0.1 A Blueprint for Æthernet Protocol

0.2 Foundations

We begin with Bartlett, Scantlebury and Wilkinson's (**?**) Alternating Bit Protocol (ABP).

Alternating messages are implemented as a single snake (a longitudinal set of bits traveling through the wire and FPGA's). Wrapping its way through the SerDes Registers of Alice and Bob on both ends.

First imagine a zero length wire connecting the chiplets for Alice an Bob (they are as adjacent as they can be on a module or motherboard). Two SerDes channels are directly connected.

Where in BSW[1], the edges of the automata are labeled with the 'alternation' bit, Lynch expands the single bit to multiple bits.

A single bit alternates, but it does not have a direction.

[Alternating 2-Bit Protocol (from Lynch):] Two bits at least have a direction in their evolution through their set.

{00 01 11 11}

OR

{11 10 01 00}

Where the next item establishes the direction of evolution of the set.

We use Ternary logic -1,0, +1 in our model of the protocol.

0 simply means 'equilibrium', which gives us the +1 and -1 complementary states that are expected of Ternary Logic. But there is a mathematical subtlety: we need both two versions of Zero, one approaching from the negative side, and one approaching from the positive side.

This provides a mechanism for *forward-propagation* and , of the state.

0.2.1 Proof of Fallibility

"the alternating validation bit becoming the additionally the *alternation* bit for message transmission in one direction, while the alternation bot for the reverse directions serves additionally as a validation bit" SBW **?**

0.3 Alternating Causality

QUESTION: What is the difference between reliable and fallible?

../Figures/BSW.pdf

Figure 1: Aternating Bit Protocol(ABP)

[1] A note on reliable full-duplex transmission over half-duplex links

Lynch's scheme is constructed from independent simplex procedures

0.4 Framing the Vocabulary

In 1967 *Bartlett, Scantlebury & Wilkinson* (BSW) sketched the *alternating-bit protocol* (ABP): add one history bit to every frame, wait for an ACK that echoes it, and retransmit until the right ACK appears. ABP wraps an *unreliable* medium and presents a service that looks reliable—even *infallible* in steady state.

Nancy Lynch later formalised these ideas with the *I/O-automaton*: a fallible channel is one whose execution may deviate from the specification, subject only to fairness.

Term	Meaning
Unreliable	Frames may be lost.
Fallible	Channel may violate any promise (drop, duplicate, reorder, corrupt).
Reliable	No drops in steady state; recovery still required.
Infallible	No violations ever; no recovery logic needed above the link.

Figure 2: Taxonomy of link qualities.

0.5 Where Classical Ethernet Fits

Original 10 Mb/s Ethernet (and most "best-effort" variants since) offers a CRC to *detect* corruption but no link-level retransmission. Frames can be dropped by congestion, policing, or topology loops. Hence raw Ethernet is both *unreliable and fallible*; higher layers—typically TCP—supply ABP-like recovery.

0.6 How InfiniBand Raises the Game

InfiniBand *embeds* ABP into silicon:

- Per-hop *credit flow control* makes buffer overflow almost impossible.
- Link-level CRC plus optional *link retransmission* retries any corrupted frame.
- Reliable Connection and Reliable Datagram queue pairs carry *ACK-/NACK* sequence numbers end-to-end, guaranteeing exactly-once, in-order delivery across multi-switch fabrics.

To software, the fabric appears nearly *infallible*; drops are rare and localized.

0.7 Reliable vs. Infallible, Unreliable vs, Fallible

Table **??** highlights the nuance. Priority Flow Control (PFC) can render an Ethernet link *loss-less* in steady state, but deadlock, mis-configuration, or burst congestion can still drop frames. Such a link is "reliable yet fallible." Infiniband's credit + retransmit pipeline, by contrast shifts real-world operation toward "reliable and almost infallible."

0.8 Why Ethernet Still Struggles

1. **Retrofitting**: inserting link retransmission into the IEEE 802 stack breaks long-standing timing and compatibility assumptions.
2. **Congestion domain**: shallow switch queues and ECMP paths leave more surfaces for loss than InfiniBand's strict hop-by-hop credits.

3. **Layering philosophy**: because TCP "already" ensures delivery, many operators accept occasional loss rather than pay silicon cost for hardware recovery.

0.9 Lessons from BSW and Lynch

- *Make reliability local*. ABP attaches one bit; InfiniBand embeds a few more. End-to-end recovery alone expands the failure scope.
- *Fail fast*. InfiniBand retransmits on explicit NACK within microseconds; Ethernet traditionally converts a microsecond drop into a millisecond TCP timeout.
- *Separate reliability from recovery*. Even a reliable link needs a failsafe plan; design that plan explicitly.

0.10 Conclusion

[CONCLUSION:] BSW showed that a single alternating bit can tame a capricious wire; Lynch supplied the proof rules. InfiniBand adopted both insights in hardware and delivers a fabric whose normal behavior feels *infallible*. Classic Ethernet remains best-effort—unreliable and fallible—and relies on upper layers for recovery. Bridging that gap means absorbing more of the ABP playbook at the link: credits, link-level retransmission, and tight ACK/NACK loops that shrink recovery from milliseconds to microseconds.

0.11 Forward and Backpropagation Through the Lens of the Two-State Vector Formalism

The *Two-State Vector Formalism* (TSVF), developed by Aharonov and collaborators, describes a quantum system using both a forward-evolving state vector from the past and a backward-evolving vector from the future, forming a complete description of the system between two time boundaries.

This formalism maps intriguingly well onto the two-phase behavior of supervised learning:

- **Forward propagation:** evolving the input forward through the network to predict an output.
- **Backpropagation:** retroactively applying a loss function at the output and propagating error information backward through the network to adjust weights.

Forward propagation plays the role of the forward-evolving quantum state $|\psi(t)\rangle$, and backpropagation corresponds to the backward-evolving dual state $\langle\phi(t)|$. Together, they constrain the learning dynamics at each layer via a two-state viewpoint.

0.12 Forward Propagation as Forward Evolution

In TSVF:

$$|\psi(t)\rangle = U(t, t_0)|\psi(t_0)\rangle,$$

where U is the unitary time evolution operator from an initial preparation at t_0.

In machine learning:

$$a^{(l+1)} = f^{(l)}(W^{(l)}a^{(l)} + b^{(l)}),$$

where the activations $a^{(l)}$ propagate the input forward.

0.13 Backpropagation as Backward Evolution

In TSVF, a post-selected state $\langle\phi(t_1)|$ evolves backward:

$$\langle\phi(t)| = \langle\phi(t_1)|U(t_1, t),$$

complementing the forward-evolving $|\psi(t)\rangle$.

In neural nets:

$$\delta^{(l)} = (W^{(l+1)})^\top \delta^{(l+1)} \circ f'^{(l)}(z^{(l)}),$$

where $\delta^{(l)}$ encodes error at layer l and propagates backward to adjust weights.

0.14 Two-State Update Rule

In TSVF, expectation values take the form:

$$\langle A \rangle_w = \frac{\langle \phi | A | \psi \rangle}{\langle \phi | \psi \rangle},$$

and represent weak values or amplitudes constrained by both past and future states.

In machine learning, the update rule:

$$\Delta W^{(l)} \propto \delta^{(l)} (a^{(l-1)})^\top$$

depends on the forward activations $a^{(l-1)}$ and backward error signal $\delta^{(l)}$. Together, they act like a sandwich operator:

$$\text{Update}^{(l)} \sim \langle \phi^{(l)} | \text{operator} | \psi^{(l)} \rangle.$$

0.15 Time-Symmetric Learning View

Rather than treating backpropagation as a mere computational trick, TSVF offers a time-symmetric interpretation:

- Both the input and the desired output state determine the intermediate learning dynamics.
- Each layer mediates between past input and future supervision, forming a time-bridging node.

0.16 Comparison Table

Concept	Neural Network	TSVF QM
Initial input	x	$\|\psi(t_0)\rangle$
Prediction process	Forward prop	$U(t, t_0)$ evolution
Target supervision	Loss function	Post-selection $\langle \phi(t_1) \|$
Error signal	$\delta^{(l)}$	$\langle \phi(t) \|$
Intermediate activity	$a^{(l)}$	$\|\psi(t)\rangle$
Weight update	$\delta^{(l)} (a^{(l-1)})^\top$	$\langle \phi \| A \| \psi \rangle$

Figure 3: Analogies between supervised learning and TSVF.

0.17 FITO Thinking vs. Time Symmetry

Most machine learning frameworks assume *Forward-In-Time-Only (FITO)* causality: input causes output, and learning proceeds only by adjusting from past to future. TSVF suggests a richer model:

- Supervision from the future constrains the learning of the past.
- This bidirectional model aligns with concepts from goal-driven behavior and active inference.

0.18 Conclusion

The TSVF reframing of forward and backpropagation illuminates the deeper time-symmetric structure underlying learning. Far from being just a computational trick, backpropagation can be seen as a physical dual to forward propagation—both necessary to fully specify a learning system between two boundary conditions.

This is highly relevant to our model of the protocol, where we use reversible state machines to specify forward propagation, with whatever reversible glitches might be needed to handle failures are implemented as reversible steps, and the backpropagation as the credit-based flow control mechanism.

Because they are completely symmetric, packets being sent and packets being unsent are fully managed by the flow control system.

0.19 IP and Patent Implications

The core concept of credit-based flow control is now public domain, but a handful of post-2005 implementation patents remain enforceable through at least 2036. Modern designs should audit those families but can build freely on the expired foundational work.

0.20 Practical Take-Aways

1. **Freedom to Operate.** A straightforward "one credit = one buffer" design can rely on the expired Intel/Compaq and Brocade patents for prior-art cover. Avoid features identical to the still-active patents or license them.
2. **Design Around.** Active claims tend to be narrow. You can sidestep the Mellanox "macro credit" idea by limiting link span or by using rate-based pacing instead of credit aggregation.