ДОДАТОК А

ОРИГІНАЛЬНІЙ ОПИС ФУНКЦІОНАЛЬНОСТІ ВІДКРИТТЯ КАНАЛУ

Functionality $\mathcal{F}_{\text{PayNet}}$ - open

1: Upon receiving (openChannel, *Alice*, *Bob*, x, *tid*) from *Alice*:
2: ensure *tid* hasn't been used by *Alice* for opening another channel before
3: choose unique channel ID *fchid*
4: `pendingOpen` (*fchid*) $\leftarrow$ (*Alice*, *Bob*, x, *tid*)
5: send (openChannel, *Alice*, *Bob*, x, *fchid*, *tid*) to $\mathcal{S}$

6: Upon receiving (channelAnnounced, *Alice*, $p_{Alice,F}$, $p_{Bob,F}$, *fchid*, *pchid*, *tid*) from $\mathcal{S}$:
7: ensure that there is a `pendingOpen`(*fchid*) entry with temporary id *tid*
8: add "*Alice* announced", $p_{Alice,F}$, $p_{Bob,F}$, *pchid* to `pendingOpen`(*fchid*)

9: Upon receiving (checkForNew, *Alice*, *Bob*, *tid*) from *Alice*:
10: ensure there is a matching `channel` in `pendingOpen`(*fchid*), marked with "*Alice* announced"
11: (funder, fundee, x, $p_{Alice,F}$, $p_{Bob,F}$) $\leftarrow$ `pendingOpen` (*fchid*)
12: send (read) to $\mathcal{G}_{\text{Ledger}}$ as *Alice* and store reply to Σ_{Alice}
13: `checkClosed`(Σ_{Alice})
14: ensure that there is a TX $F \in \Sigma_{Alice}$ with a $(x, (p_{\text{funder},F} \wedge p_{\text{fundee},F}))$ output
15: mark `channel` with "waiting for fundingLocked"
16: send (fundingLocked, *Alice*, Σ_{Alice}, *fchid*) to $\mathcal{S}$

17: Upon receiving (fundingLocked, *fchid*) from $\mathcal{S}$:
18: ensure a `channel` is in `pendingOpen`(*fchid*), marked with "waiting for fundingLocked" and replace mark with "waiting for channelOpened"
19: send (read) to $\mathcal{G}_{\text{Ledger}}$ as *Bob* and store reply to Σ_{Bob}
20: `checkClosed`(Σ_{Bob})
21: ensure that there is a TX $F \in \Sigma_{Bob}$ with a $(x, (p_{\text{funder},F} \wedge p_{\text{fundee},F}))$ output
22: add `receipt`(`channel`) to `newChannels`(*Bob*)
23: send (fundingLocked, *Bob*, Σ_{Bob}, *fchid*) to $\mathcal{S}$

24: Upon receiving (channelOpened, *fchid*) from $\mathcal{S}$:
25: ensure a `channel` is in `pendingOpen`(*fchid*), marked with "waiting for channelOpened" and remove mark
26: offChainBalance (funder) $\leftarrow$ offChainBalance (funder) + x
27: onChainBalance (funder) $\leftarrow$ onChainBalance (funder) $-$ x
28: `channel` $\leftarrow$ (funder, fundee, x, 0, 0, *fchid*, *pchid*)
29: add `channel` to `channels`
30: add `receipt`(`channel`) to `newChannels`(*Alice*)
31: clear `pendingOpen`(*fchid*) entry

ДОДАТОК Б

ОРИГІНАЛЬНІЙ ОПИС ФУНКЦІОНАЛЬНОСТІ ЗДІЙСНЕННЯ ПЛАТЕЖІВ ПО КАНАЛУ

Functionality $\mathcal{F}_{\text{PayNet}}$ - pay

1: Upon receiving $\left(\text{PAY}, Bob, x, \overrightarrow{\texttt{path}}\right)$ from $Alice$:
2: choose unique payment ID $payid$
3: add $\left(Alice, Bob, x, \overrightarrow{\texttt{path}}, payid\right)$ to `pendingPay`
4: send $\left(\text{PAY}, Alice, Bob, x, \overrightarrow{\texttt{path}}, payid, \text{STATE}, \Sigma\right)$ to $\mathcal{S}$

5: Upon receiving (UPDATE, `receipt`, $Alice$) from $\mathcal{S}$:
6: add `receipt` to `updatesToReport`($Alice$) // trust $\mathcal{S}$ here, check on RESOLVEPAYS
7: send (CONTINUE) to $\mathcal{S}$

ДОДАТОК В

ОРИГІНАЛЬНІЙ ОПИС ФУНКЦІОНАЛЬНОСТІ ВІДКРИТТЯ КАНАЛУ

Functionality $\mathcal{F}_{\text{PayNet}}$ - close

1: Upon receiving (CLOSECHANNEL, `receipt`, *tid*) from *Alice*
2: ensure that there is a `channel` $\in$ `channels` : `receipt` (`channel`) = `receipt` with ID *tid*
3: retrieve *fchid* from `channel`
4: add (*fchid*, `receipt`(`channel`), $\perp$) to `pendingClose`(*Alice*)
5: do not serve any other (PAY or CLOSECHANNEL) message from *Alice* for this channel
6: send (CLOSECHANNEL, `receipt`, *tid*, *Alice*) to $\mathcal{S}$