0.1 Links

A LINK is a bidirectional tunnel-element; an autonomous communication entity between *two* CELLs. Think of LINKs as compute elements with their own autonomous and independent failure domain. Physically, the LINK comprises the cable and SerDes' on both ends to form a self contained execution environment.

LINKs are autonomous in that they maintain state: pending transactions, reversibility buffers, sequence tracking, and retry logic. They mediate causality between two CELLs and enforce atomic delivery guarantees over physical media that may be noisy, lossy, or delayed.

A healthy LINK behaves like a lock-free memory bus: it transmits events, ensures ordering, and preserves invertibility for transactional safety. But unlike a memory bus, it must contend with delay, noise, and the limits of the speed of light. Its job is to conceal those imperfections behind a deterministic, reversible interface.

LINKs are not passive – they can be reset, throttled, or even reprogrammed in the field. They may expose telemetry, accept diagnostic pings, or reconfigure modulation in response to environmental conditions.

0.1.1 Link Utilities

Physical LINKs Implement utilities that used to be in logical link domains above L2: in L3, L4, or L7; composed into an abstraction of logical links. This is an illusion. If the pairing of Shannon information is thrown away at layer 2, it cannot be recovered in higher layers. This is addressed in more detail in the *Key Issue* section below.

An example[1] LINK utility is *The I Know That You Know That I Know* (TIKTYKTIK) property; which enables us to address some of the most difficult and pernicious problems in *distributed systems* today.

Another example LINK utility is *Indivisible Unit of Information* (IUI). Unlike *replicated* state machines (RSM's) used throughout distributed applications today, LINKs *are* state machines: the two halves of which maintain *shared state* through hidden packet exchanges. When a local agent or actor is ready, theIUI protocol transfers *indivisible* tokens across the LINK to the other agent, *atomically* (all or nothing) [2].

TIKTYKTIK and IUI properties are mathematically *compositional*.

What's necessary is an *entanglement* between state machines – locking them together silently in normal operation, and failing locally at the first failure. The entanglement cannot be recovered if information from events can disappear. This is the only solution to the problem in

[1] Synchronization of timing domains in computers generally start from the processor clock on the motherboard, and fan out through the logic into the I/O subsystems. IUI lives in the LINK between two *independent* computers, and although it receives information from either side, it is not synchronized with either side. This independent asynchronous domain (already exploited in the HFT Industry) – enables failure independence and atomicity.

[2] LINKs are *exquisitely* sensitive to packet loss. This is intentional: we turn the FLP result *upside down*, and use "a single unannounced process death" to guarantee the atomic property for IUI.

the latency–disconnection ambiguity [Ref: CAP Theorem Tradeoffs]. To put it in terms an engineer can internalize, a system that fails instantly, can heal immediately.

0.1.2 Failure Modes

The *shared state* property is strengthened by mechanisms to recover from each type of failure. The more types of failures, the more complex and intractable this becomes. `LINK`s are independent failure domains, with (effectively) one failure hazard: *disconnection* [3]; which is straightforward to recover from.

[3] In any physical system it is possible to drop packets, it will be much rarer but it is still possible. `LINK`s can recover from individually dropped or corrupted packets, and *shared state integrity* can be maintained through out the successive reversibility recovery – back to the equilibrium state.