TD-Link — Speaking Script (15 min)

Condensed from full notes — 1532 words, ~14 min reading

Slide 1

Good morning, everyone. Today I will present TD-Link — a deterministic optical daisy-chain interconnect for large-scale particle physics detector systems. The challenge we address is how to synchronise thousands of front-end channels across a large experiment to within tens of picoseconds, reliably, and at a cost that makes large-scale deployment practical.

Slide 2

This is the FERS-5200 readout system that TD-Link was designed to serve. FERS is a family of compact, low-cost acquisition boards for large detector arrays — silicon photomultipliers, scintillators, and similar detectors. When you deploy hundreds of these boards distributed over tens of metres, they must all share a common time reference that does not degrade as the system grows.

Slide 3

This slide shows the internal block diagram of a FERS board. The relevant point for our discussion is the TDlink interface at 3.125 gigabits per second, highlighted in green.

Slide 4

TD-Link is the optical link technology that connects these boards. It carries both data and timing on a single duplex optical fibre. Multiple concentrators can be chained together — synchronised either through TD-Link itself or via GPS reference — to scale up to thousands of channels.

Slide 5

Here are the key parameters. TD-Link operates at 3.125 gigabits per second, chosen for compatibility with low-cost FPGA transceivers. It uses a ring topology: the concentrator transmits to the first board, which forwards to the next, and so on, closing back to the concentrator. Up to 16 boards per chain, 8 chains per concentrator — 128 boards total. The physical medium is standard SFP optical transceivers and multimode fibre. The benefit is a dramatic reduction in cabling complexity.

Slide 6

This slide summarises the timing architecture, which has three layers. First: clock generation at the concentrator. The master derives a 156.25 megahertz system clock from a low-jitter PLL locked to a 10 megahertz TCXO. Second: clock recovery at each slave. Each board recovers the clock from the incoming serial data, cleans it with an external jitter-cleaner PLL, and uses it as the transmit clock for the next hop. Third: global time synchronisation. The concentrator measures the round-trip delay to each board and applies a per-board correction. All boards reach a common time zero with a precision of around 25 picoseconds RMS.

Slide 7

Data transport uses a token-based train protocol: the concentrator sends an empty train downstream, each board appends its data as the train passes through, and CRC is recalculated at every hop.

Slide 8

Inside each FERS node, the recovered clock drives all the local logic — TDC, ADC, and the output serialiser. The elastic buffers are disabled to enforce a fixed, deterministic latency through each node.

Slide 9

This diagram shows the recovery chain in more detail: the transceiver CDR extracts the clock from the data stream, the jitter cleaner PLL produces a clean copy, and this clean clock drives both the digitisers and the next-hop serialiser.

Slide 10

T0 is the reference signal that defines the time origin for all detector channels. The concentrator broadcasts a T0 command down the chain. Because the link latency to each board is known, the concentrator compensates for the propagation delay and ensures that all boards execute the command at exactly the same absolute time.

Slide 11

We now come to the core synchronisation challenge. Each FPGA transceiver uses its own QPLL — a dedicated PLL for the serial bit clock. At power-up, each QPLL locks to the reference with an arbitrary output phase. Two QPLLs driven by the same reference will lock with different, unpredictable phase offsets. Links from different FPGA quads, or from different concentrators, therefore have an unknown phase relationship. We solve this by inserting a programmable clock conditioner — the AD9545 — to apply a zero-delay phase correction.

Slide 12

This slide illustrates the topology: a master concentrator distributes the reference to one or more slaves, and each slave then propagates the synchronised clock down its own FERS ring.

Slide 13

To measure the phase offset, we use the DDMTD — Dual Digital Mixer Time Difference — adopted from the White Rabbit project at CERN. Instead of comparing two clocks at their native frequency, which requires very fast analog circuitry, we mix both clocks with a third frequency that is very close but slightly offset by delta-f. The result is two low-frequency beat signals whose phase difference is proportional to

the original phase difference, but magnified by the ratio of the input frequency to delta-f. Our implementation achieves about 0.6 picoseconds of resolution, using entirely digital logic in the FPGA.

Slide 14

This diagram shows the closed loop. The DDMTD phase detector measures the offset between the recovered clock and the external reference; a PI controller drives the phase of the VCXO PLL output. The training goal is to drive this phase difference to zero, locking the distributed clock to the external 10 megahertz reference.

Slide 15

We now examine why deterministic latency matters. Each link must add a known, fixed latency. If the FIFO fill level is random, the round-trip measurement changes at every reset, and the timestamp compensation breaks. The TX elastic buffer is a FIFO between two clock domains. Its default mode adds variable latency. Bypassing it removes the variation but breaks the clock-domain crossing safety, and creates a dual-purpose conflict for the phase interpolator.

Slide 16

In buffer bypass mode, the FIFO is removed and the PCS logic runs directly from the user clock. The TX phase interpolator must then shift the user clock until the two domains meet the setup-and-hold window of the serialiser flip-flops.

Slide 17

The phase interpolator shifts the serialiser clock in fine steps of around 18.6 picoseconds. Without alignment, every lane has a different random phase after reset. The PI is used to force all lanes to the same phase, eliminating lane-to-lane skew.

Slide 18

Here is the design tension. The phase interpolator has two jobs. Job one: maintain the setup-and-hold margin between the user clock and the serialiser clock. Job two: align the serialiser clocks across lanes so that all receivers see a phase-aligned 156 megahertz. Because each QPLL starts with a random phase, using the PI to satisfy job two breaks the timing for job one. The interpolator cannot do both at the same time.

Slide 19

This slide presents our solution: the deterministic TX buffer. The idea is to re-enable the elastic buffer and use it as a phase-measurement device. The fill level of the FIFO depends directly on the phase difference between the write and read clocks. The status flag txbufstatus zero — signalling whether the buffer is at least half-full — acts as a one-bit phase detector. This flag drives the TX phase interpolator: the PI shifts the serialiser clock step by step until the FIFO stabilises at its midpoint. The elastic buffer thereby handles both the clock-domain crossing and the phase measurement, at no additional hardware cost.

Slide 20

The alignment is implemented as a small finite state machine. In phase one, the FSM establishes a known starting point and then increments the interpolator until the buffer becomes half-full. In phase two, the rising edge of the half-full flag marks the precise half-fill condition, and the FSM freezes the interpolator at that code. The PI is then fixed — it adds no further jitter — with a residual inter-lane skew of only a few picoseconds.

Slide 21

We now move to the measurement results. Timing jitter between two FERS boards is measured using the on-board TDC: both boards timestamp a common pulse, and the difference gives the RMS jitter between them.

Slide 22

This is the cable reference: both boards share the same timing clock via short copper cables. The measured jitter is our baseline — the intrinsic jitter floor of the FERS hardware itself.

Slide 23

Here is the same measurement, but with the clock distributed via TD-Link along the daisy chain. The result is that the jitter between any two boards remains approximately constant, regardless of how many boards separate them.

Slide 24

This measurement compares two boards on different links but within the same FPGA quad. Because the four lanes share a common PLL, their output clocks are naturally phase-coherent.

Slide 25

Going one step further: now we compare boards on different quads — links zero and seven — within the same concentrator. The two QPLLs are independent, so the deterministic buffer alignment is essential to bring the jitter down.

Slide 26

The most demanding case: two boards belonging to two different concentrators, synchronised through the external 10 megahertz reference. The measured jitter remains within a few tens of picoseconds RMS.

Slide 27

For comparison, this is the gold-standard reference for the same two-concentrator scenario, using a direct coaxial cable splitter instead of TD-Link. The cable reference gives only 5 to 10 picoseconds less jitter — a small overhead for the practical advantages of an all-digital, optical solution.

Slide 28

To conclude. This slide compares TD-Link against White Rabbit, the most established sub-nanosecond timing protocol in high-energy physics. White Rabbit achieves sub-nanosecond accuracy over multi-kilometre fibres, but it is built for a star or tree network with a central switch. TD-Link targets a daisy-chain ring of front-end boards inside a detector volume. The two systems address different deployment constraints: White Rabbit is the natural choice for a star topology; TD-Link, when the boards are chained together. We have shown that TD-Link delivers physics-grade timing — within 30 picoseconds RMS — over an all-optical, switch-free architecture. Thank you for your attention.