State Channel Security Considerations & Solutions

Joseph Poon (潘志豪)
<joseph@lightning.network>

On the blockchain, adversarial economic considerations are security considerations

Off-Chain Contract Consensus. Disagreement about current state of the contract can occur if the contract isn't designed optimally. "Double spend problem"

Penalties Need to be Predefined. Tort isn't a thing (absent an arbitration clause), you should construct fidelity-bonds beforehand to enforce off-chain behavior.

Costs of Enforcement. Much like how burdensome legal costs are a factor (in some cases, feature) in court, this is a major factor for off-chain contracts. It is easy to make 100-page legal documents, enforcing it in the court of law is annoying.

Off-Chain Contract Consensus

Consensus Fault between Channel Participants. If two parties disagree on the state, that's a problem. If one person is acting faulty, whatever. *But they can both be right!*

Natural Solution? Go to the Blockchain! Each party competitively tries to get their preferred state on-chain as soon as possible. This results in significantly greater costs in contract enforcement, and a high degree of unpredictability.

Using Only Simple Constructions helps prevent this problem. Only allow one party to attest to a certain state on-chain, else a timeout by the other party. Multiple possible attestations by 2+ parties means you're going to have a bad time.

Penalties Need to be Predefined

You can’t break an Autonomous AI’s kneecaps because they don’t have knees. If someone’s acting faulty or outright abusive, you need some kind of penalties to enforce good behavior. Define it.

Fidelity bonds. Lots of em. If it’s on-chain, you know the current state and can run the rounds live. Off-chain, you need to enforce good behavior.

Goal: Create economic incentives to stay off-chain. Otherwise, you lose the efficiency gains.

Costs of Enforcement: Systemic Risks

Mass-spam. On-chain enforceability before time periods are assumed. Contract terms can change if you cannot get your state updates in time.

Higher Costs During These Periods. Natural solution is to pay for block inclusion however possible. Economic rent from forcing channel closure.

Use Relative Timestamps! If you're enforcing posting of fidelity bonds, make sure to use relative dates, costs should be more predictable that way.

All Fidelity Bonds Need to Resolve This. On-chain or off-chain, the ability to attest to a state is a significant factor in agreements.

Cost of Enforcement: Modular Contracts

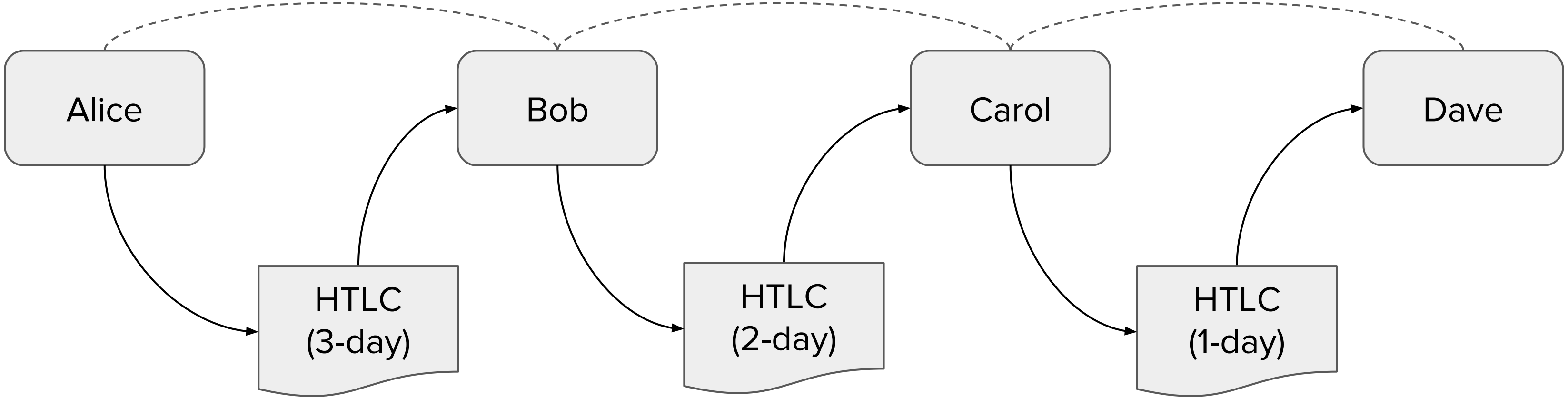
Keep it Simple. A contract can be split into many contracts. More of a UTXO model (modular). Makes chains of contract flows easier.

Network Costs. When computing across a network of participants, costs multiply.

Off-Chain Must Still Resolve High-Costs of Blockchain Enforcement. Don’t ignore the need to enforce as an eventuality, resolve the costs in doing so.

Fidelity Bonding Example. Lightning paper was about enforcing off-chain state (balances between participants). This construction is useful in a generic sense.

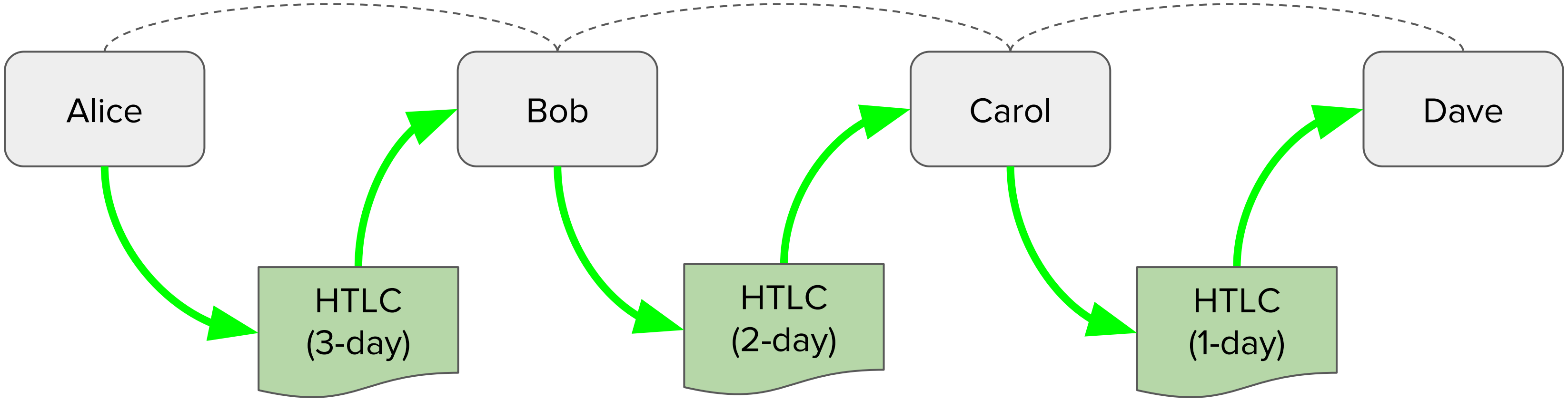
HTLC - Alice Makes a Contract (T0)



Decrementing timeouts, standard HTLC.
For Alice&Bob’s 3-day timeout HTLC, either party can broadcast that state on-chain.

(dotted lines are communication lines between parties)

HTLC - Off-Chain Cooperative Computation (T0)

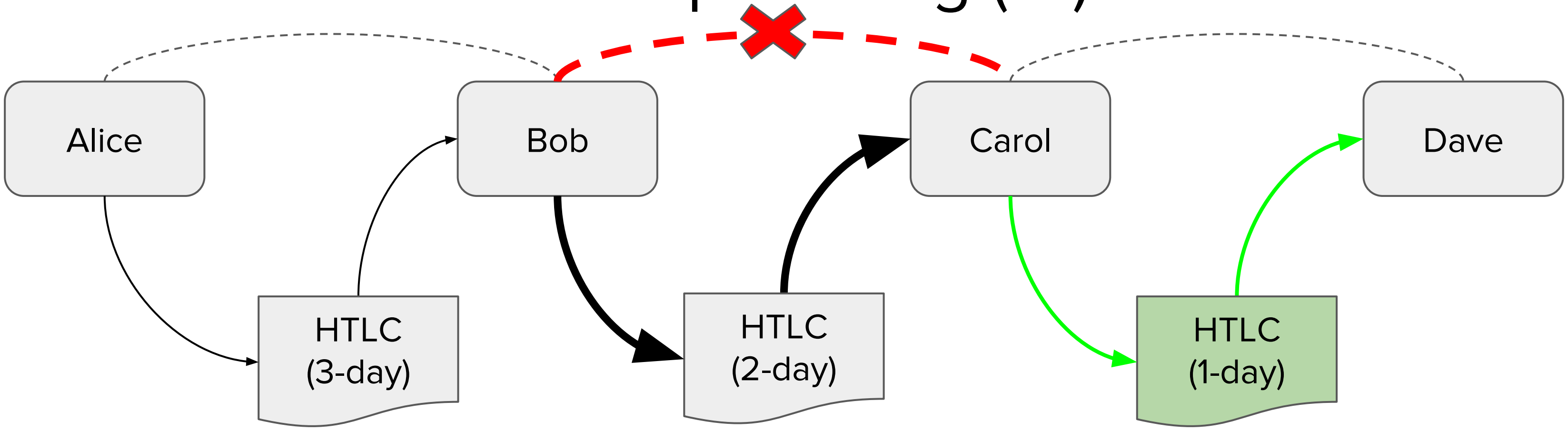


Green = Off-Chain

If one’s neighbors accept the computed state and are online, then it’s possible to close it all out off-line and pay-out according to the terms of the contract. This is provided by a proof of the inputs and resulting redemption.

(NOTE: It may not be secure to allow the sender to assert state, only the receiver should do so, otherwise there could be local-state consensus faults and it’ll be pushed on-chain if 2 states are valid)

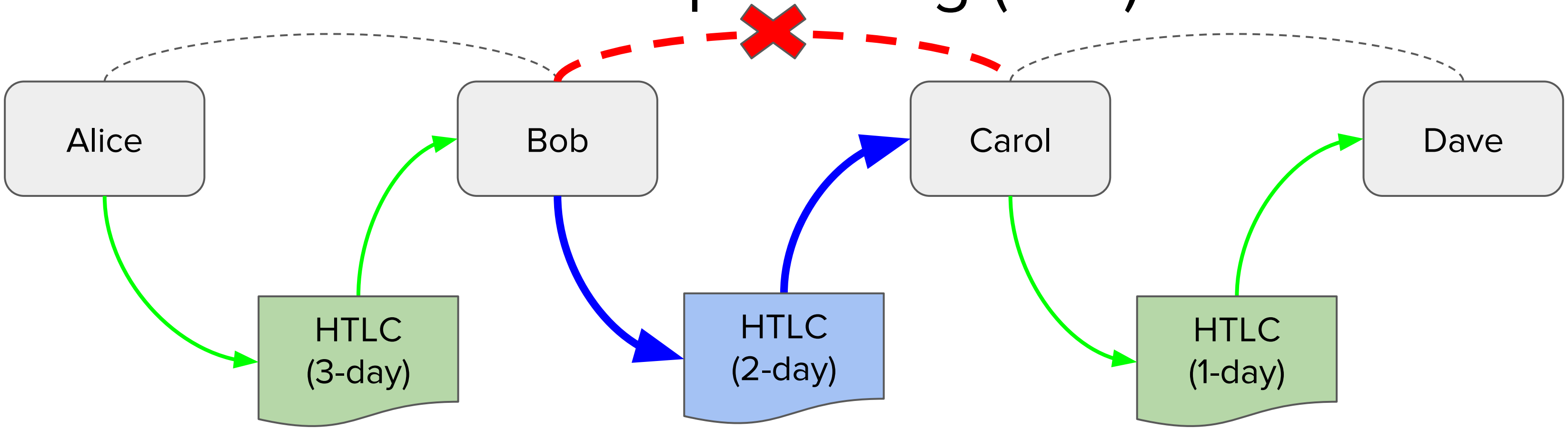
HTLC - Carol and Bob Stop Talking (T1)



Let’s presume instead of redeeming off-chain that Carol and Bob stop communicating. The only way for Carol to redeem her funds is by going to the blockchain!

(dotted lines are communications between parties)

HTLC - Carol and Bob Stop Talking (<T3)



Green = Off-Chain
Blue = On-Chain

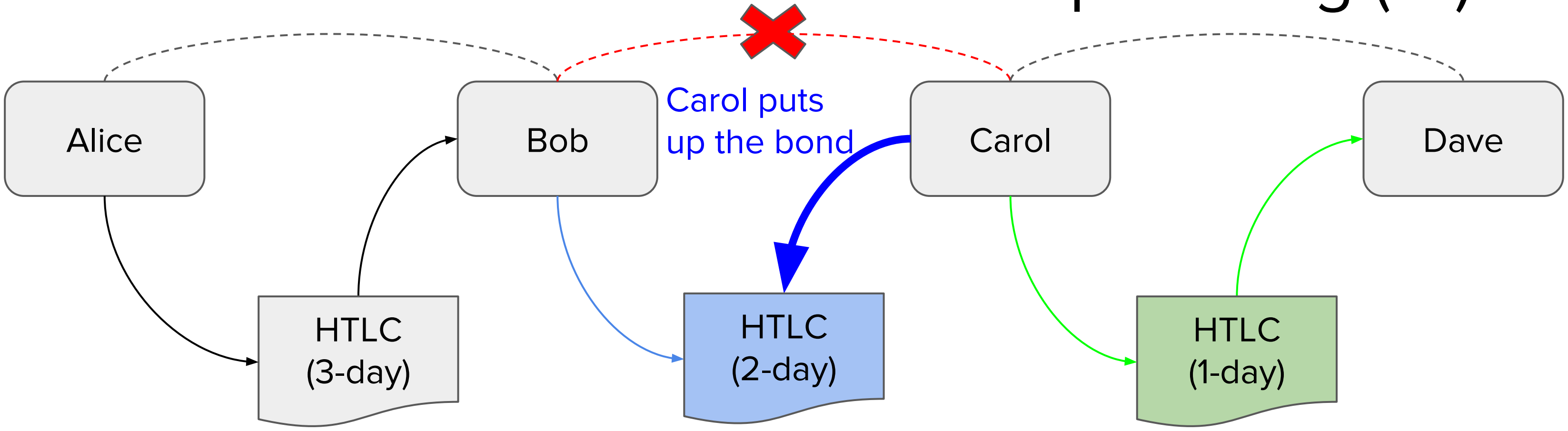
If Bob and Carol stop communicating, their contract must be on-chain. Everything else is off-chain. The contract gets computed and Carol gets her funds.

Instead of computing the network contract state on-chain, just attest the result and give other participants the ability to dispute.

What is computed should instead be bonded with inputs and output states.

With a TrueBit-like construction
https://medium.com/@chriseth/truebit-c8b6a129d580

Bonded Assertion - Carol and Bob Stop Talking (T1)

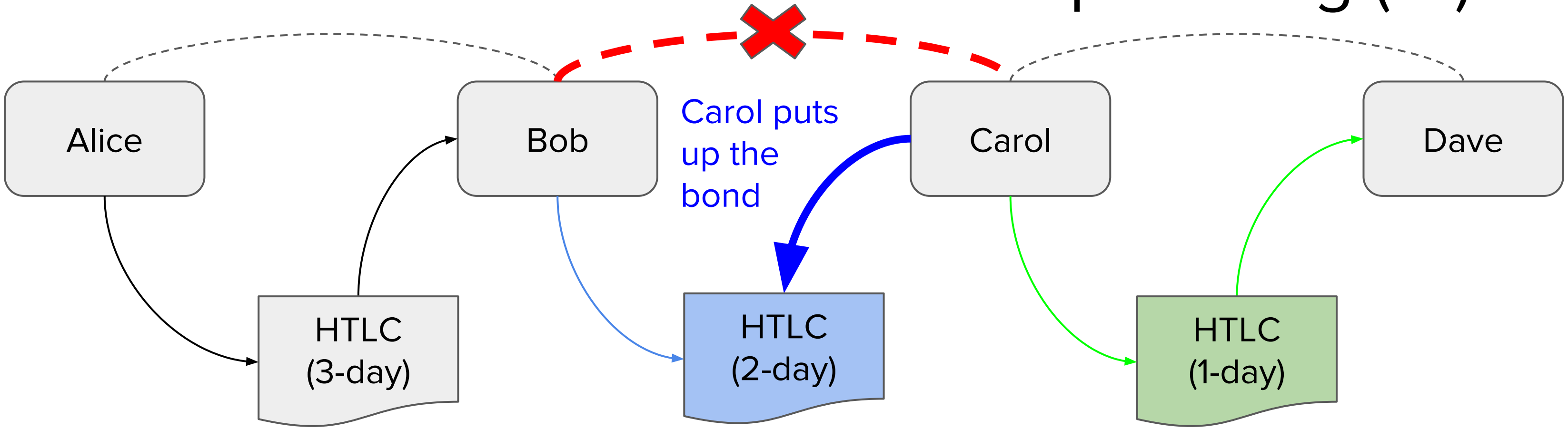


HTLC in this case is a “Pull funds or timeout” contract
Blue = On-blockchain
Light-Blue = Spent on-chain

When Carol closes out her channel on-chain by broadcasting all contracts/payments in-transit, she doesn’t force on-blockchain computation of the contracts. She just broadcasts the data which isn’t shared with Bob, and a bond asserting the contract result.

Carol is attesting to a particular computed result, and saying “If I’m wrong, we can do the full computation on-chain and you can claim the bond”

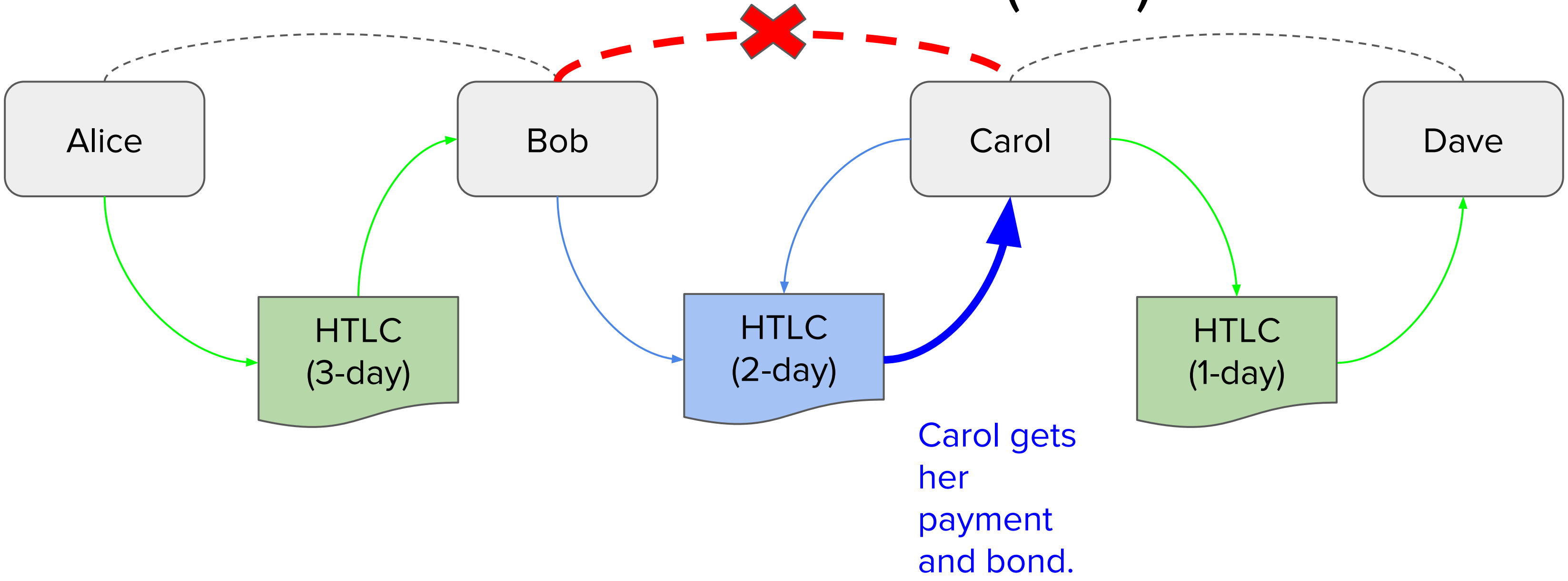
Bonded Assertion - Carol and Bob Stop Talking (T1)



Blue = On-blockchain
Light-Blue = Spent

Assume all bonds must be redeemed within 1 day relative of on-chain broadcast.

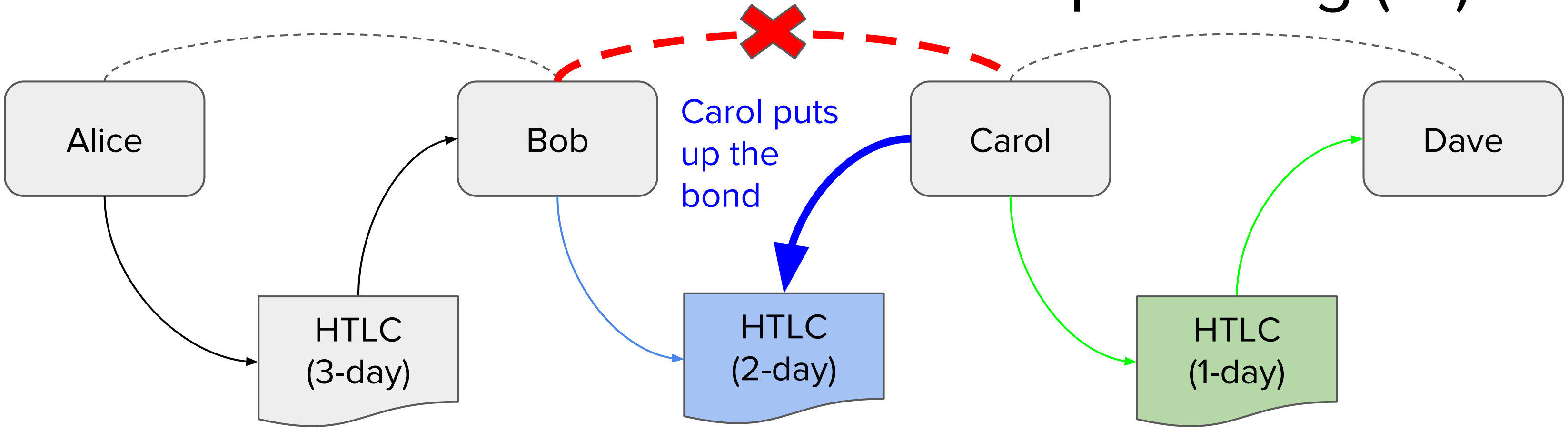
Bonded Assertion - Carol is Correct (<T4)



If Carol’s assertion is correct, she gets her money as part of the original HTLC, as well as her bond is refunded.

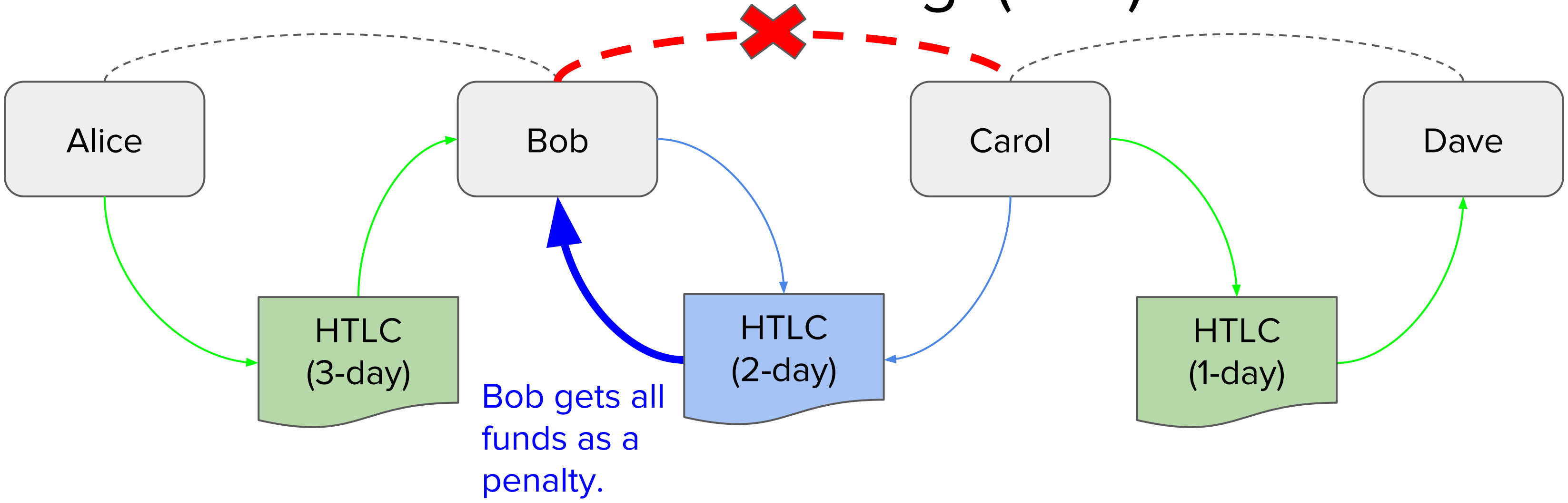
Alice sent the money to Bob (using the computed data from the on-chain Bond)

Bonded Assertion - Carol and Bob Stop Talking (T1)



Blue = On-blockchain
Light-Blue = Spent

Bonded Assertion - Carol is Wrong! (<T4)



If Carol’s assertion is incorrect, Bob makes the blockchain actually compute the full contract and he claims the HTLC and the Bond.

(A further bond can be attached to Bob forcing on-chain computation)

Third parties can put up the bond as well when closing out off-chain contracts. They just need to have the contract, inputs, and output and attest on behalf of others (for a fee).

Reduce/eliminate the need to do full on-chain computation of complex contracts, even in the event of on-chain enforcement of off-chain states. Necessary for confidence in off-chain computation.

Thanks!

Joseph Poon (潘志豪)
<joseph@lightning.network>