*) by *Alice*:
    ensure $State =$ INIT
    $State \leftarrow$ OPENING VIRTUAL CHANNEL
    do LN with *Bob* if he received (ALLOW VIRTUAL, c, *Bob*) (other box, hopefully same as the above)

// notification to fundee
On (ALLOW VIRTUAL, c, *Bob*) by *Alice*:
    ensure $State =$ INIT
    $State \leftarrow$ READY FOR VIRTUAL CHANNEL
    `funder` $\leftarrow$ *Bob*
    `coins` $\leftarrow c$

On (FUND, c, hops, sid_{Alice}, sid_{Bob}) by $\mathcal{E}$:
    ensure $State \in \{$OPEN BASE, OPEN VIRTUAL$\}$
    if do VChan with hops (other box) successful **then**
        send (FUND, c, sid_{Bob}) to sid_{Alice}
    end if

Fig. 1.

Functionality $\mathcal{F}_{\text{Chan}}$ – base

Initialisation: // runs on first activation
    $State \leftarrow$ INIT
    $(\text{locked}_A, \text{locked}_B) \leftarrow (0, 0)$

On (OPEN, c, `tx_out`, sk, $pk_{A,\text{out}}$, $pk_{B,\text{out}}$) by *Alice*:
    ensure $State =$ INIT
    $pk \leftarrow PK(sk)$; $(sk_{A,F}, pk_{A,F}) \leftarrow$ KEYGEN(); $(sk_{B,F}, pk_{B,F}) \leftarrow$ KEYGEN()
    $F \leftarrow$ TX {input: `tx_out`, output: $(c, 2/\{pk_{A,F}, pk_{B,F}\})$}
    $F \leftarrow F.\text{sign}(sk)$
    $State \leftarrow$ WAITING FOR LEDGER
    send (OPEN, F) to $\mathcal{A}$

On (CHECK) by $\mathcal{E}$:
    ensure $State =$ WAITING FOR LEDGER
    send (READ) to $\mathcal{G}_{\text{Ledger}}$ and assign reply to Σ
    ensure $F \in \Sigma$
    $c_A \leftarrow c$; $c_B \leftarrow 0$
    $State \leftarrow$ OPEN BASE
    output (OPEN SUCCESS) to $\mathcal{E}$

On (PAY, x) by $Dave \in \{Alice, Bob\}$:
    ensure $State \in \{$OPEN BASE, OPEN VIRTUAL$\}$
    ensure $c_D - \text{locked}_D \geq x$
    send (PAY, x, $Dave$) to $\mathcal{A}$ and expect reply (OK)
    $c_D \leftarrow c_D - x; c_{\bar{D}} \leftarrow c_{\bar{D}} + x$ // $\bar{D}$ is *Alice* if D is *Bob* and vice-versa
    output (PAY SUCCESS) to $Dave$

Fig. 2.

Functionality $\mathcal{F}_{\text{Chan}}$ – close

On (CLOSE) by *Alice*:
  if *State* = OPEN BASE **then**
    $C \leftarrow$ TX {input: F.out, outputs: $(c_A, pk_{A,\text{out}}), (c_B, pk_{B,\text{out}})$}
    $C \leftarrow C.\text{sign}(\text{sk}_{\text{A,F}}, \text{sk}_{\text{B,F}})$
    State $\leftarrow$ CLOSED
    input (SUBMIT, C) to $\mathcal{G}_{\text{Ledger}}$
  else if *State* = OPEN VIRTUAL **then**
    State $\leftarrow$ CLOSED
    output (CLOSING, c_A, c_B) to `opener`
  end if

On (CLOSING, c_{left}, c_{right}) by $\mathcal{F}_{\text{Chan}}$:
  ensure *State* $\in$ {OPEN BASE, OPEN VIRTUAL}
  ensure $((c_L, c_R), \text{hops}, (\mathit{Charlie}, \mathit{Dave}), (\mathit{Frank}, \mathit{George}), \text{id}) \in$ `funded` with *Frank* $\in \{\mathit{Alice}, \mathit{Bob}\}$
  ensure $c_{\text{left}} \leq c_L + c_R$
  remove entry from `funded`
  output (CLOSED VIRTUAL, c_{right}, id) to *Frank*

On (CLOSED VIRTUAL, c_{right}, id) by $\mathcal{F}_{\text{Chan}}$:
  ensure *State* $\in$ {OPEN BASE, OPEN VIRTUAL}
  ensure $(\text{virtual}, c, \mathcal{F}_{\text{Chan}}, \mathit{Dave}, \text{id}) \in$ `funded`
  ensure $c_{\text{right}} \leq c$
  send (CLOSED) to virtual and expect reply YES
  $c_D \leftarrow c_D + c_{\text{right}}$
  remove entry from `funded`

On (CLOSED) by P:
  if *State* = CLOSED **then**
    send (YES) to P
  else
    send (NO) to P
  end if

Fig. 3.

Functionality $\mathcal{F}_{\text{Chan}}$ – virtual

On (FUND YOU, c) by *Charlie*: // *Alice* is funded by *Charlie*
  ensure *State* = INIT
  relay message to $\mathcal{A}$ and ensure reply is (OK)
  $c_A \leftarrow c; c_B \leftarrow 0$
  `opener` $\leftarrow$ *Charlie*
  State $\leftarrow$ OPEN VIRTUAL
  output (OK) to *Charlie*

On (FUND, c, hops, `sub_parties` = (fundee, counterparty), `outer_parties` = (*Charlie*, *Dave*)) by *Alice*: // we fund another channel
  ensure *State* $\in$ {OPEN BASE, OPEN VIRTUAL}
  ensure $c_A - \text{locked}_A \geq c$
  $(L_0, R_0) \leftarrow (Alice, Bob)$
  generate random id
  for all $(L_i, R_i) \in$ hops **do** // $i \in \{1, \ldots, |\text{hops}|\}$
    ensure $R_{i-1} = L_i$
    send (ALLOW FUND, c, `sub_parties`, `local_funder` = L_i, id, $i \stackrel{?}{=} |\text{hops}|$) to L_i as *Alice* and ensure reply is (OK)
  end for
  $c_A \leftarrow c_A - c$
  add ((c, 0), hops, `sub_parties`, `outer_parties`, id) to `funded`
  input (FUND YOU, c, counterparty) to fundee as *Alice* and ensure output is (OK)
  output (OK) to *Alice*

On (ALLOW FUND, c, `sub_parties`, D, id, `is_last`) by *Charlie*:
  ensure *State* $\in$ {OPEN BASE, OPEN VIRTUAL}
  ensure $D \in \{Alice, Bob\}$
  ensure $c_D - \text{locked}_D \geq c$
  output received message to D and ensure reply is (OK)
  $\text{locked}_D \leftarrow \text{locked}_D + c$
  if `is_last` **then**
    add ((0, c), $\perp$, `sub_parties`.reverse(), (*Dave*, $\perp$), id) to `funded`
  end if
  send (OK) to *Charlie*

Fig. 4.