Lessons from the ILP2/3 experiments

Evan Schwartz

Starting point:

1. What is the right architecture for Interledger?
2. How do we build a real money open Interledger ASAP?

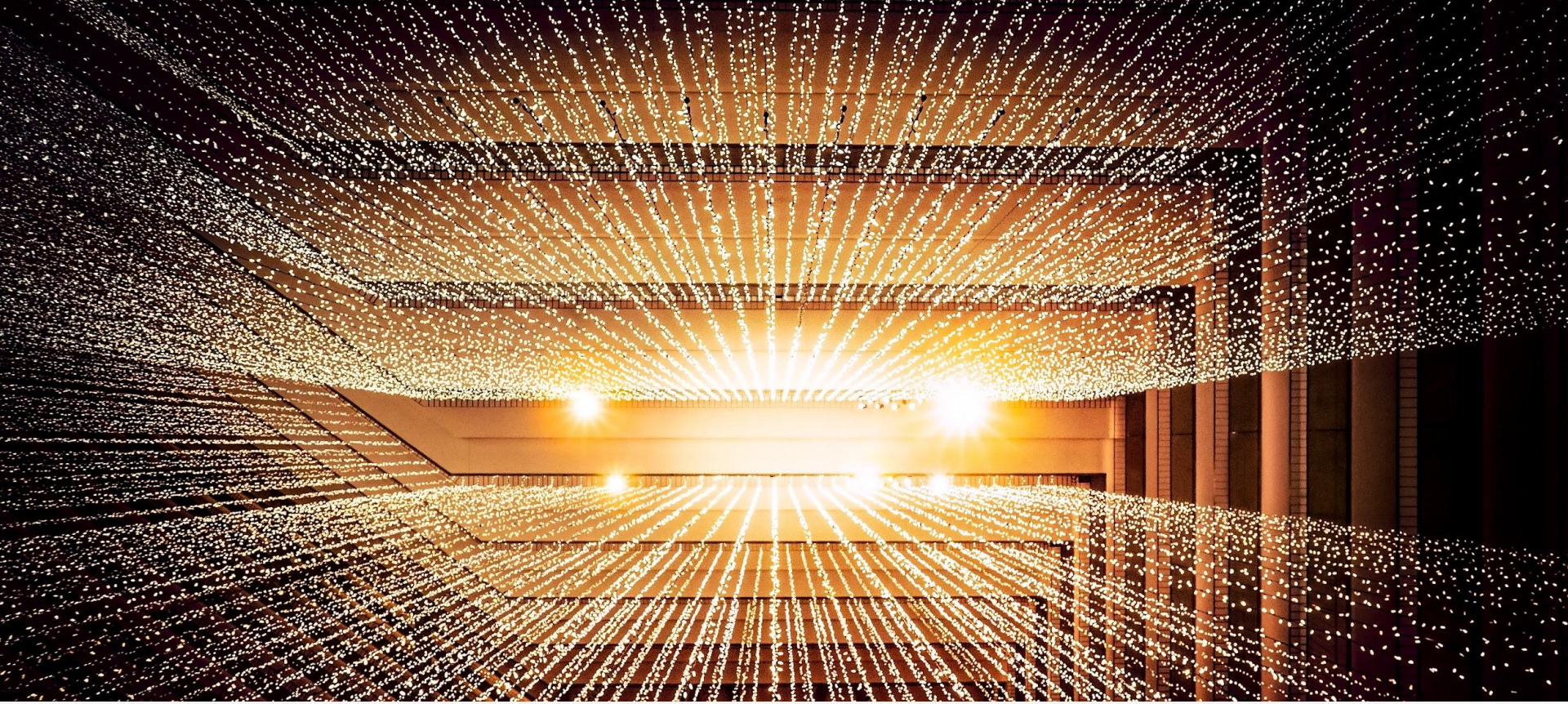
Fast Hashed Timelock Agreements (HTLAs) Only

Simple payment channels:

- balance complexity and risk
- are supported by major cryptocurrencies
- connectors don’t custody funds

Connectors will limit payment size and amount of money in flight

Every path will have a
Maximum Payment Size (MPS)
like the Internet's
Maximum Transmission Unit (MTU)



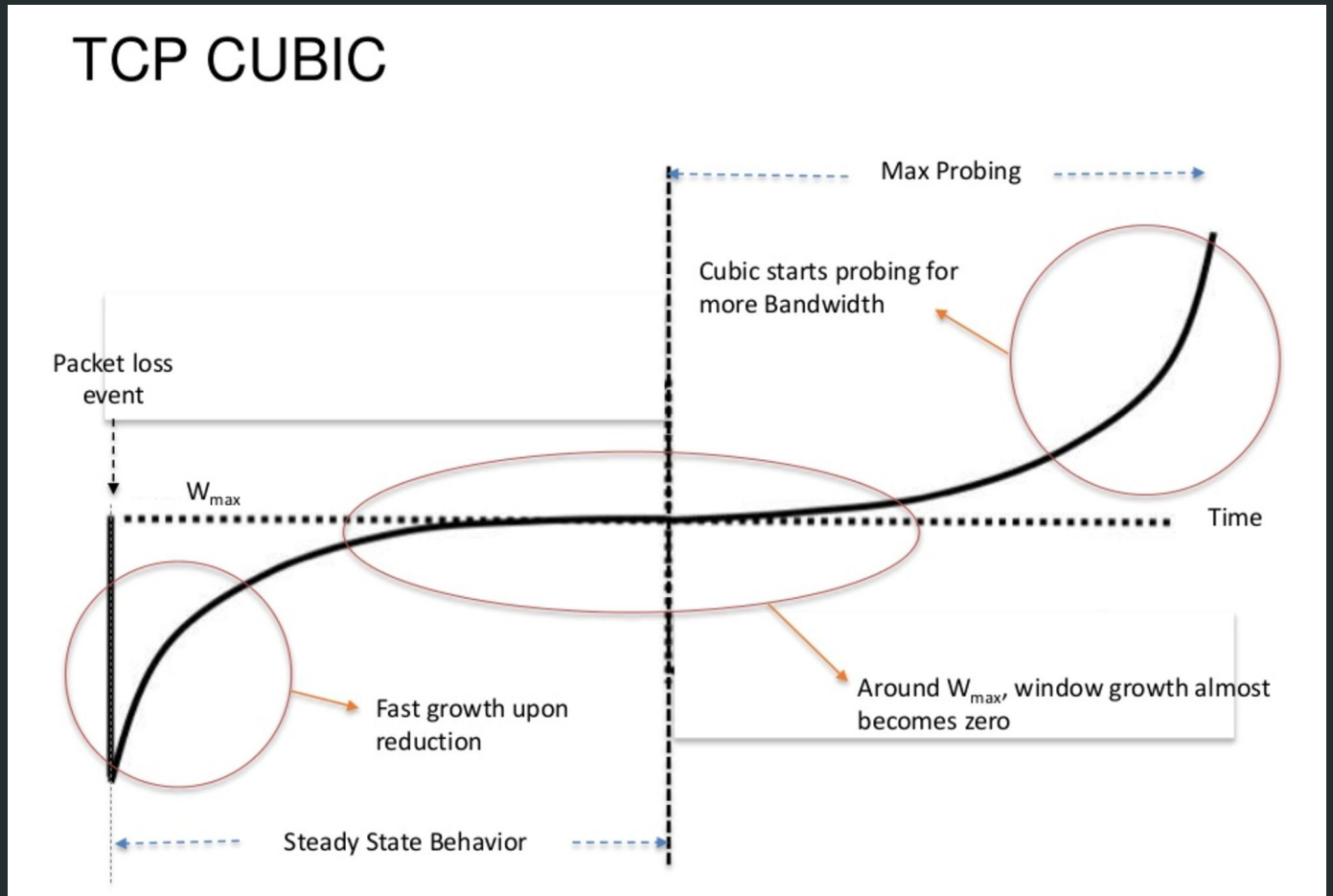
Build for small payments. **Many** small payments.

“Streaming payments change everything”

- Stefan Thomas

Chunked payments will
make it so users don’t
notice the low MPS,
just like TCP on the Internet

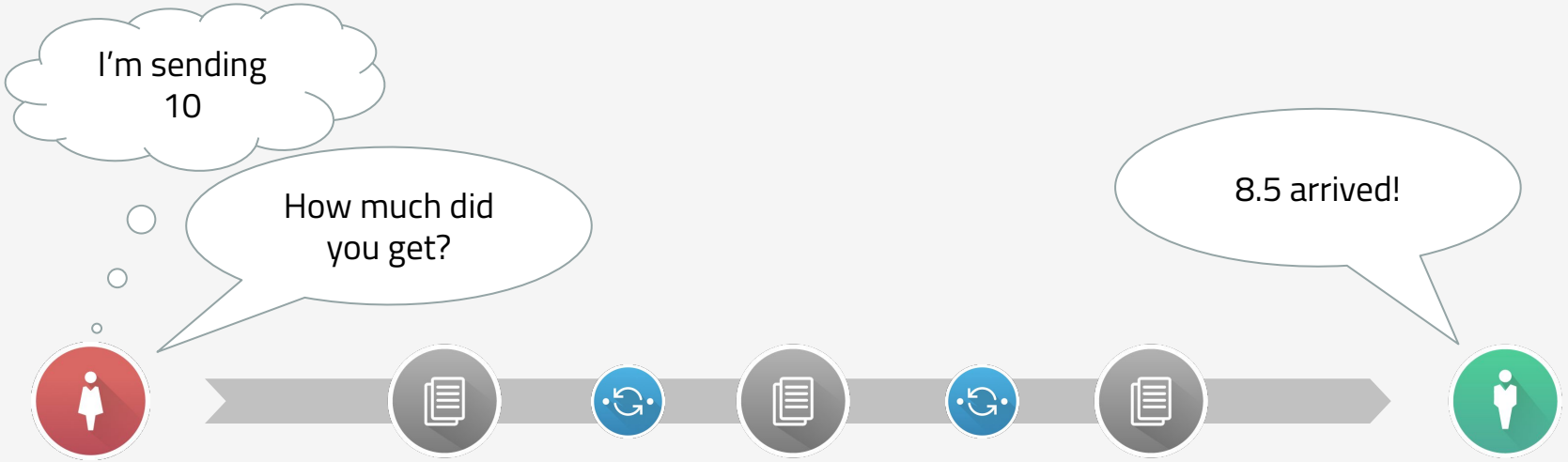
So similar that we may be able to use TCP CUBIC



With chunked payments,
you don’t know the exact destination amount
to put into the ILP packet

And quoting up-front with **ILQP doesn't help**, because you need to track the rate changing over time anyway

Proposal: Replace ILQP with End-to-End Quoting



Proposal: Remove Amount From ILP Packet

```
{
   account: "g.crypto.bitcoin.1BvBMSEY...",
   amount: "10",
   data: "SGVsbG8gV29ybGQhCg..."
}
```

Alternative: Make Zero Amount Indicate “Forward Only”

```
{
   account: "g.crypto.bitcoin.1BvBMSEY...",
   amount: "0",
   data: "SGVsbG8gV29ybGQhCg..."
}
```

If connectors only “forward”, they simply apply local rate to incoming transfer amount (no need to know others’ rates)

With only 2 fields in the ILP packet and without ILQP, **standard encoding (OER) is less important**

Destination ILP address and data can be included directly in ledger layer transfer object*

* Credit to Adrian Hope-Bailie for questioning the separation between the ILP packet and transfer

ILP could be as simple as:

```
POST / HTTP/1.1
ILP-Destination: g.crypto...
ILP-Amount: 10
ILP-Expiry: 2017-11-29T...
ILP-Condition: ax8j7s0ky5...

<transport data>
```

Transfer amount, not destination amount →

* Credit to Michiel de Jong for originally suggesting a text-based encoding for the ILP packet

When transfers are fast,
fulfillment is just the response*:

```
HTTP/1.1 200 OK
ILP-Fulfillment: 7sflc01sza...

<transport data>
```

* Credit to Michiel de Jong for suggesting making the fulfillment a response instead of a separate request

Prepared transfers
don’t need to be persisted, so
payment speed = network latency
+ in-memory processing*

* Credit to Michiel de Jong for this really awesome observation

"Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away"

… and that seems very close!

Open Questions

1. How large will the average Maximum Payment Size be?
2. Deprecate ILP packet w/ amounts or use zero amount?
3. Deprecate ILQP?
4. Do we need a standard packet encoding?
5. Recommend HTTP or Websocket (BTP) ledger protocol?
6. Can connectors automatically route based on peer rates?
7. Use a plugin- or “middleware function”-based architecture?