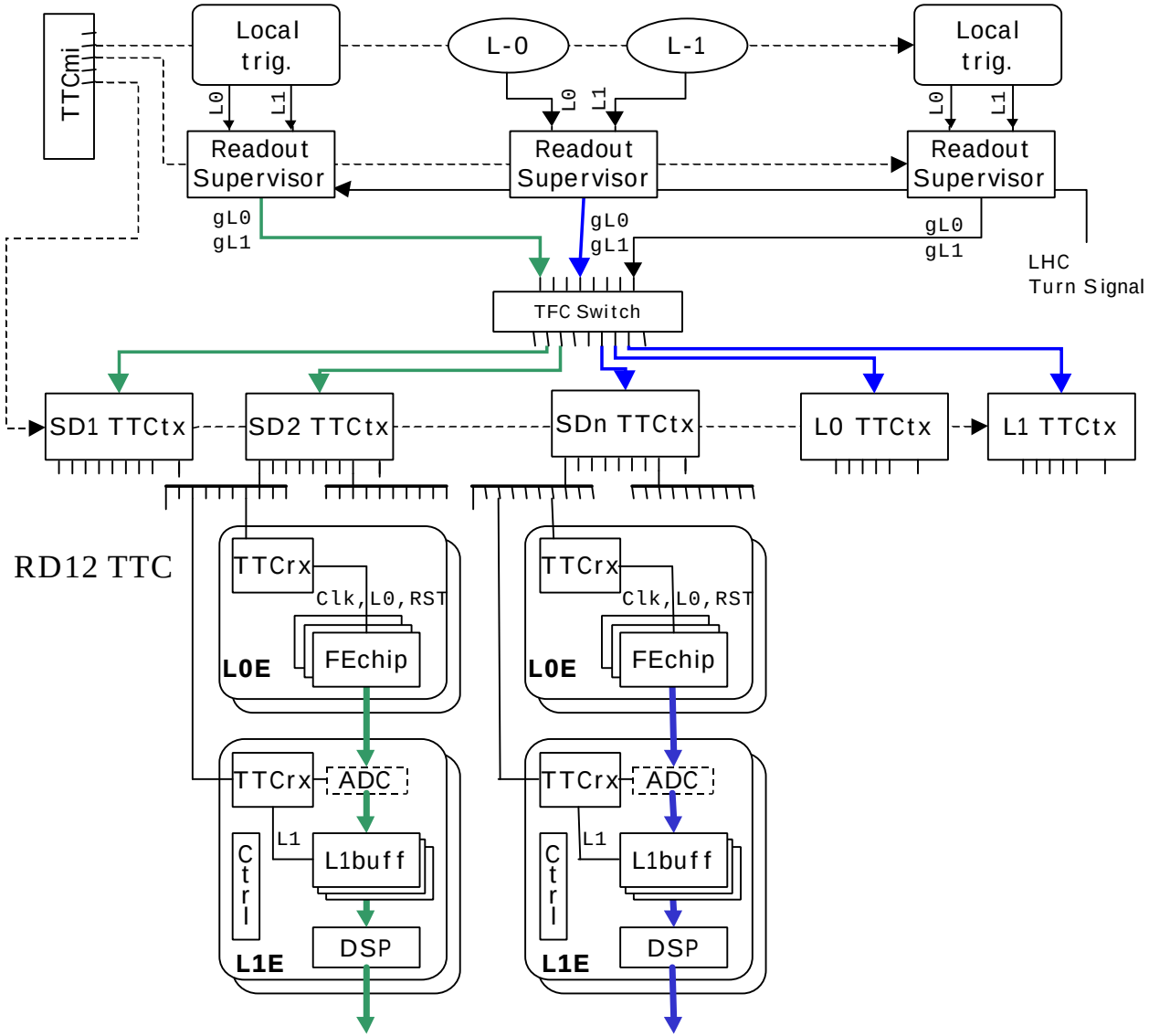
Throttling:
Infrastructure, Dead Time, Monitoring

Beat Jost
Cern EP

TFC Architecture



Problem Description (I)

- LHCb readout protocol is pure push-through, i.e. each source of data sends data without knowledge of buffer state in the destination
- if destination buffer run short, data transfers have to be stopped

  Done by disabling the trigger (Throttle)

- Buffers at various levels
  - Level-0 pipeline
  - Level-0 de-randomizing buffer
  - Level-1 trigger buffers
  - Level-1 pipeline
  - Level-1 de-randomizing buffer
  - FEM buffers, RUs, SFCs, Farm CPUs

Problem Description (II)

- Central buffer control is no problem as long as all the buffers are
  - filled/emptied synchronously (e.g. L0 de-randomizer)
  - or filled synchronously and emptied with a maximum latency (e.g. L1 de-randomizer)
    - ↪ can lead to unnecessary throttling...
- De-centralized buffer control poses problem of numbers of sources
  - ~1000 L1 electronics boards
  - ~x00 FEM modules
  - ~100 RU modules
  - ~100 SFCs

Proposal

- L0 Pipeline
  - No problem, L0 trigger has fixed latency
- L0 de-randomizers
  - monitored centrally by Readout Supervisor. Throttling L0 trigger internally
- Level-1 Buffers
  - handled by timeout in Level-1 trigger (maximum processing time)
- Level-1 de-randomizers
  - monitored locally and throttling L1 trigger via hardware signal to RS
- Level-1 Trigger buffers
  - monitored locally and throttling L0 trigger via hardware signal to RS
- FEM/RU buffers
  - monitored locally and throttling L1 trigger via hardware signal to RS
- SFC (and CPU) buffers
  - monitored locally and throttling L1 trigger via controls system (SW)

Throttling Support

- Hardware Throttles
  - RS has inputs for throttle signals for L0 and L1 trigger
  - TFC switch has two reverse paths for L0 and L1 throttles (don't forget partitioning!!)
  - to cope with the many sources of throttle signal a module performing basically a logical OR of the inputs will be needed (should be no problem)
- Software Throttles
  - The ECS interface to the RS will allow to throttle L0 or L1 triggers (prob. only throttling of L1 trigger will be used)

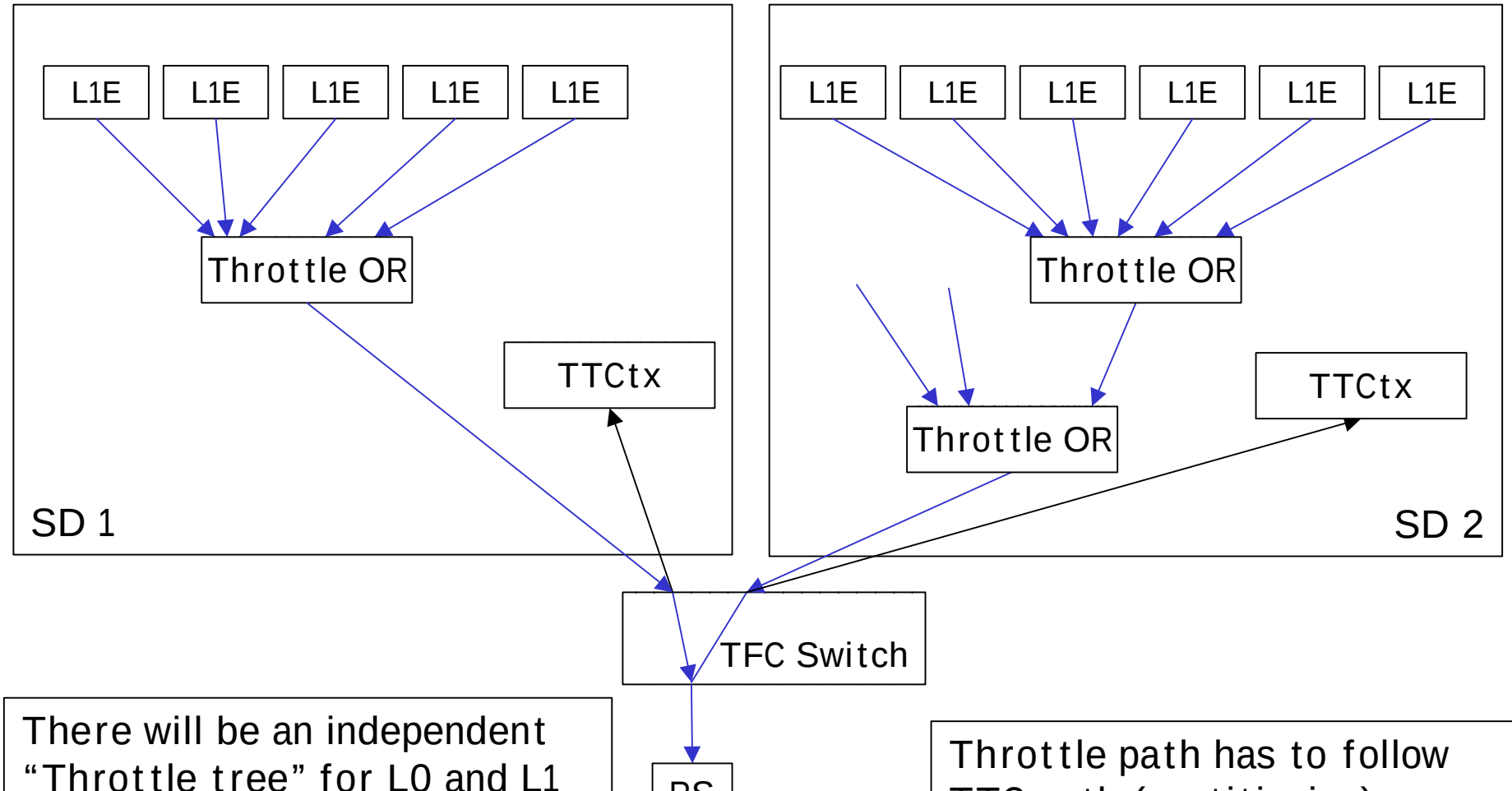
Side remark: Originally it was foreseen that all throttling would be done through the ECS system. Long and variable latency makes this difficult to implement (complicated algorithms).

Monitoring

- The RS will count the lost events (i.e. the number of events for which a positive trigger decision has been converted to a negative trigger decision) for L0 and L1 hardware and software throttles separately. In addition the total number of events lost in the two cases (L0 and L1) will be counted.
- The RS will also count the number of BXs during L0 throttling
- The RS will implement a programmable throttle timeout after which an alarm is raised to the ECS.
- The TFC switch will register the time (differentially and integrated) for which the throttle is asserted for each throttle source (history?)
- The Throttle ORs will gave the same monitoring information for each port as the TFC switch.

All this information will be available to the ECS for monitoring/alarming

Hardware Setup



There will be an independent "Throttle tree" for L0 and L1 per sub-detector (if needed)

Throttle path has to follow TTC path (partitioning)

Issues

- Throttling philosophy agreed?
- Throttling architecture agreed?
- Sufficient Monitoring?
- AOI?