Università degli Studi di Ferrara

Dottorato di Ricerca in FISICA

Ciclo XXII

Coordinatore: Prof. Filippo Frontera

Management, Optimization and Evolution of the LHCb Online Network

Candidato: Guoming Liu

Tutore: Prof. Mauro Savriè

Co-Tutore: Dr. Niko Neufeld (CERN)

Anno accademico 2007-2009

Abstract

The LHCb experiment is one of the four large particle detectors running at the Large Hadron Collider (LHC) at CERN. It is a forward single-arm spectrometer dedicated to test the Standard Model through precision measurements of Charge-Parity (CP) violation and rare decays in the b quark sector. The LHCb experiment will operate at a luminosity of $2 \times 10^{32} cm^{-2} s^{-1}$, the proton-proton bunch crossings rate will be approximately 10 MHz. To select the interesting events, a two-level trigger scheme is applied: the first level trigger (L0) and the high level trigger (HLT). The L0 trigger is implemented in custom hardware, while HLT is implemented in software runs on the CPUs of the Event Filter Farm (EFF). The L0 trigger rate is defined at about 1 MHz, and the event size for each event is about 35 kByte. It is a serious challenge to handle the resulting data rate (35 GByte/s).

The Online system is a key part of the LHCb experiment, providing all the IT services. It consists of three major components: the Data Acquisition (DAQ) system, the Timing and Fast Control (TFC) system and the Experiment Control System (ECS). To provide the services, two large dedicated networks based on Gigabit Ethernet are deployed: one for DAQ and another one for ECS, which are referred to Online network in general. A large network needs sophisticated monitoring for its successful operation. Commercial network management systems are quite expensive and difficult to integrate into the LHCb ECS. A custom network monitoring system has been implemented based on a Supervisory Control And Data Acquisition (SCADA) system called PVSS which is used by LHCb ECS. It is a homogeneous part of the LHCb ECS. In this thesis, it is demonstrated how a large scale network can be monitored and managed using tools originally made for industrial supervisory control.

The thesis is organized as the follows:

Chapter 1 gives a brief introduction to LHC and the B physics on LHC, then describes all sub-detectors and the trigger and DAQ system of LHCb from structure to performance.

Chapter 2 first introduces the LHCb Online system and the dataflow, then focuses on the Online network design and its optimization.

In ***Chapter 3***, the SCADA system PVSS is introduced briefly, then the architecture and implementation of the network monitoring system are described in detail, including the front-end processes, the data communication and the supervisory layer.

Chapter 4 first discusses the packet sampling theory and one of the packet sampling mechanisms: sFlow, then demonstrates the applications of sFlow for the network trouble-shooting, the traffic monitoring and the anomaly detection.

In ***Chapter 5***, the upgrade of LHC and LHCb is introduced, the possible architecture of DAQ is discussed, and two candidate internetworking technologies (high speed Ethernet and InfiniBand) are compared in different aspects for DAQ. Three schemes based on 10 Gigabit Ethernet are presented and studied.

Chapter 6 is a general summary of the thesis.

Introduzione

L'esperimento LHCb, è uno dei Quattro grandi rivelatori di particelle operanti sul Large Hadron Collider (LHC) del CERN. Esso consiste in uno spettrometro in avanti a braccio singolo, dedicato a misure di precisione della violazione della parità/Coniugazione di Carica (CP) finalizzate a sondare la validità del modello Standard (SM). L'esperimento LHCb lavorerà con una luminosità nominate L=$2* 10^{32} cm^{-2} s^{-1}$ e una frequenza di bunch crossing (BC) protone protone pari a 10 MHz. La selezione degli eventi è realizzata mediante un trigger a due livelli: il primo livello (L0) e quello di alto livello (HLT). Il livello L0 è realizzato mediante elettronica dedicata, mentre lo HLT è essenzialmente un trigger "software" che opera sulle CPU del complesso di computer dell'EFF ("Event Filter Farm"). La frequenza del primo livello è circa 1 MHz e la dimensione tipica dell'evento è di circa 35 Kbytes. La frequenza di dati che ne deriva è 35 Gbytes/s e costituisce una importante sfida tecnologica.

Il sistema Online è una parte chiave cruciale dell'esperimento LHCb poich fornisce tutti i servizi informatici (IT). L'infrastruttura è composta da tre componenti: "Data Acquisition" (DAQ), "Timing and Fast Control" (TFC) and "Experiment Control System" (ECS). Il sistema Online è stato sviluppato sulla base di due grandi reti dedicate basate su tecnologia Gigabit Ethernet: la "DAQ network" e la "ECS network" che compongono la "Online network". Reti cos grandi richiedono un costante e affidabile monitoraggio per il corretto funzionamento e va considerato che sistemi di gestione commerciali sono particolarmente costosi e difficili da integrare in LHCb ECS. Il sistema di monitoraggio di rete che fa parte dell'ECS di LHCb è stato implementato sulla base di un sistema SCADA (controllo di supervisione e acquisizione dati) chiamato "PVSS". Questo sistema è di tipo omogeneo. In questa tesi viene dimostrato come reti di grandi dimensioni possano essere monitorate e gesite utilizzando strumenti progettati per il controllo industriale.

La tesi è divisa nel modo seguenete:

Il capitolo 1 contiene una breve introduzione a LHC, alla fisica del quark b e comprende una descrizione di tutti i rivelatori, dei “trigger” e del sistema DAQ di LHCb dal punto di vista dell’infrastruttura e delle prestazioni.

Il capitolo 2 inizialmente introduce il sistema Online di LHCb e descrive il flusso di trasmissione delle informazioni, soffermandosi successivamente sulla progettazione e ottimizzazione della rete.

Nel capitolo 3 viene introdotto PVSS e vengono descritti in dettaglio l’implementazione del sistema di monitoraggio di rete, l’architettura e i suoi componenti: front-end, comunicazioni di dati e “supervisory layer” (strato di supervisione)

Il capitolo 4 inizialmente introduce la teoria del packet sampling e il sistema sFlow adottato. Successivamente si sofferma sull’implementazione di quest’ultimo per il troubleshooting di reti, il monitoraggio del traffico e il rilevamento delle anomalie.

Nel capitolo 5 viene introdotta la programmazione dell’aggiornamento di LHC e di LHCb. Una possibile architettura di rete per la DAQ viene successivamente presentata insieme alle due possibili soluzioni tecnologiche necessarie alla sua implementazione (Ethernet e InfiniBand). Tre proposte di soluzione basate su tecnologia 10G Ethernet vengono presentate e studiate.

Il capitolo 6 è dedicato alle conclusioni.

Acknowledgements

I am so grateful for all the support I have received whilst researching and writing up this thesis.

First, I would like to thank Dr. Niko Neufeld, my friend and supervisor at CERN. He lead me into the field of networking and network-based DAQ for high energy physics experiments, he guided and accompanied me at almost every step to this work in a friendly and encouraging way. He always covered me even when I made some stupid mistakes. My work would have been impossible without the great help from Niko. He spent a lot of time to discuss and correct my thesis even during nights and weekends.

Then I would like to express my great gratitude to Prof. Mauro Savriè, my supervisor in Ferrara University. We spent a lot of time together at my early stage of study, working on Multi-Wire Proportional Chamber (MWPC) for the LHCb Muon detector. Even though this is not directly related to my thesis, I learned from him a lot of knowledge on particle detectors and the methods to do research more importantly. He always supported me throughout my study and encouraged me to develop new interests, he gave me the opportunity to come and work on my thesis at CERN.

I am very grateful to all my colleagues in the LHCb Online Group. As the leader of the Online group, Beat Jost always shown his support and advised me with his great experiences, he also took a lot of my hard work. My friends and colleagues Loic Brarda, Gary Moine, Jean-Christophe Garnier, Enrico Bonaccorsi, Rainer Schwemmer, Juan Manuel Caicedo Carvajal and Alexander Mazurov, they taught me a lot in their areas of expertise, and we had a good collaboration. Clara Gaspar, Alba Sambade Varela and Maria Del Carmen Barandela Pazos gave me a lot of helps and support on PVSS.

I would like to thank my colleagues in Ferrara, Wander Baldini, Simone Brusa, Stefania Vecchi, Silvia Pozzi and Antonio Franconieri. They gave me a lot of helps not only to my work but also to my life in Ferrara.

A special thanks goes to Prof. Yigang Xie, Prof. Cuishan Gao and Prof. Xiaoxi Xie from Institute of High Energy Physics, China. They introduced me into the field of high energy physics and gave me many useful and constructive advices.

I would also like to thank Carles Kishimoto from CERN IT-CS department and Marc Bruyere from Force10Networks, who gave me a lot of helps on the network devices.

A warm “thank you” goes to all my dear friends who support me all the time.

Last but not least I would like to thank my family for their support and encouragement in every single day of my life.

Contents

List of Figures

List of Tables

Abbreviations

ACL	Access Control List
ALICE	A Large Ion Collider Experiment
ARP	Address Resolution Protocol
ATLAS	A Toroidal LHC ApparatuS
ATM	Asynchronous Transfer Mode
BASE	Basic Analysis and Security Engine
CAN	Controller Area Network
CERN	European Council for Nuclear Research
CLI	Command-Line Interface
CMS	Compact Muon Solenoid
COTS	Commercial Off-The-Shelf
CP	Charge Parity
CU	Control Unit
DAQ	Data Acquisition
DBM	DataBase Manager
DCS	Detector operations

DDR	Double Data Rate
DIM	Distributed Information Management
DP	Data Point
DPE	Data Point Element
DPT	Data Point Type
DU	Device Unit
ECAL	Electromagnetic Calorimeter
ECS	Experiment Control System
EFF	Event Filter Farm
EVM	Event Manager
FDB	Forwarding DataBase
FEE	Front-End Electronics
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GbE	Gigabit Ethernet
GBT	GigaBit Transceiver
GEM	Gas Electron Multiplier
GPN	General Purpose Network
HCAL	Hadron Calorimeter
HLT	High-Level Trigger
HMI	Human Machine Interface
HPC	High-Performance Computing

HPD	Hybrid Photon Detector
IBTA	InfiniBand Trade Association
ICMP	Internet Control Message Protocol
IDS	intrusion detection system
IP	Internet Protocol
IT	Inner Tracker
JCOP	Joint Controls Project
L0	the first level
L0DU	L0 Decision Unit
LAG	Link Aggregation
LAN	Local Area Network
LC	Line Card
LCG	LHC Computing Grid
LEP	Large Electron Positron
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty
LINAC	Linear Accelerator
LLDP	Link Layer Discovery Protocol
LOM	LAN-on-Motherboard
MAC	Media Access Control
MAPMT	Multianode Photomultiplier Tubes
MEP	Multi-Event Packet

MIB	Management Information Base
MWPC	Multi-Wire Proportional Chambers
NAS	Network Attached Storage
NFS	Network File System
NIC	Networks Interface Card
ODIN	Readout Supervisor
OID	Object Identifier
OT	Outer Tracker
PDU	Protocol Data Unit
PID	Particle Identification
PLC	Programmable Logic Controller
PS	PreShower
PS	Proton Synchrotron
QDR	Quad Data Rate
QSFP	Quad Small Form-factor Pluggable
RDB	Relational DataBase
RDMA	Remote Direct Memory Access
RICH	Ring Imaging Cherenkov
RMON	Remote Network Monitoring
RN	Readout Network
ROB	Readout Board
RPM	Route Processor Module

RTU	Remote Terminal Units
SCADA	Supervisory Control And Data Acquisition
SDR	Single Data Rate
SFM	Switch Fabric Module
sLHC	super LHC
SM	Standard Model
SMB	Server Message Block
SNMP	Simple Network Management Protocol
SPD	Scintillator Pad Detector
SPECS	Serial Protocol for ECS
SPS	Super Proton Synchrotron
ST	Silicon Tracker
STDIN	Standard Input
STDOUT	Standard Output
TCP	Transmission Control Protocol
TFC	Timing and Fast Control
TN	Technical Network
TOE	TCP/IP Offload Engine
TT	Tracker Turicensis
TTL	time to live
UDP	User Datagram Protocol
UDP	User Datagram Protocol

UIM	User Interface Manager
VELO	Vertex Locator
VLAN	Virtual Local Area Network
WLS	Wavelength Shifting Fibers

Chapter 1

The LHCb Experiment

1.1 The Large Hadron Collider

1.1.1 The Large Hadron Collider Machine

The Large Hadron Collider (LHC) [1] is the largest particle accelerator built at CERN, the European Council for Nuclear Research, straddling at the border between Switzerland and France at an average depth of 100 m underground. The LHC has a 26.7 km long double-ring, and it is formed by eight arcs and eight straight sections (IRs) where the experimental regions and the utility insertions are located (see Fig. 1.1). The LHC will be capable of delivering two counter rotating proton beams at the unprecedented energy of 7 TeV each hence offering collisions at a center-of-mass energy of 14 TeV. This is by far the highest achievable center-of-mass energy ever. The beams cross at a rate of 40 MHz.

The infrastructure of LHC, much of which is previously used for the Large Electron Positron (LEP), comprises the pre-acceleration system, injection system and tunnel, see Fig. 1.2. The protons are accelerated through many stages. This process starts with the Linear Accelerator (LINAC2) accelerating protons up to an energy of 50 MeV. These particles are delivered to the Proton Synchrotron Booster (PSB) which accelerates the protons up to 1.4 GeV. Then the protons are accelerated up to 26 GeV by the Proton Synchrotron (PS), and transported to the Super Proton Synchrotron (SPS). The SPS accelerates the protons to an energy of 450 GeV. The protons are finally injected in the LHC accelerator where they reach 7 TeV in 2012.

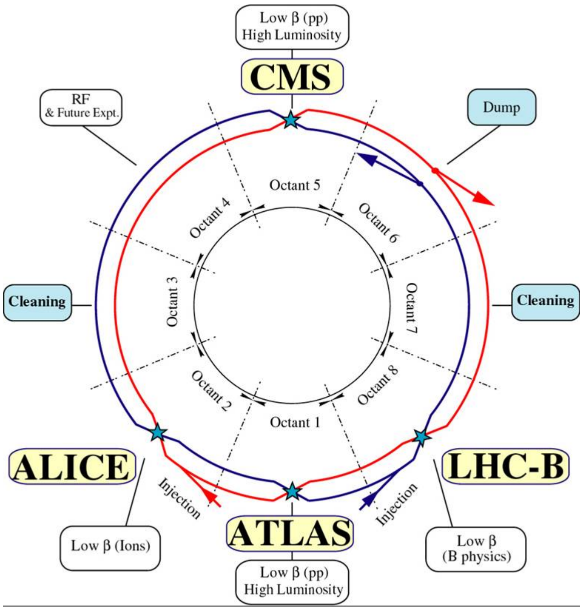


Figure 1.1: Basic layout of the Large Hadron Collider. Beam 1 circulates clockwise and Beam 2 counterclockwise.

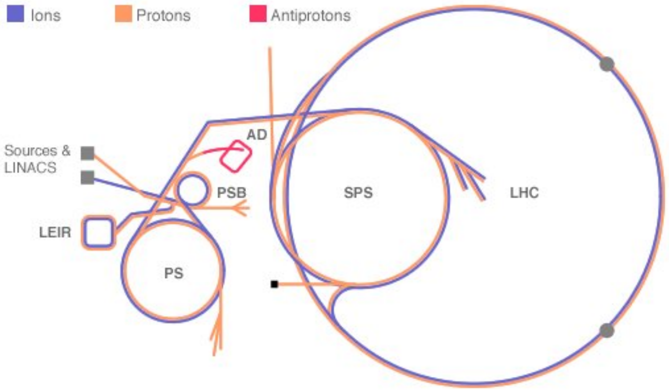


Figure 1.2: Schematic layout of the CERN accelerator complex

The LHC has two independent accelerating pipes in order to circulate protons in both directions. The pipes are mainly composed of vacuum vessel, superconducting dipoles and quadrupoles focusing magnets, radio-frequency accelerating cavities and cryogenic cooling. A huge cryogenic cooling system is required to maintain the liquid helium at the temperature of 1.9 K to keep the magnets cold. The beams travel in opposite directions in separate beam pipes - two tubes kept at ultrahigh vacuum, $\sim 10^{-4}$ Pa [2]. They are guided around the accelerator ring by a strong magnetic field, achieved using superconducting electromagnets. A total of 1232 magnetic dipoles with 8.4 T are used to bend the beams, and 392 quadrupole magnets are used to focus the beams [3]. The particles are accelerated close to light speed before collision in the radio-frequency cavities.

There are two important parameters for the particle accelerator: energy and luminosity. The center-of-mass energy, $\sqrt{s}$, of the collisions determines the energy available to produce new particles. The luminosity is a measure of the number of particles colliding per second per effective unit area of the overlapping beams. If two beams contains k bunches and n_1 and n_2 particles collide with a frequency f , then the luminosity is defined as:

$$L = \frac{n_1 \cdot n_2 \cdot k \cdot f}{4\pi\sigma} \tag{1.1}$$

where σ is the beam cross-sectional area. The event rate is measured by the colliders luminosity and is proportional to the interaction cross section. It expresses the number of particle collisions that take place every second.

The basic design parameters of the LHC machine are summarized in Table 1.1

1.1.2 The LHC Experiments

At LHC, there are four main experiments (see Fig. 1.3) run by international collaborations, bringing together scientists from institutes all over the world. Each experiment is characterized by its unique particle detector.

- ATLAS: A Toroidal LHC ApparatuS [4]
  ATLAS is one of the two general-purpose detectors at LHC. It will investigate a wide range of physics, including the search for the Higgs boson, extra dimensions and particles that could make up dark matter.

Parameter	Value
Proton beam energy	7 TeV
Number of bunches per beam	2808
Number of particles per bunch	1.15 x 10^{11}
Circulating beam current	0.584 A
RMS of bunch length	7.55 cm
Peak luminosity	1.0 x 10^{34} cm^{-2} s^{-1}
Collision time interval	24.95 ns
Number of main bends	1232
Field of main bends	8.4 T

Table 1.1: Basic design parameter values of the LHC machine

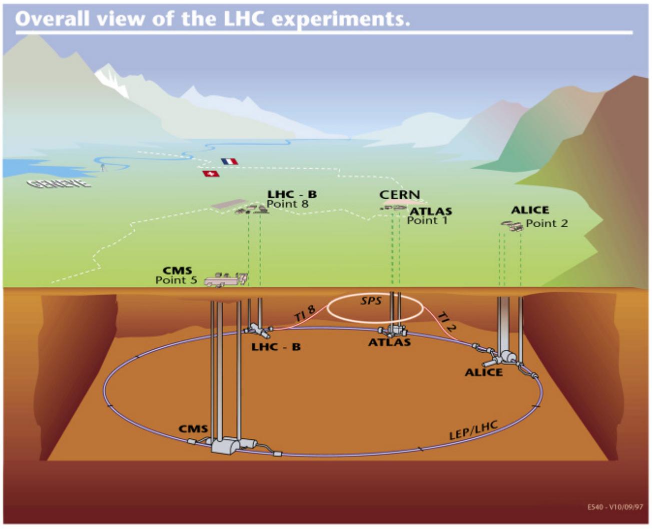


Figure 1.3: The four main experiments at LHC: ATLAS, CMS, ALICE and LHCb

- CMS: Compact Muon Solenoid [5]

  CMS is also a general-purpose detector and has the same scientific goals as the ATLAS experiment. It uses different technical solutions and design of its detector magnet system to achieve these. It is vital to have two independently designed detectors for cross-confirmation of any new discovery.

- LHCb: Large Hadron Collider beauty [6]
  LHCb is primarily dedicated to Charge-Parity (CP) violation and rare decays studies in the b sector. This experiment will test the flavour sector of the Standard Model (SM) and search for new physics by making precise measurements of B meson decays.

- ALICE: A Large Ion Collider Experiment [7]
  The main goal of this experiment is to produce, detect and study the nature of the quark-gluon plasma. Different from the other three experiments, ALICE will have Pb-Pb collisions in its interaction region. This investigation is considered fundamental to the understanding of the evolution of the early universe.

1.2 The B Physics at LHC

1.2.1 Theory Overview: CP Violation in the Standard Model

The Standard Model of particle physics is a theory which describes the strong, weak and electromagnetic fundamental forces, as well as the fundamental particles that make up all matter: it is a quantum field theory, and consistent with both quantum mechanics and special relativity. The basic building blocks of matter are six leptons and six quarks that interact by means of force-carrying particles called bosons. Every phenomenon observed in nature should be understood as the interplay of the fundamental particles and forces of the Standard Model.

The Standard Model is a very successful theory which explains, within experimental precision, all experimental phenomena yet witnessed in the laboratory and also predicts the new ones. However, it is not a definitive theory and it suffers from several limitations. First of all, it does not include gravitation, dark matter or dark energy. The essence of the mass is not resolved; whenever the mass of a particle is to be known, it has to be determined experimentally. Furthermore, the Standard Model can not explain the excess of matter over antimatter observed in the universe.

A fundamental requirement of any theory attempting to explain matter-antimatter imbalance in the early universe is that Charge-Parity (CP) symmetry could be violated. CP violation was first discovered in neutral kaon decays in 1964 [8]. Its origin is still one of the outstanding mysteries of elementary particle physics. First, CP violation in the weak interaction is generated by the complex three-by-three unitary matrix, known as the CKM matrix, introduced by Cabbibo [9], Kobayashi and Maskawa [10] Observed CP-violating phenomena in the neutral-kaon system are consistent with this mechanism. However, it cannot be excluded that physics beyond the Standard Model contributes, or even fully accounts for the observed phenomena. Furthermore, it is one of the three ingredients required to explain the excess of matter over antimatter observed in our universe. The three necessary ingredients are known as "Sakharov conditions" proposed by Andrei Sakharov in 1967 [11].

- Baryon number B violation.
- C- and CP-violation.
- Interactions out of thermal equilibrium.

The level of CP violation that can be generated by the Standard Model weak interaction is insufficient to explain the dominance of matter in the universe. This calls for new sources of CP violation beyond the Standard Model.

In the B-meson system there are many decay modes available, and the Standard Model makes precise predictions for CP violation in a number of these. The B-meson system is therefore a very attractive place to study CP violation, and to search for a hint of new physics.

1.2.2 B Meson Production at LHC

LHC will be the most copious source of B mesons, compared to the other operating or under-construction machines. In LHCb, the interaction point is displaced, only 74.3% of the bunches will collide, therefore the average bunch crossing rate will be 30 MHz. The average running luminosity of the LHCb experiment has been chosen to be $L = 2 \cdot 10^{32} cm^{-2} s^{-1}$, in order to reduce the multiple primary pp interactions rate and reduce the radiation damage of the detectors. Those with multiple *pp* interactions are much more difficult to be reconstructed than single *pp* interaction due to the increased particle density. At this luminosity, the effective bunch crossing rate is about 10 MHz, and a 100 kHz $b\bar{b}$ pair production rate is expected, hence 10^{12} $b\bar{b}$ pairs will be produced in one year.

For $b\bar{b}$ production at LHC, the angular distribution of the b and $\bar{b}$ hadrons is strongly correlated and is expected to peak in the forward directions, as is seen from the histogram in Fig. 1.4.

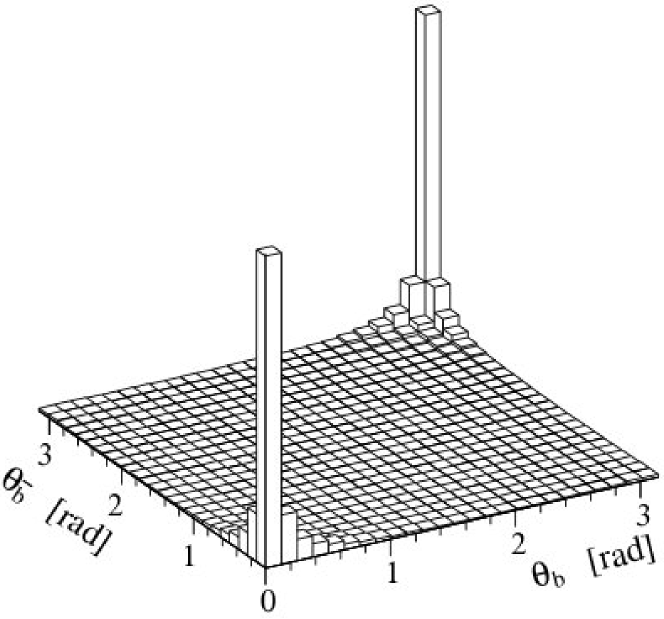


Figure 1.4: Polar angles (σ) of b and $\bar{b}$ hadrons at LHC by the PYTHIA event generator

The b events will be a small part of the total production at LHC:

$$\frac{\sigma_{b\bar{b}}}{\sigma_{inel.}} = 0.6\% \tag{1.2}$$

Therefore, in order to study b physics at LHC, a very fast and robust trigger system is required for an efficient selection of the interesting events.

1.3 The LHCb Experiment

The LHCb experiment is dedicated to the study of CP violation and other rare phenomena concerning the b quarks. Selection and reconstruction of rare B-decays in an environment with high background rates implies the following experimental requirements for LHCb:

- A fast and efficient triggering scheme to reduce the large background of non-$b\bar{b}$ events.
- To correctly reconstruct the B mesons, a precise determination of the invariant mass is necessary. Therefore the tracking system is designed for a high momentum resolution of typically subsection $\delta p/p$= 0.4%.
- To measure B_s oscillations, excellent vertexing and decay-time resolution is required. The Vertex Locator (VELO) is optimized for this task and has a proper time resolution of 40 fs.
- To identify different B meson decays with identical topology, a good particle identification (PID) is necessary.

  Efficient electron and Muon identification are required for trigger purposes and for tagging with semi-leptonic decays.

  K/π separation is exploited in kaon tagging and used to separate final states which exhibits different sources of CP-violation.

Based on the expected properties of B-meson production at LHC, and taking into account the available budget and the limited space for the detector, the LHCb detector is designed as a single arm spectrometer with an angular acceptance between 10 and 300 mrad in the horizontal plane (i.e. the bending plane of the magnet), and 250 mrad in the vertical plane (non-bending plane). The LHCb detector layout is shown in Fig. 1.5. The right-handed coordinate system adopted has the z axis along the beam, and the y axis along the vertical.

LHCb comprises a vertex detector system (including a pile-up veto counter), a tracking system (partially inside a dipole magnet), aerogel and gas RICH counters, an electromagnetic calorimeter with preshower detector, a hadron calorimeter and a Muon detector. All detector subsystems, except the vertex detector,

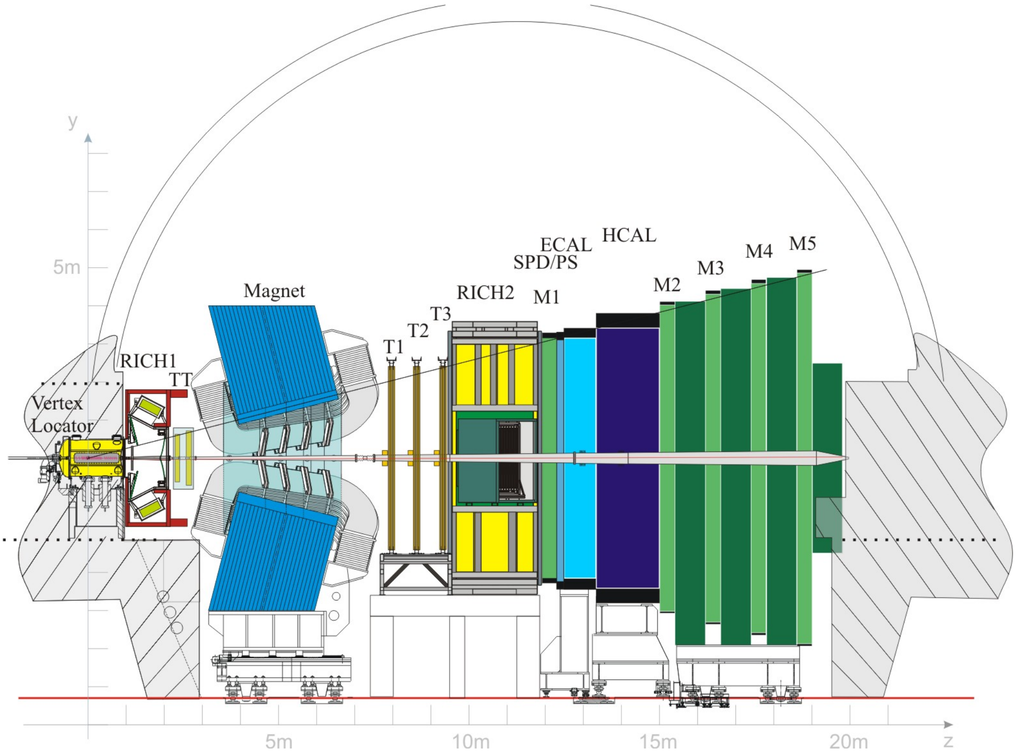


Figure 1.5: Overview of the LHCb detector

are assembled in two halves, which can be separated horizontally for assembly and maintenance, as well as to provide access to the beam pipe.

1.3.1 The Magnet

A warm dipole magnet is used in the LHCb experiment to measure the momentum of charged particles [12]. The measurement covers the forward acceptance of ±250 mrad vertically and of ±300 mrad horizontally.The magnet is of two saddle-shaped coils in a window-frame yoke with sloping poles in order to match the required detector acceptance. Tracking detectors in the magnetic field have to provide momentum measurement for charged particles with a precision of about 0.4% for momenta up to 200 GeV/c. The design of the magnet with an integrated magnetic field of 4 Tm has to accommodate the contrasting needs for a field level inside the RICHs envelope less than 2 mT and a field as high as possible in the regions between the vertex locator and the Trigger Tracker tracking station. A good field uniformity along the transverse coordinate is required by the Muon trigger.

1.3.2 The Vertex Locator

The VELO [13] detector system comprises a silicon vertex detector and a pile-up veto counter. Vertex reconstruction is a fundamental requirement for the LHCb experiment. Displaced secondary vertices are distinctive features of b-hadron decays. The measured track coordinates are used to reconstruct the production and decay vertices of hadrons and to provide an accurate measurement of their decay lifetimes. The VELO data are also a vital input to the High Level Trigger, informing the acquisition system about possible displaced vertices which are a signature of the B-mesons in the event. The High Level Trigger requires all channels to be read out within 1 μs. The pile-up veto counter is used in the first level (L0) trigger to suppress events containing multiple pp interactions in a single bunch-crossing, by counting the number of primary vertices.

The VELO is composed of a series of silicon detector stations placed along the beam direction. They are positioned at a radial distance from the beam which is smaller than the aperture required by LHC during injection and therefore must be retractable. This is achieved by mounting the detectors in a setup similar to Roman pots, see Fig. 1.6.

The layout of the VELO and the pile-up veto stations is shown in Fig. 1.7. Each station is made of two planes of sensors, measuring the radial and the angular components of all tracks. Apart from covering the full LHCb forward angular acceptance, the VELO also has a partial coverage of the backward hemisphere to improve the primary vertex measurement. To facilitate the task of the L0 trigger to select events with only one p-p interaction per bunch crossing, two additional R sensors are placed upstream of the VELO stations forming the pile-up veto stations.

The LHCb VELO silicon sensors are very complex devices. Both R and ϕ sensors are shown in Fig. 1.8. They are made of 300 μm thick n-on-n single sided

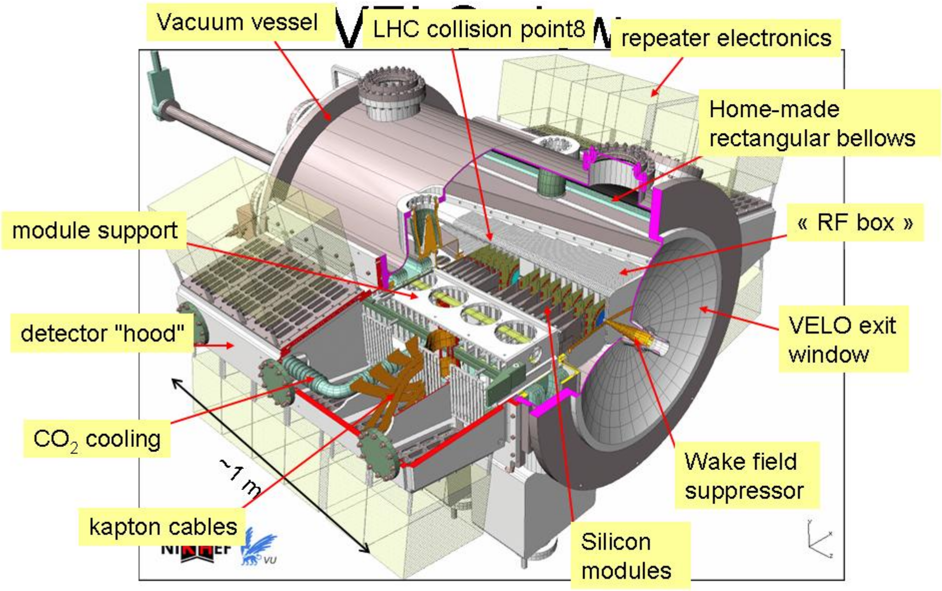


Figure 1.6: Overview of VELO

silicon wafers with AC coupling to the readout electronics covering the angle of 182°, including a 2° overlap area with the opposite sensor.

The strips in the R-sensor are concentric semi-circles segmented into four 45° sectors. Each sector has 512 strips. The radial coverage of sensor varies from the inner radius of ~8.2 mm to the outer radius of ~ 41.9 mm. The pitch between individual strips increases linearly with the radius from 40.0 μm to 101.6 μm.

The strip orientation in the ϕ sensor is semi-radial. The sensor is divided into an inner and an outer region. The inner and outer regions contains 683 and 1365 strips respectively, chosen to equalize the occupancy in the two regions. The pitch of the inner region varies from 35.5 μm to 78.3 μm and of the outer region from 39.3 μm to 96.6 μm. The sensors are flipped from station to station and the strips are tilted with a stereo angle, which is different in sign and magnitude for the inner and outer region.

The resolution on the primary vertex position is ~ 40 μm in z and ~ 10 μm in

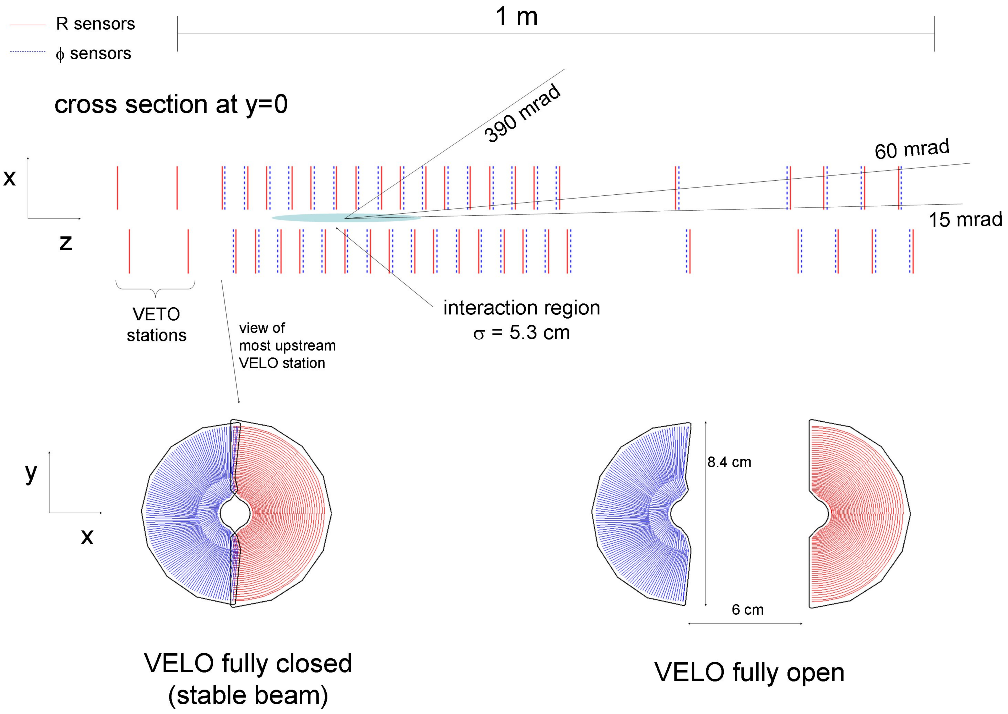


Figure 1.7: Arrangement of the VELO modules along the beam axis

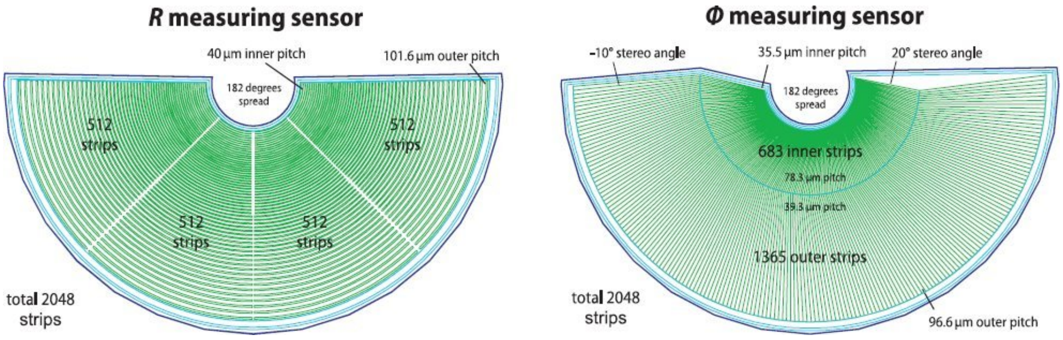


Figure 1.8: The schematic view of R and ϕ silicon sensors

x and y. For secondary vertices the spatial resolution depends on the number of tracks but on average it varies from 150 to 300 μm in z. This roughly corresponds to a resolution of 50 fs for the B-lifetime.

1.3.3 Tracking

The LHCb tracking system is used together with the VELO detector to reconstruct charged particle tracks within the detector acceptance. The LHCb tracking system consists of four planar tracking stations: the Tracker Turicensis (TT) upstream of the dipole magnet and T1-T3 downstream of the magnet. TT uses silicon microstrip detectors. In T1-T3, silicon microstrips are used in the region close to the beam pipe (Inner Tracker, IT), whereas straw-tubes are employed in the outer region of the stations (Outer Tracker, OT). The TT and the IT were developed in a common project called the Silicon Tracker (ST).

In LHCb, charged particle trajectories are reconstructed by the Vertex detector placed at the interaction point and by the Tracking stations. The simulation is shown in Fig. 1.9. The magnet provides bending power for charged particles to allow particle momentum measurements. The tracking stations provide measurements of track coordinates for momentum determination in the horizontal bending plane of the magnet and sufficient resolution for pattern recognition in the vertical coordinate.

1.3.3.1 Trigger Tracker

The Trigger Tracker (TT) [14] is located downstream of RICH1 and in front of the entrance of the LHCb magnet. It fulfills a two-fold purpose. Firstly, it will be used in the High Level trigger to assign transverse-momentum information to large-impact parameter tracks. Secondly, it will be used in the offline analysis to reconstruct the trajectories of long-lived neutral particles that decay outside of the fiducial volume of the Vertex Locator and of low-momentum particles that are bent out of the acceptance of the experiment before reaching tracking stations T1T3.

TT consists of two stations separated by a distance of 27 cm, and each station has two layers of silicon covering the full acceptance. The strips in the four layers are arranged in stereo views, x-u and v-x, corresponding to angles with the vertical y axis of 0°, 5°, +5° and 0°. The stereo views allow a reconstruction

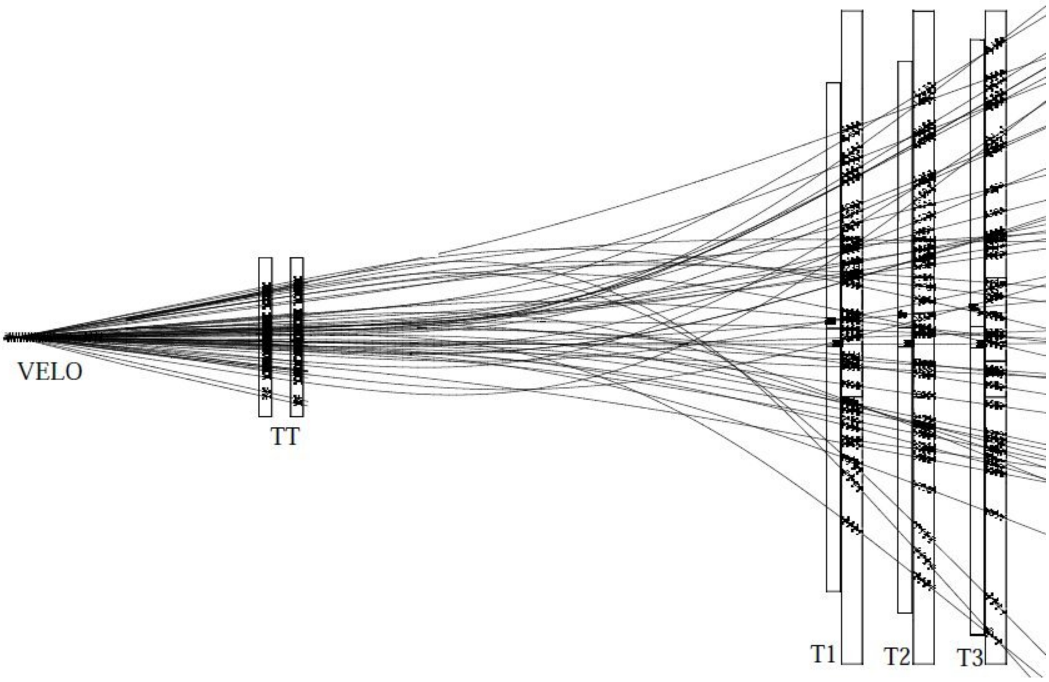


Figure 1.9: Display of an average-multiplicity event in the bending plane of the tracking system, showing the reconstructed tracks and their assigned hits

of tracks in three dimensions. The vertical orientation of the strips is chosen to obtain a better spatial resolution in the horizontal plane (bending plane of the magnet), resulting in a more accurate momentum estimate.

A layer is built out of 11 cm × 7.8 cm sensors as depicted in Fig. 1.10. The active area of the station will be covered entirely by silicon microstrip detectors with a strip pitch of 198 μm and strip lengths of up to 33 cm, the silicon sensors cover a surface of about 8.4 m^2 in total. Depending on their distance from the horizontal plane, the strips of three or four sensors are connected so that they can share a single readout. The spatial resolution is $\sim$50 μm by clustering neighboring strips.

1.3.3.2 Inner Tracker

The Inner Tracker (IT) [15] covers the innermost region of the T1, T2 and T3 stations, which receives the highest flux of charged particles. It consists of four

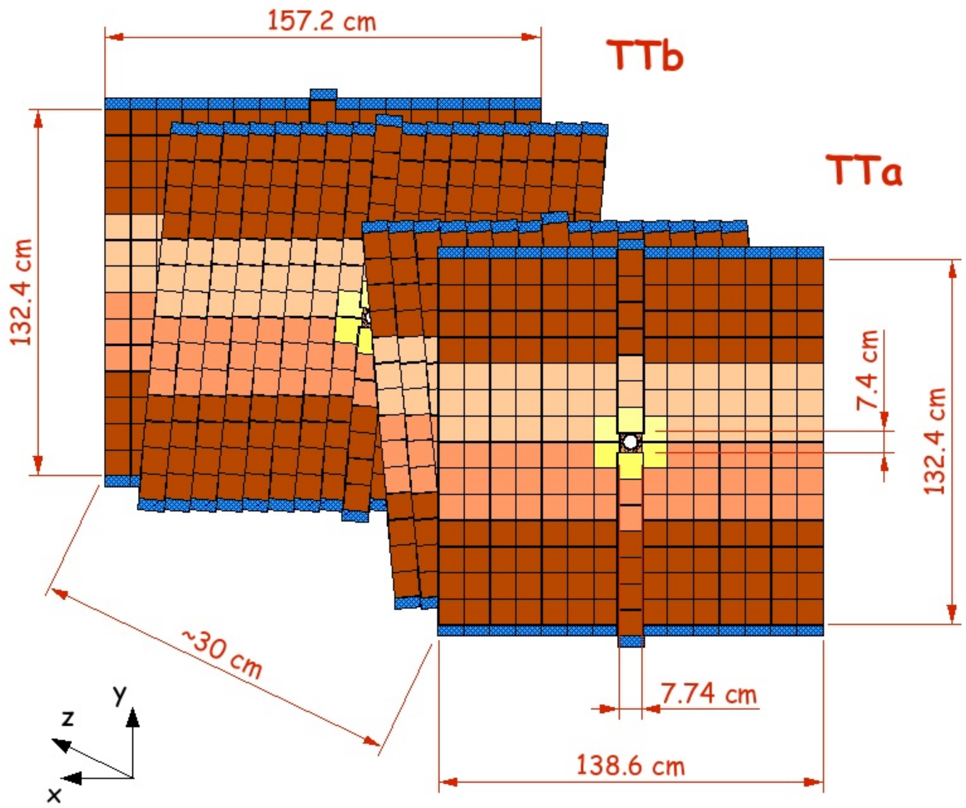


Figure 1.10: The four layers of the TT: $x - u - v - x$. The two middle layers are rotated ±5° around the z-axis to give the strips a stereo angle

cross-shaped station equipped with silicon sensors, placed around the beam. The silicon foils are 300 μm thick and have a 230 μm strip pitch, resulting in a resolution of approximately 70 μm.

An IT station consists of four boxes of silicon sensors, placed around the beam pipe in a cross-shape. It spans about 125 cm in width and 40 cm in height (see Fig. 1.11). Each station box contains four layers in an $x - u - v - x$ topology similar to that in the TT. The silicon sensors have the same dimensions as in the TT. The strip pitch is 198 μm, resulting in a resolution of approximately 50 μm.

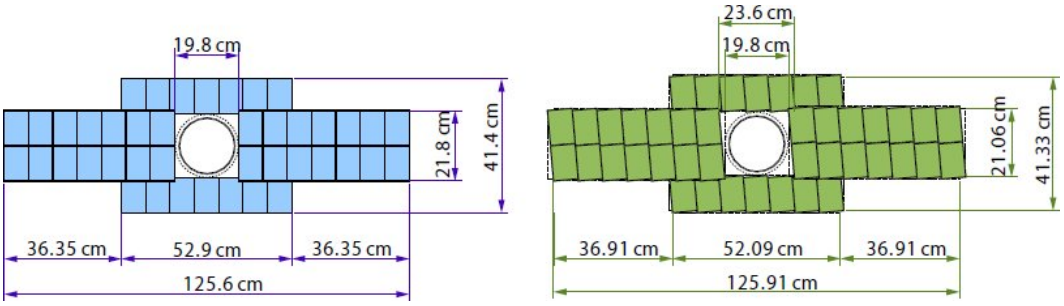


Figure 1.11: Layout of IT x and u layer with the silicon sensors in the cross shaped configuration

1.3.3.3 Outer Tracker

The Outer Track [16] is a straw tube detector which complements the IT by covering the remaining LHCb acceptance in T1, T2 and T3 station, as shown in Fig. 1.12. It is similarly laid out, with each of the three tracking stations having= four straw tube OT layers associated with it, arranged in the same manner $(x-u-v-x)$ as those of the TT and IT.

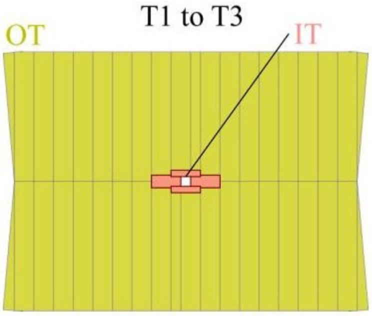


Figure 1.12: Layout of OT station (front view). In the centre the four boxes of the IT station are depicted.

The layout of the straw-tube modules is shown in Fig. 1.13. The modules are composed of two staggered layers (monolayers) of 64 drift tubes each. In the longest modules (type F) the monolayers are split longitudinally in the middle into two sections composed of individual straw tubes. Both sections are read out from the outer end. In addition to the F-type modules there exist short modules (type S) which are located above and below the beam pipe. These modules have about half the length of F-type modules, contain 128 single drift tubes, and are read out only from the outer module end. The inner diameter of the straws is 5.0 *mm*, and the pitch between two straws is 5.25 *mm*. As a counting gas, a mixture of Argon (70%) and CO_2 (30%) is chosen in order to guarantee a fast drift time (below 50 ns), and a sufficient drift-coordinate resolution (200 μm). The gas purification, mixing and distribution system foresees the possibility of circulating a counting gas mixture of up to three components in a closed loop.

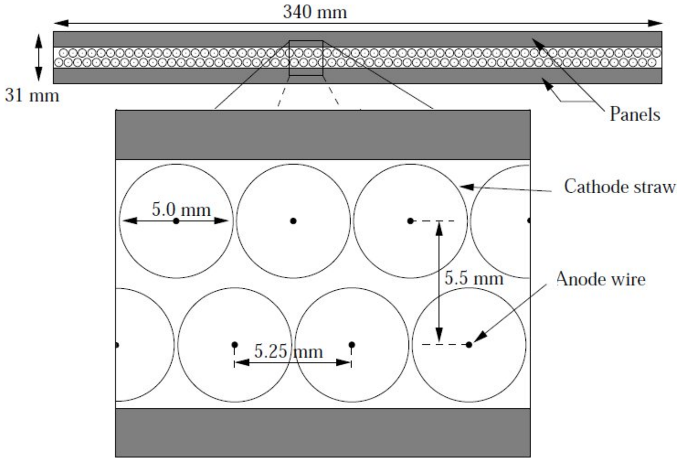


Figure 1.13: Cross section of an OT module

1.3.4 Particle Identification

Particle identification (PID) is a fundamental requirement for LHCb, and it is provided by the two Ring Imaging Cherenkov (RICH) detectors, the Calorimeter system and the Muon Detector.

For the common charged particle types (e, μ, π, K, p), electrons are primarily identified using the Calorimeter system, muons with the Muon Detector, and hadrons with the RICH system. However, the RICH detectors can also help improve the lepton identification, so the information from the various detectors is combined. Neutral electromagnetic particles (γ, π^0) are identified using the

Calorimeter system, where the $\pi^0 \rightarrow \gamma\gamma$ may be resolved as two separate photons, or as a merged cluster. Finally K_S^0 are reconstructed from their decay $K_S^0 \rightarrow \pi^+\pi^-$ [14]. These various particle identification techniques are described in the following sections.

1.3.4.1 The RICH

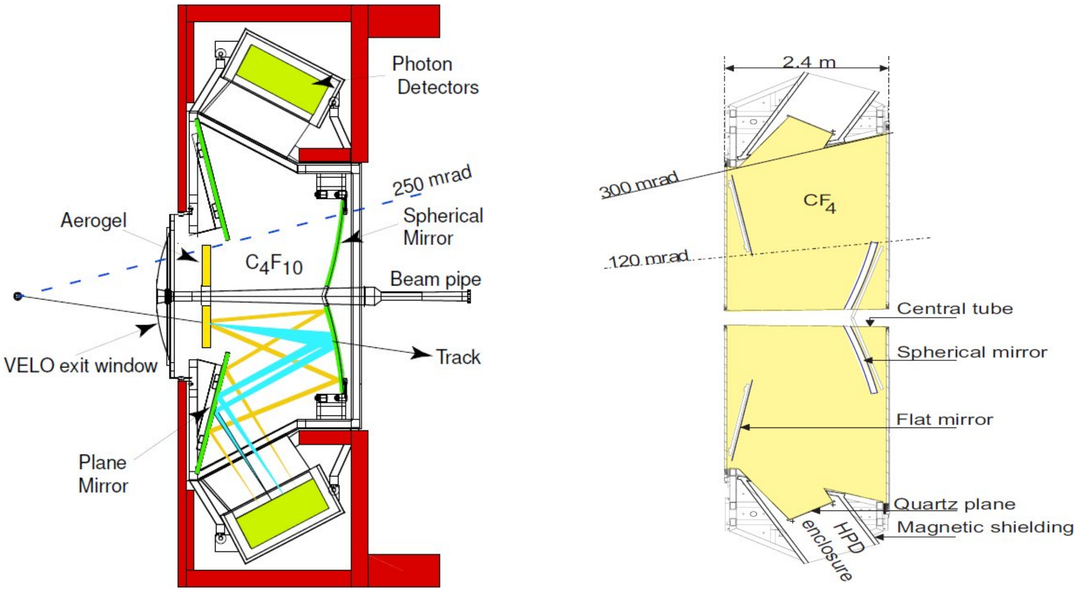
The main aim of the RICH detectors is hadron identification in LHCb, especially the separation of pions from kaons [17]. The Cherenkov radiation phenomenon is exploited in the RICH detectors in order to perform particle identification. When charged particles pass through a transparent medium at a constant speed greater than the speed of light in that medium, Cherenkov radiation is emitted at a constant angle, the Cherenkov angle θ_c, to the direction of motion of the particle. The Cherenkov angle depends only on the speed of the particle and is given by [18]

$$\cos\theta_c \simeq \frac{1}{\beta_0 n} \tag{1.3}$$

where n is the refractive index of the medium, $\beta_0 = v/c$ is the initial velocity fraction of the radiating particle.

Both detectors measure the Cherenkov angle. At large polar angles the momentum spectrum is softer while at small polar angles the momentum spectrum is harder. To cover the full momentum range, this is accomplished by two Ring Imaging Cherenkov detectors. The upstream detector, RICH1, covers the low momentum charged particle range ∼1 - 60 GeV/c using aerogel and C4F10 radiators, while the downstream detector, RICH2, covers the high momentum range from ∼15 GeV/c up to and beyond 100 GeV/c using a CF4 radiator. Overview of RICH1 and RICH2 are shown in Fig. 1.14a and Fig. 1.14b.

In both RICH detectors the focusing of the Cherenkov light is accomplished using a combination of spherical and flat mirrors to reflect the image out of the spectrometer acceptance. Hybrid Photon Detectors (HPDs) are used to detect the Cherenkov photons in the wavelength range 200 - 600 nm. The HPDs are surrounded by external iron shields and are placed in MuMetal cylinders to permit operation in magnetic fields up to 50 mT. The angular distribution of the



(a) RICH1 (b) RICH2

Figure 1.14: A cross section of the design of RICH, showing the optical path, mirrors, mirror supports

Cherenkov radiation emitted by a charged particle is related to the velocity of the particle.

A particle is identified by its rest mass and electric charge. The electric charge is determined by the bending of the particle trajectory in the magnetic field. Combining the measurement of the velocity performed by the RICH detectors with the momentum from the tracking system and the magnetic field, one is able to identify the ultra-relativistic particles. The velocity of a particle is related to the angular distribution of the Cherenkov radiation emitted by the charged particle. The resolution on the reconstructed Cherenkov angle has the following contributions: emission point, chromatic dispersion of the radiators, pixel of the photon detector and the tracking.

1.3.4.2 The Calorimeter System

The calorimeter system performs several functions. It selects transverse energy hadron, electron and photon candidates for the L0 trigger, which makes a decision 4μs after the interaction. It provides the identification of electrons, photons and hadrons as well as the measurement of their energies and positions. The reconstruction with good accuracy of π^0 and prompt photons from the primary interactions is essential for flavour tagging and for the study of B-meson decays and therefore is important for the LHCb physics program [6].

To meet the fast triggering requirements, the chosen structure for the calorimeter system consists of three elements: a Scintillator Pad Detector and single-layer Preshower (SPD/PS) detector, followed by a Shashlik electromagnetic calorimeter (ECAL) and a scintillating tile hadron calorimeter (HCAL).

- The Scintillator Pad Detector (SPD) identifies charged particles by means of 15 mm-thick scintillator tiles, which allow to separate photons from electrons. The light produced by a ionizing particle traversing the tiles is collected by Wavelength Shifting Fibers (WLS). The re-emitted green light is guided outside the detector acceptance towards the channel Multianode Photomultiplier Tubes (MAPMT) via clear plastic fibers. All the calorimeter detectors follow the same basic principle: scintillation light is transmitted to a Photomultiplier Tubes by wavelength shifting fibers. The SPD is followed by the PreShower detector that consists of 12 mm-thick lead plane placed in front of 15 mm-thick scintillator plane. The lead plates allow electrons to interact and hence produce an extra shower before reaching the scintillator plates.

- The electromagnetic calorimeter adopts the shashlik calorimeter technology [19], i.e. a sampling scintillator/lead structure readout by plastic WLS fibers. ECAL is realized as a rectangular wall constructed out of 3312 separate modules of square section. It is subdivided into three sections, Inner, Middle and Outer, comprising modules of the same size but different granularities. The performance of ECAL should comply with the following list of specifications [20].

  * an energy resolution $\sigma E/E(GeV)$ on the level of $10\%/\sqrt{E} \oplus 1\%$

  * a fast response time compatible with LHC bunch spacing 25 ns

- The LHCb hadron calorimeter (HCAL) is a sampling device made from iron and scintillating tiles, as absorber and active material respectively.

The special feature of this sampling structure is the orientation of the scintillating tiles that run parallel to the beam axis. In the lateral direction tiles are interspersed with 1 cm of iron, whereas in the longitudinal direction the length of tiles and iron spacers corresponds to the hadron interaction length λ_I in steel. The energy resolution is:

$$\frac{\sigma(E)}{E} = \frac{80\%}{\sqrt{E}} \oplus 10\% \quad (E \textit{ in GeV}) \tag{1.4}$$

1.3.4.3 The Muon Detector

Muon triggering and offline Muon identification are fundamental requirements of the LHCb experiment. Muons are present in the final states of many CP-sensitive B decays, in particular the two gold-plated decays, $B_d^0 \to J/\psi(\mu^+\mu^-)K_S^0$ and $B_d^0 \to J/\psi(\mu^+\mu^-)\phi$ [14]. They play a major role in CP asymmetry and oscillation measurements, since muons from semi-leptonic b decays provide a tag of the initial state flavor of the accompanying neutral B mesons. In addition, the study of rare B decays such as the flavour-changing neutral current decay, $B_d^0 \to \mu^+\mu^-$, may reveal new physics beyond the Standard Model [6].

Muon particles with high transverse momentum are typical signatures of a b-hadron decay, and the Muon system provides fast information for the high-P_T Muon trigger at the first level (L0) and Muon identification for the high-level trigger (HLT) and offline analysis.

The Muon system [21], shown in Fig. 1.15, is composed of five stations (M1-M5) of rectangular shape, placed along the beam axis. The full system comprises 1380 chambers and covers a total area of 435 m^2. The inner and outer angular acceptances of the Muon system are 20 (16) mrad and 306 (258) mrad in the bending (non-bending) plane respectively. Station M1 is placed in front of the calorimeters and is used to improve the P_T measurement in the trigger. Stations M2 to M5 are placed downstream the calorimeters and are interleaved with iron absorbers 80 cm thick to select penetrating muons. The minimum momentum of a Muon to cross the five stations is approximately 6 GeV/c since the total absorber thickness, including the calorimeters, is approximately 20 interaction lengths.

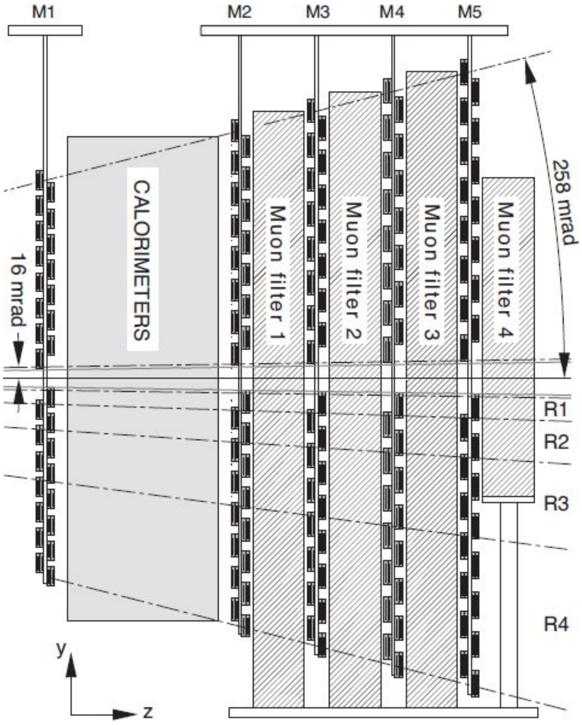


Figure 1.15: Side view of the Muon system

In the innermost region of the first Muon station, Gas Electron Multiplier (GEM) detectors have been adopted, because the innermost region is exposed to a high particle rate, which reaches the values of about 230 kHz/cm^2 at the nominal luminosity value. Multi-Wire Proportional Chambers (MWPC) have been adopted as the baseline detector in all the other regions where the expected particles rates are lower.

Further, each Muon Station is divided into four regions, R1 to R4 with increasing distance from the beam axis, as show in Fig. 1.16. The linear dimensions of the regions R1, R2, R3, R4, and their segmentations scale according to the ratio 1:2:4:8.

To provide a high-p_T Muon trigger at the L0 trigger with a 95% efficiency within a latency of 4.0 μs, each station has a time resolution sufficient to give a 99% efficiency within a 20 ns time window. The Muon system unambiguously identifies the bunch crossing which generated the detected muons, and then selects

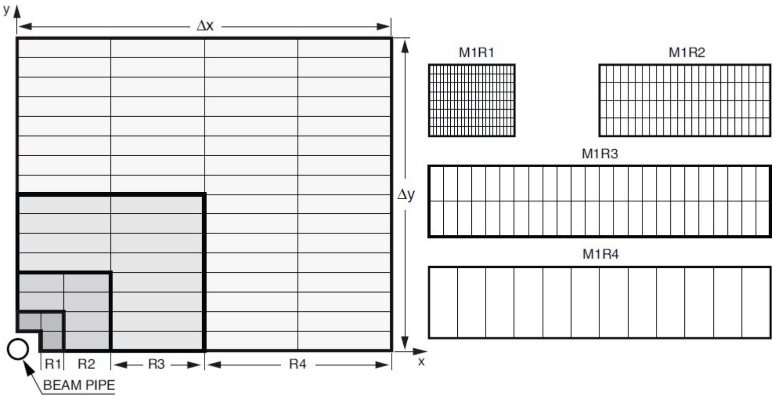


Figure 1.16: Left: front view of a quadrant of a Muon station. Right: division into logical pads of four chambers belonging to the four regions of station M1.

the Muon track and measure its p_T with a resolution of 20%.

1.3.5 The LHCb Trigger and DAQ

The LHC bunches cross at a rate of 40 MHz. LHCb will operate at a luminosity of $2 \times 10^{32} cm^{-2} s^{-1}$, about fifty times less than the LHC design luminosity. At the lower operating luminosity, the LHCb spectrometer will have only 10 MHz of crossings with "visible interactions", defined as interactions that produce at least two charged particles with sufficient hits to be reconstructed in the spectrometer [22, 23].

At the nominal luminosity of LHCb, the bunch crossings with visible pp interactions are expected to contain a rate of about 100 kHz of bb-pairs. However, only about 15% of these events will include at least one B meson with all its decay products contained in the spectrometer acceptance. Furthermore the branching ratios of interesting B meson decays used to study for instance CP violation are typically less than 10^3. The offline analysis uses event selections based on the masses of the B mesons, their lifetimes and other stringent cuts to enhance the signal over background.

Fig. 1.17 is the overview of the LHCb trigger system, showing the triggers and the expected trigger rate, and the the sub-detectors participating involved. The trigger is based on a two-level system and exploits the fact that b-flavoured hadrons are heavy and long-lived. The first level (L0), is implemented in custom hardware. Its main goal is to select high transverse energy, E_T , particles using partial detector information. The L0 trigger reduces a rate of 10 MHz of crossings with at least one visible interaction to an output rate of 1 MHz.

After a L0 accept, all the detector information is then read out and fed into the High Level Trigger (HLT). This software trigger runs in the event-filter farm composed of about 1700 CPU nodes. From the HLT, events are selected at a rate of 2 kHz and sent for mass storage and subsequent offline reconstruction and analysis.

1.3.5.1 The L0 Trigger

The L0 trigger is a custom hardware trigger system, its purpose is to reduce the LHC beam crossing rate to 1 MHz. The main goal of the L0 trigger is three-fold:

- to select high E_T particles: hadrons, electrons, photons, neutral pions and muons;
- to reject complex/busy events which are more difficult and take longer to reconstruct;
- to reject beam-halo events.

The level-0 trigger uses information from the calorimeter system, the Muon chambers and the pile-up system.

- The pile-up system provides rejection of events by identifying bunch crossings with multiple primary vertices. Two-interaction crossings are identified with an efficiency of 60% and a purity of about 95%. [23]
- The calorimeter system provides candidate hadrons, electrons, photons and neutral pions. The calorimeter system outputs to the L0 decision unit (L0DU) the highest E_T hadron, electron, photon and π^0 candidates, and the total HCAL E_T and SPD multiplicity.
- The Muon chambers allow stand-alone Muon reconstruction with a p_T resolution of 20%. The two highest p_T muon candidates from each quadrant are searched and sent to the L0DU.

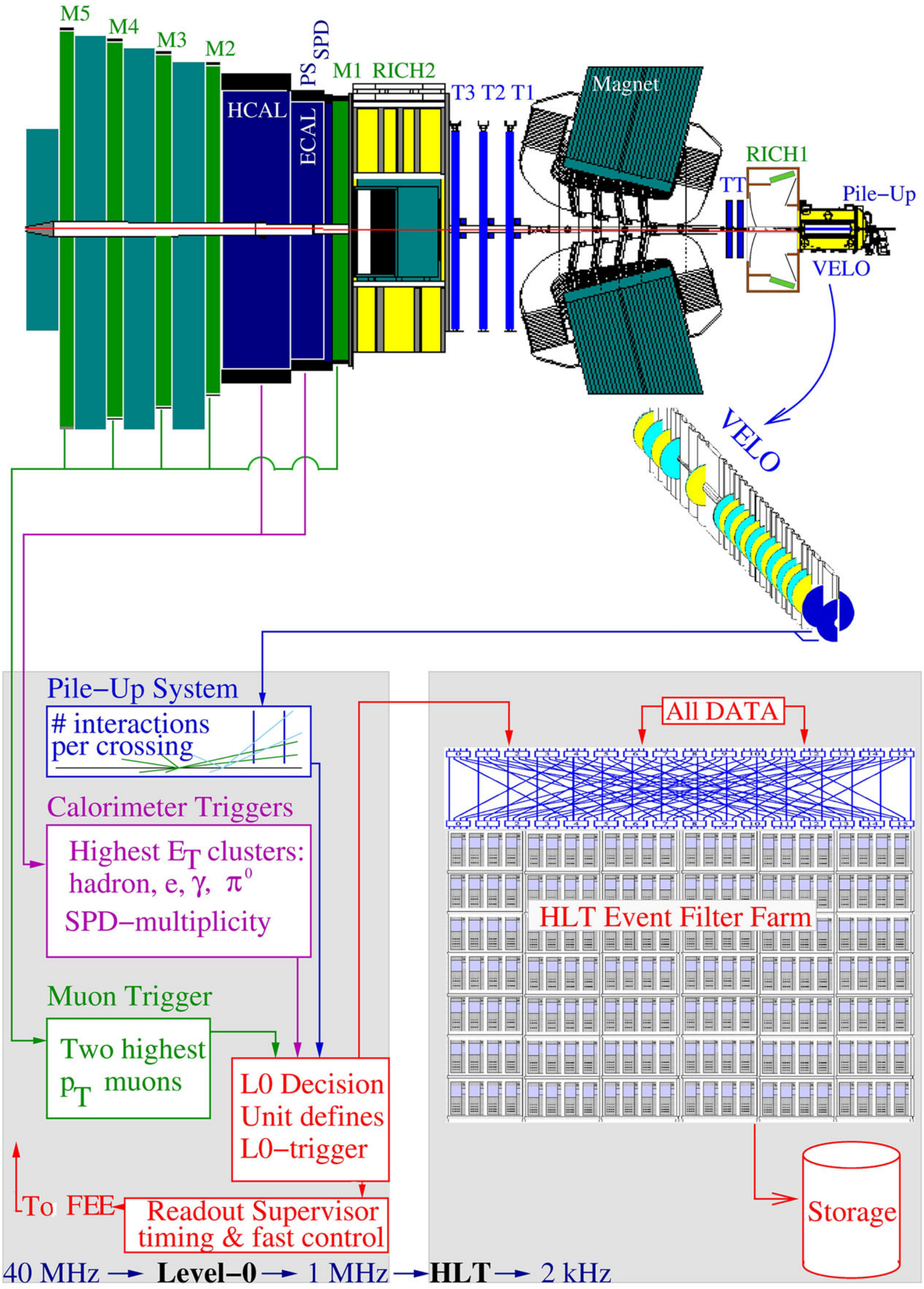


Figure 1.17: Overview of the LHCb trigger

The L0 decision unit combines the output from these three components and issues the final trigger decision. This decision is passed to the Readout Supervisor which transmits its L0 decision to the Front-End Electronics (FEE). The latency of the L0 trigger is $4\mu s$.

1.3.5.2 The High-Level Trigger

The High Level Trigger (HLT) is fully implemented in software, it consists of a C++ application which runs on every CPU of the Event Filter Farm (EFF). The EFF contains up to 2000 computing nodes. The HLT is very flexible and will evolve with the knowledge of the first real data and the physics priorities of the experiment. In addition the HLT is subject to developments and adjustments following the evolution of the event reconstruction and selection software.

The HLT is subdivided in two stages, HLT1 and HLT2. HLT1 applies a progressive, partial reconstruction seeded by the L0 candidates. Different reconstruction sequences (called alleys) with different algorithms and selection cuts are applied according to the L0 candidate type. HLT1 should reduce the rate to a sufficiently low level to allow for full pattern recognition on the remaining events, which corresponds to a rate of about 30 kHz. HLT1 starts with so-called alleys, where each alley addresses one of the trigger types of the L0 trigger. The combined output rate of events accepted by the HLT1 alleys is sufficiently low to allow an off-line track reconstruction. At this rate HLT2 performs a combination of inclusive trigger algorithms where the B decay is reconstructed only partially, and exclusive trigger algorithms which aim to fully reconstruct B hadron final states.

The HLT reduce the rate to about 2 kHz, the rate at which the data is written to storage for further analysis.

1.3.5.3 Data Acquisition

The role of the Data Acquisition (DAQ) system is to transport the event fragments, selected by L0DU, from the front-end electronics (FEE) to the HLT farm. First, the data are readout and buffered in the front-end electronics during the latencies of the hardware triggers. The selected events will be sent to the readout boards, called TELL1 or UKL1, where the front-end links are multiplexed and event fragments are assembled and zero-suppressed. Then the data are sent to the Readout Network (RN), then are transported to a node in the HLT farm. The details of DAQ system will be described in chapter 2.

1.4 Summary

The LHCb experiment is one of the four large particle detectors running at the Large Hadron Collider accelerator at CERN. It is a forward single-arm spectrometer dedicated to test the Standard Model through precision measurements of CP violation and rare decays in the b quark sector. In this chapter, all the sub-detectors and the trigger and DAQ system of LHCb were described, from structures to expected performances. More information can be found in the paper [6].

Chapter 2

The LHCb Online System and Its Networks

The Online system is a key part of the LHCb experiment. It comprises all the IT infrastructure (both hard- and software). The main task of the Online system is to operate the experiment, and transport data from the front-end electronics to permanent storage for physics analysis. This includes not only the movement of the data themselves, but also the configuration of all operational parameters, their monitoring and the control of the entire experiment. The Online system also must ensure that all detector channels are properly synchronized with the LHC clock [24, 25]. The architecture of LHCb Online system is shown in Fig. 2.1.

The LHCb Online system consists of three subsystems:

- the Data Acquisition (DAQ) system: to collect the event fragments originating from front-end electronics and transport the data belonging to a given bunch crossing, identified by the trigger, to permanent storage.

- the Timing and Fast Control (TFC) system: to synchronise the entire readout of the LHCb detector between the front-end electronics and the Online processing farm by distributing the beam-synchronous clock, the Level-0 trigger, synchronous resets and fast control commands.

- the Experiment Control System (ECS): to monitor and control the operational state of the LHCb detector. This includes the classical "slow" control,

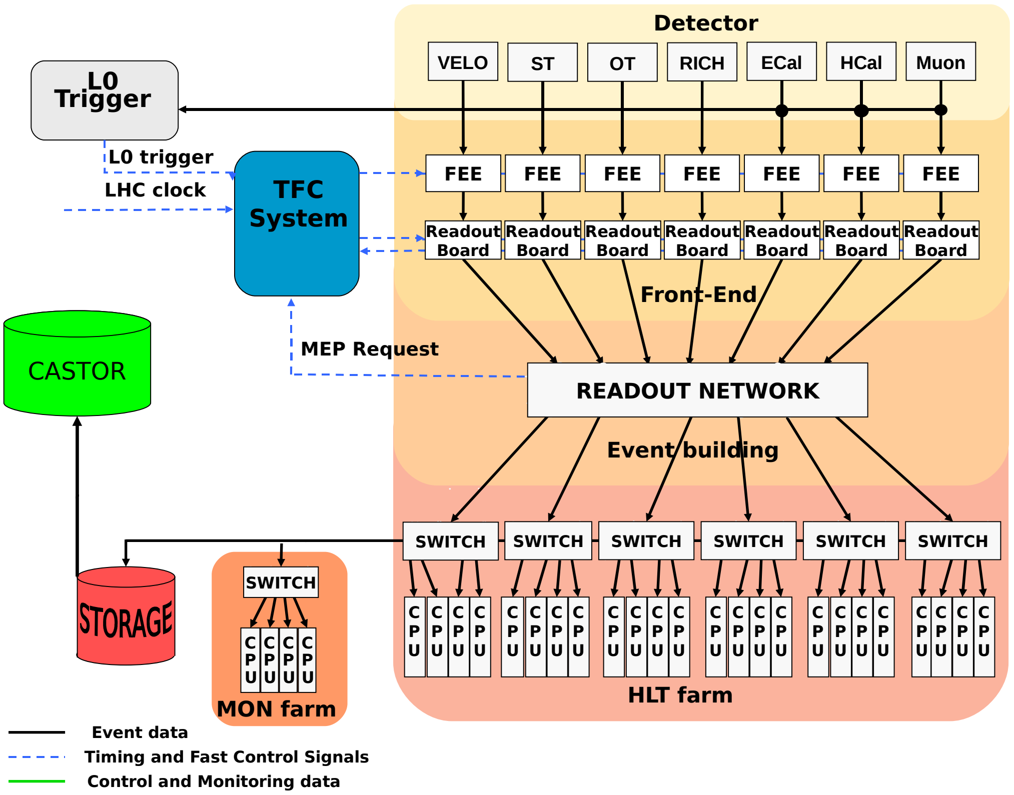


Figure 2.1: The architecture of LHCb Online system

such as high and low voltages, temperatures, gas flows and pressures, and also the control and monitoring of the Trigger, TFC and DAQ systems (traditionally called “run-control”).

2.1 Data Acquisition

2.1.1 Physics Requirements

The DAQ is responsible for the data taking from the front-end electronics to the HLT farm. Reliability and efficiency are expected from the DAQ system in order to record as many interesting events as possible.

For the events selected by the L0 trigger, the entire detector is read out, because LHCb physics requires the full event data. The data acquisition system should ensure the error-free transmission of the data from the front-end electronics to the HLT farm, and then to the storage device. This data transfer should not introduce any dead-time, if the system is operated within the design parameters.

2.1.2 Data Readout

As shown in Fig. 2.1, the data acquisition is composed of the following major components:

- Front-end Electronics (FEE) and Readout Boards
- Readout Network
- HLT CPU Farm

There are two kinds of readout-links in LHCb, the front-end electronics of VELO sends the data over analogue cables and digitizes them only at the input of the readout boards, while the other detectors digitize the signal in the front-end electronics and transmit them over optical fibres. Here the description of the front-end readout is based on the latter, but almost everything applies as well to VELO. First the front-end electronics captures the analogue signals from the detectors and converts them into digital data which are then stored in a 160 cells deep L0 buffer pipeline at 40 MHz. As had been explained in Section 1.3.5.1, the signals from the calorimeters, the MUON and the Pile-Up sub-detectors are in addition sent to the L0 trigger system. Thereafter, the L0 decision is sent to the output stage of the L0 buffer pipeline. According to the decision received, data is either rejected or written to the L0 derandomizer in order to adapt the rate of data transmission to the capacity of the front-end links. The maximum delay allowed is 4.0 μs (160 clocks).

The front-end electronics feeds the events selected by L0 trigger into 330 Readout Boards (TELL1/UKL1) [26] via approximately 5000 optical links, each link has a maximum raw band-width of 160 MB/s. The data are processed in four pre-processing FPGAs(Field Programmable Gate Array), where common-mode processing, zero-suppression or data compression is performed depending on the needs of individual detectors. The resulting data fragments are collected by a fifth FPGA (SyncLink) and formatted into a raw IP-packet that is subsequently sent to the readout network via the 4-port GbE(Gigabit Ethernet) mezzanine card. The data rate is reduced by a factor 4 to 8, depending on the attached sub-detector. A block-diagram of the TELL1 board is shown in Fig. 2.2

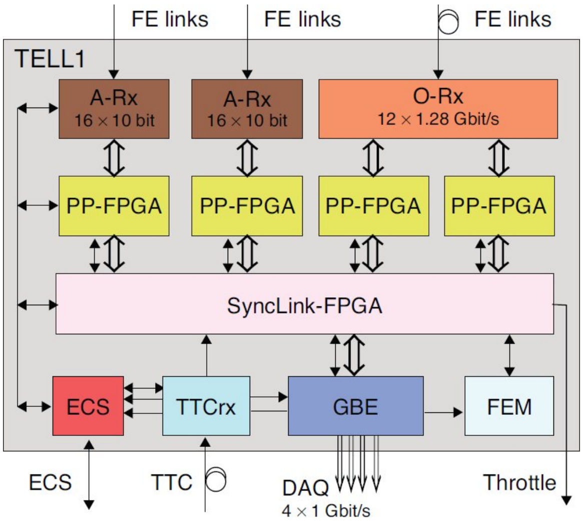


Figure 2.2: Functional block-diagram of the TELL1 readout-board. Both options for the input mezzanine cards are shown for illustration. In reality the board can only be either optical or analogue.

A simple dataflow protocol is used on the readout network. It is based push paradigm with central load-balancing and flow-control. Data is transferred to the next stage when some new data is available. The data flow is supervised by the central readout supervisor ("ODIN"). There is no synchronization or communication between components of the same horizontal level. Any data-sink (server-PC in the farm) declares its availability to the readout supervisor, which then sends trigger commands and destination broadcasts to the readout boards.

A custom network transport protocol was defined for the data transfer from the readout boards to farm-nodes, which is called Multi-Eventfragment Protocol [27]. This is a lightweight, unreliable datagram protocol over IP similar to the User Datagram Protocol (UDP). There is no retransmission mechanism.

The event size in LHCb is quite small, on average every Readout Board will have only $\sim$ 100 bytes per trigger. The Multi-Eventfragment Protocol allows to assemble the data from multiple events into a multi-event packet (MEP) in order to reduce the overhead from protocol headers and to mitigate the packet rate at the receiving nodes. In the Multi-Eventfragment Protocol, the packing factor, namely the number of events in a single MEP, can be tuned to gain a better efficiency.

For the event building, the CPU farm nodes announce their availability to receive MEPs by sending MEP Requests to ODIN when the local event queue contains less than a certain number of events. ODIN thus assigns the destinations of the MEPs dynamically according to the availability of the farm nodes. The destinations are broadcast to the TELL1 readout boards over the TTC system, along with other information, i.e. packet factor, and event ID.

All the data are forwarded in an Ethernet network, which is described in Section 2.3.2.

2.1.3 High Level Trigger and Data Storage

In the CPU farm, the HLT algorithm selects interesting interactions; upon a positive decision, the data are subsequently sent to permanent storage. The HLT is expected to reduce the overall rate from the original trigger rate of 1 MHz to $\sim$ 2 kHz by a factor of 500. As shown in Fig. 2.1, accepted events are sent off the farm nodes, routed by the edge switch (also called access switch) to the LHCb Online storage cluster, where all the selected events are written into raw data files. The size of the raw data file is normally $\sim$2 GB. The raw data files are registered in a run-database and transferred to the CERN mass-storage facility CASTOR as soon as they are closed. The storage system has a capacity of $\sim$ 40 TB, which should offer sufficient buffer space to cope with possible interruptions of the transfer to permanent storage at CERN. For security reasons, a firewall is deployed in the LHCb Online network. Only a few trusted nodes in the CERN computing center are allowed to access the LHCb Online storage cluster.

2.2 The Experiment Control System

The LHCb Experiment Control System (ECS) will handle the configuration, monitoring and operation of all experiment equipments involved in the different activities of the experiment. This encompasses not only the traditional detector control domains, such as high and low voltages, temperatures, gas flows, or pressures, but also the control and monitoring of the Trigger, TFC and DAQ systems [24, 28]. The scope of LHCb ECS is shown in Fig. 2.3

- Data acquisition and trigger (DAQ): Timing, frontend electronics, readout network, Event Filter Farm, etc.

- Detector operations (DCS): Gases, high voltages, low voltages, temperatures, etc.

- Experimental infrastructure: Magnet, cooling, ventilation, electricity distribution, detector safety, etc.

- Interaction with the outside world: LHC Accelerator, CERN safety system, CERN technical services, etc.

The ECS software is based on PVSS II [29], a commercial SCADA (Supervisory Control And Data Acquisition) system. A common project: the Joint Controls Project (JCOP) [30] has been setup between the four LHC experiments to define a common architecture and a framework to be used by the experiments in order to build their control systems. This toolkit provides components needed for building a homogeneous ECS system, and facilitates integrating the sub-systems coherently. Based on this framework, a hierarchical and distributed system was designed as depicted in Fig. 2.4.

The ECS is organized in a tree, where parent-nodes "own" child-nodes and send commands to them. Children send "states" back to their parents. Nodes are of two types:

- Device Units: which are capable of driving the equipment to which they correspond[1].

[1] Graph-theoretically speaking these are the leave-nodes

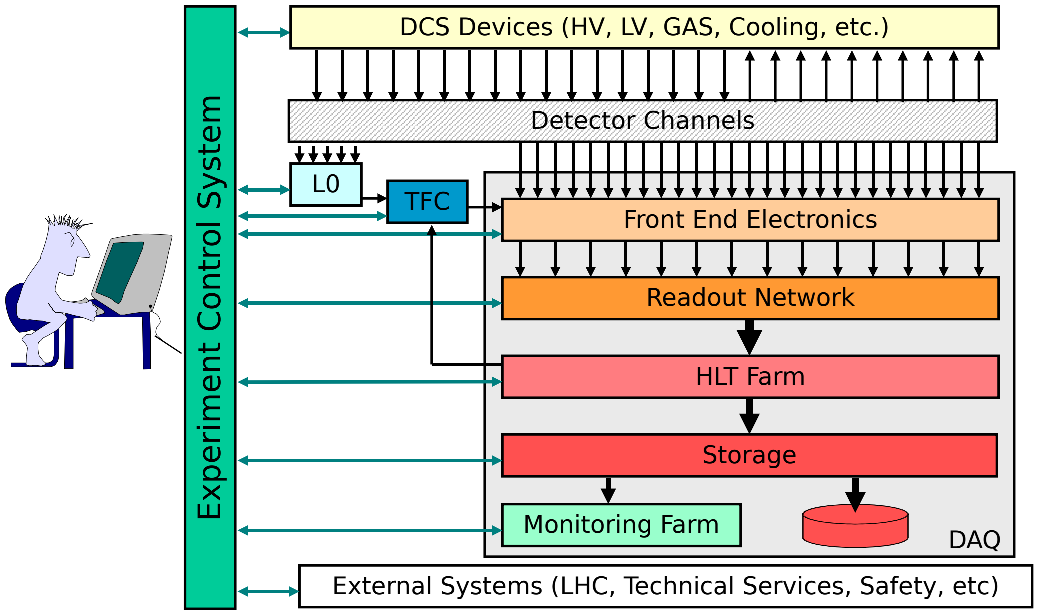

Experiment Control System
DCS Devices (HV, LV, GAS, Cooling, etc.)
Detector Channels
L0
TFC
Front End Electronics
Readout Network
HLT Farm
Storage
Monitoring Farm
DAQ
External Systems (LHC, Technical Services, Safety, etc)

Figure 2.3: Scope of the Experiment Control System

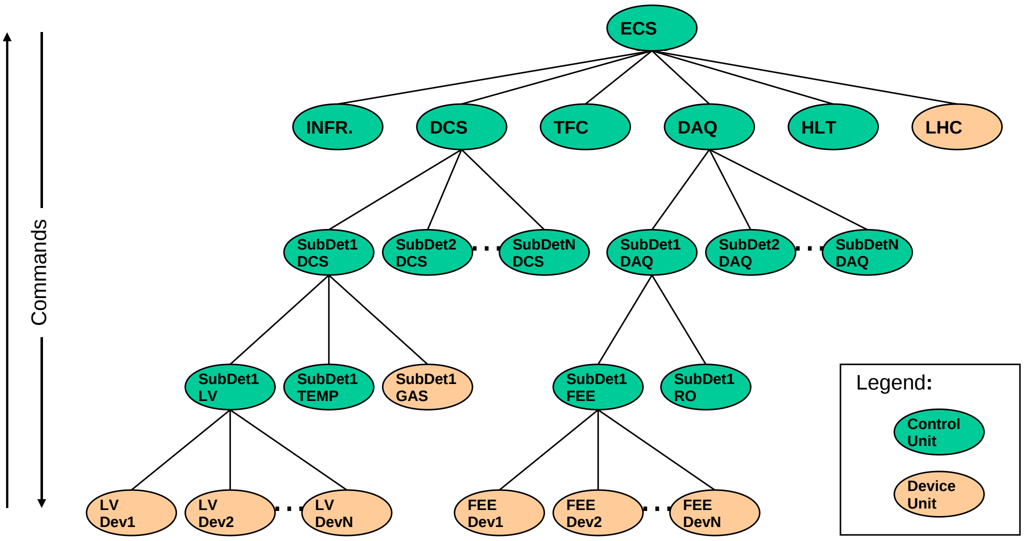


Figure 2.4: The Architecture of ECS Software

- Control Units: which can monitor and control the subtree below them, i.e., they model the behaviors and the interactions between components.

The hardware components of the ECS are diverse, mainly as a consequence of the variety of the equipment to be controlled, ranging from standard crates and power supplies to individual electronics boards. In LHCb, a large effort was made to minimize the number of different types of interfaces, connecting buses and devices. The field buses have been restricted to: SPECS (Serial Protocol for ECS), CAN (Controller Area Network), Ethernet.

The State sequencing in the ECS system is achieved by using a Finite State Machine (FSM) package, based on SMI++ [31] that allows creating complex logic needed. Device units and control units are all modeled as a Finite State Machine. FSM represents the state of each sub-system in a simple way, provides convenient mechanism to model the functionality and behavior of a component, and provides an intuitive user interface for the experiment shifts.

Physically the LHCb ECS consists of some central service servers and a large number of servers which provide the interface to the experimental equipment and supervise them. All the servers are connected to an Ethernet network. Besides, a lot of other devices, such as power supplies communicate via Ethernet as well. A large Ethernet network, called control network or ECS network, is therefore deployed specifically for the ECS, which is described in Section 2.3.1.

2.3 LHCb Online Networks

As has been explained in Sections 2.1 and 2.2, the LHCb experiment requires two dedicated networks for DAQ and ECS respectively, which are referred as *Online networks* in general. Reliable and high-performance networks are essential for the operation of LHCb. The requirements for LHCb have been described in [32]

The LHCb detector is located in an underground cavern, 100 m below the surface. The readout boards, HLT farm and most of the Ethernet devices are installed in the counting rooms underground, while the file storage, database, ECS servers and terminals are installed in a surface building. So the Online network is divided into two parts by geography, one is on the surface and the other is underground. The simplified network topology is shown in 2.5 (surface) and 2.6 (underground). These two part are connected by two 10 Gb/s fiber links as a trunk.

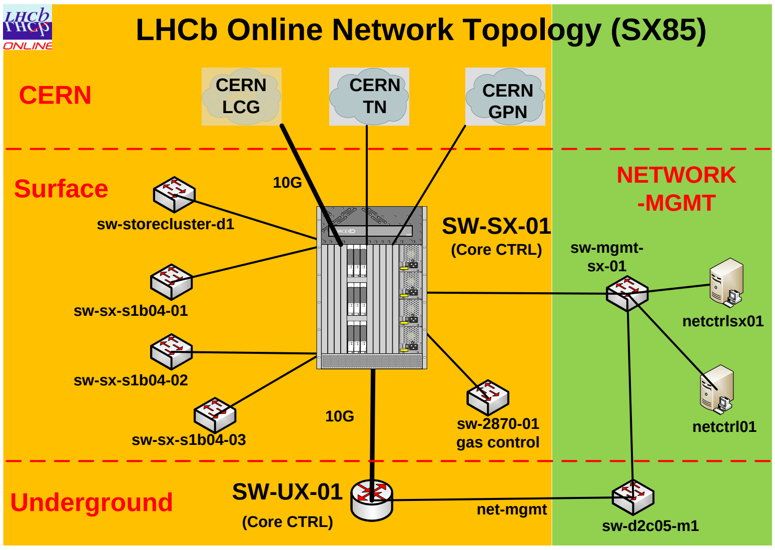
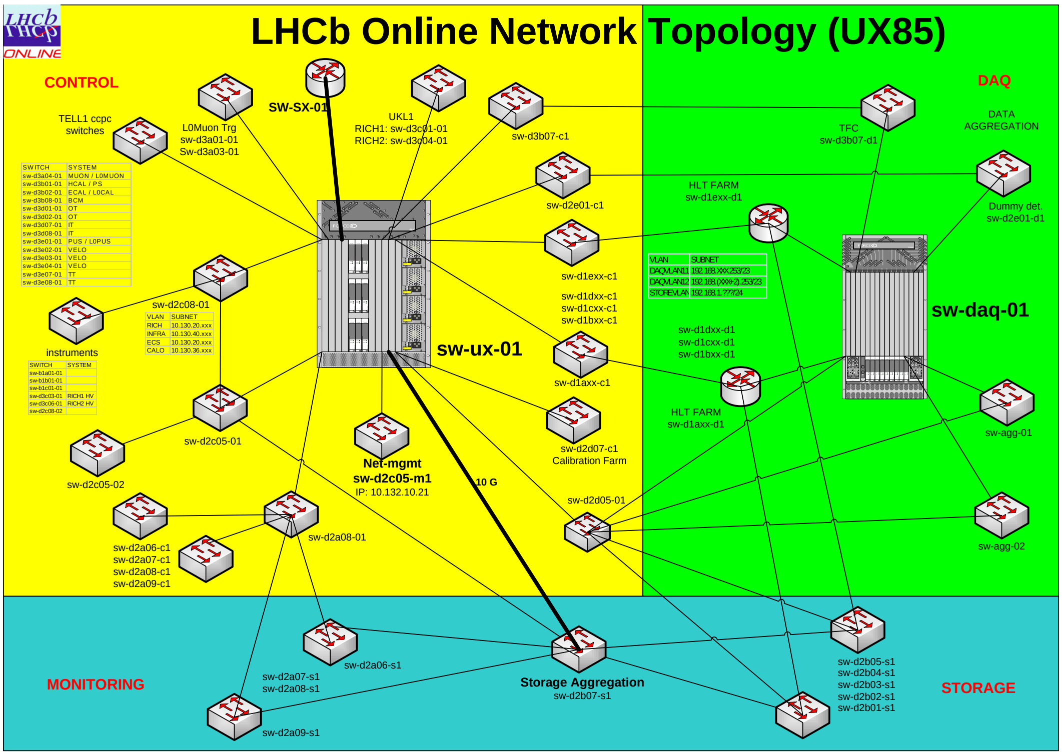


Figure 2.5: The topology of the network in the surface

In following sections, the architecture and design of the networks will be described.

2.3.1 Control Network

This network provides connectivity between any two computers inside the LHCb experiment, all the communications inside take place over the control network with the exception of the event data transfers. The ECS services and the central services like Network File System (NFS), database access, all run across this path. The control network does not have very high performance requirements, but it has to be very reliable. For management purposes, the switches of the DAQ networks are also linked to the control network as the other host nodes.



SWITCH	SYSTEM
sw-d3a04-01	MUON / L0MUON
sw-d3b01-01	HCAL / PS
sw-d3b02-01	ECAL / L0CAL
sw-d3b08-01	BCM
sw-d3d01-01	OT
sw-d3d02-01	OT
sw-d3d07-01	IT
sw-d3d08-01	IT
sw-d3e01-01	PUS / L0PUS
sw-d3e02-01	VELO
sw-d3e03-01	VELO
sw-d3e04-01	VELO
sw-d3e07-01	TT
sw-d3e08-01	TT

VLAN	SUBNET
RICH	10.130.20.xxx
INFRA	10.130.40.xxx
ECS	10.130.20.xxx
CALO	10.130.36.xxx

SWITCH	SYSTEM
sw-b1a01-01	
sw-b1b01-01	
sw-b1c01-01	
sw-d3c03-01	RICH1 HV
sw-d3c06-01	RICH2 HV
sw-d2c08-02	

VLAN	SUBNET
DAQVLAN11	192.168.XXX.253/23
DAQVLAN12	192.168.(XXX+2).253/23
STOREVLAN	192.168.1.???/24

Figure 2.6: The topology of the network underground

2.3.1.1 Architecture of the Control Network

The control network is built as a tree of switches and routers, as shown in Fig. 2.7. The core nodes of the tree are two core routers, which are located on the surface and underground respectively. The core routers are high-performance chassis-based devices E600 and E600i provided by Force10 networks [33]. They have a large set of features and provide built-in redundancy: power supplies, cooling fans, switching fabrics, routing modules, etc. Normally, the ECS devices do not need high bandwidth, so for cost reasons they are connected to fan-out switches (HP procurve 2650 [34][1]) via 100 MbE ports. These access switches are connected to the core routers via 1 GbE port. However, servers with demands of high network bandwidth or high reliability are connected directly to 1 GbE or 10 GbE ports

[1] The HP ProCurve Switch 2650 is a low-cost, stackable, multi-layer switches with 24 or 48 auto-sensing 10/100 Mb/s ports and dual-personality ports for 10/100/1000 Mb/s.

of the core routers.

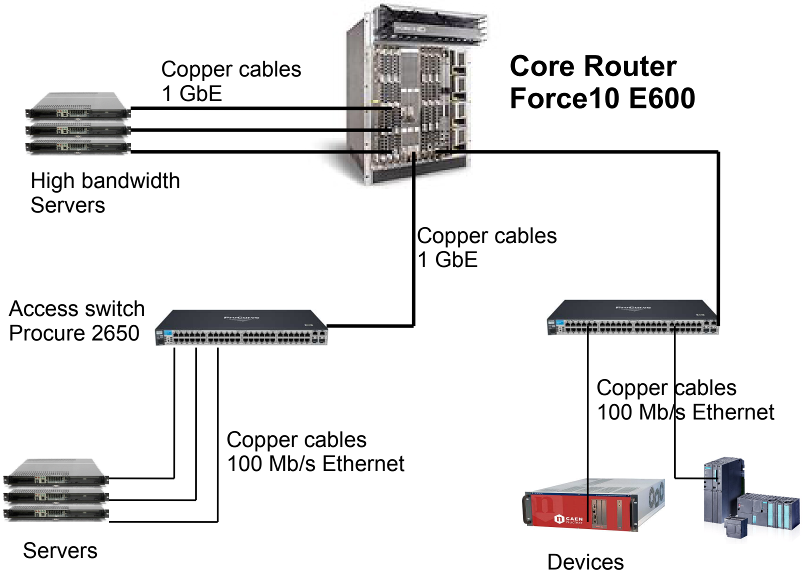


Figure 2.7: Tree Structure of the Control Network

2.3.1.2 Interfaces to CERN Network: GPN, TN and LCG

The LHCb Online network is a private local area network (LAN), which is not directly accessible from the Internet or the CERN network. Access to the LHCb Online network is only permitted via dedicated dual-homed gateways which are only accessible inside the CERN network. As shown in Fig. 2.5 and Fig. 2.6, there is only one interface connected to CERN networks individually:

- **General Purpose Network (GPN)**

  CERN GPN is the standard campus network for all general services, with access to the Internet via the CERN external firewall. The LHCb gateway traffic passes through the interface to CERN GPN, and the LHCb web services as well.

- **Technical Network (TN)**

  CERN TN is reserved for control systems, it is almost separated and not accessible from outside CERN, only authorized computers are allowed to access specific nodes in CERN TN through this interface.

- **LHC Computing Grid network (LCG)**

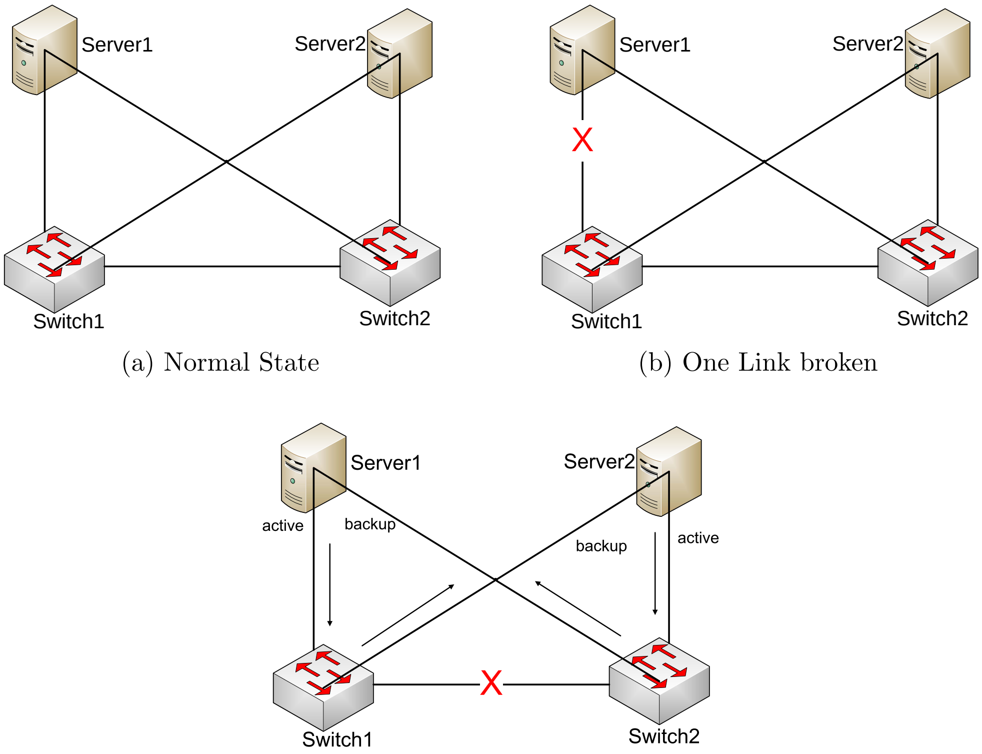
  The main purpose of this interface is to transfer physics data from the LHCb local storage to CERN CASTOR permanent storage. This interface only allows authorized servers access a set of servers in LCG network.

2.3.1.3 Redundancy for Critical Services

Redundancy is also important for LHCb, but it is not as critical as for some other services e.g. in a bank. High availability and redundancy always introduce extra cost, and most of the servers and applications are not redundant in LHCb, so the network only provide partial redundancy for those critical service like NFS and Oracle database access. The service and network configuration for NFS and Oracle database are similar, so only the network configuration for the Oracle database is described below.

There are (currently) four servers in the Oracle database cluster. The public interfaces of these server are connected to the core router directly and distributed into two different linecards, so even if one linecard fails the service will not stop. As mentioned before, the core router has high reliability and built-in redundancy. For each server, the cluster private interface used for the cluster internal communication is a link aggregation (LAG) [35] which is also known as bonding [36] in Linux. The bonding consists of two physical GbE interfaces, which are connected to the core router and a HP Procurve 3500 switch [37] individually. The bondings of the servers work in active-backup mode, only one link is active and transmits data, the other link is in standby mode. In the switches, these ports are in a Layer 2 Virtual LAN (VLAN), and these two VLANs are connected. Fig. 2.8a shows the example for two nodes.

This configuration tolerates the failure of one switch or one link, which is explained below. When one of the switches fails, all the links connected to the



(a) Normal State

(b) One Link broken

(c) Uplink broken

Figure 2.8: Oracle database cluster private network

good switch will become active, all the internal messages will pass through the good switch. If one link fails:

- the link between the servers and the switch fails as shown in Fig. 2.8b: the other link will become active and the traffic will not be interrupted in any case.
- the uplink between the switches fails as shown in Fig. 2.8c, these two VLANs in different switches are serperated:

  if all the active links are connected to the same switch, then all the traffic are transferred in this switch.

  if the active links are connected to not only one switch, the servers will use the active link to send traffic and the backup link to receive traffic from the other servers as the arrows show in Fig. 2.8c.

2.3.1.4 Management Network

The management network consists of two HP Procurve 3400 switches, one is on the surface and one is underground, they are connected by a 10 GbE fiber link. The main purpose of the management network is to manage the two core routers. The network management stations are attached to this network. Beside, the 10 GbE fiber link between surface and underground can quickly provide redundancy in case the links between the two core routers suffer serious damage.

2.3.2 DAQ Network

The DAQ network is dedicated to the data taking, which demands high bandwidth. This network is performance critical.

2.3.2.1 Constraints, requirements and choice of technology

There are several important parameters to evaluate the network performance: loss rate, latency and bandwidth and some other more specific ones.

- **Loss Rate**
  As has been explained in Section 2.1.1, the LHCb physics requires the full event data, and there is no retransmission mechanism in the DAQ protocol. If one packet gets dropped in the network, the CPU has to discard all the other received packets belonging to the same event. A zero-loss network is desired for data taking.

- **Latency**
  The latency for HLT is 4 ms, while the latency for a packet to pass through the network is about 20 μs, so the latency is not critical for the DAQ network.

- **Bandwidth**
  According to the simulation result in the DC06 exercise [38], the requirement for the data readout network is listed in Table 2.1. The total event size is $\sim$ 33 kB, the L0 trigger rate is 1 MHz, so the total input data rate is $\sim$ 36 GB/s, the total ports number is $\sim$ 1153.

Item	Value
Packing Factor	13
Network MTU	8192
Total Event Size (Bytes)	32860
Number of TELL1	308
Total Input Data Rate (MB/s)	36177
Average MEP size (Bytes)	2051
Average Link Load	43.5%
Readout Network Input Ports	653
Readout Network Input Ports	500
HLT CPU number	∼ 1700
Total Ports in used	1153

Table 2.1: Bandwidth Requirement

In addition to the technical requirements, per the DAQ network technology we has to take into account also the other aspects: industry support and cost etc. Long-term support from industry is essential in order to ensure future possible upgrades and smooth transitions to new technologies, when they become available. Costs of the initial acquisition and then of the future maintenance of the network should be kept to a minimum.

Since ∼1999, a lot of studies have been carried out in order to find the best technology for the DAQ network in LHCb [39, 24, 25, 40] and also in the other LHC experiments [41, 42]. Three candidate technologies have been studied: Asynchronous Transfer Mode (ATM), Myrinet, Gigabit Ethernet. Ethernet has a big community of users, the prices of GbE decreased continuously as GbE gained wider acceptance and more delivery. GbE had a good ratio of performance to price, and wider and longer support from industry, so it was adopted for DAQ in LHCb and the other experiments as well.

With the change of the LHCb trigger architecture, the upgrade of the DAQ network architecture was studied and published in [43].

2.3.2.2 Architecture and Implementation of the DAQ Network

The DAQ network has a similar tree structure as the control network, see Fig. 2.9. This is a multi-layer network topology. The core layer consists of a high-performance chassis switch Force10 E1200i [33], which offers non-blocking bandwidth for DAQ.

- All the data sources TELL1 are directly connected to the core router, because the average traffic rate of input links is about 43.5%, which is already quite high, and the concentration layer becomes unnecessary.
- A concentration layer is introduced to transfer the data to the HLT nodes. The traffic to each HLT node is only 20% - 30% of the line rate and depends on the number of HLT nodes.

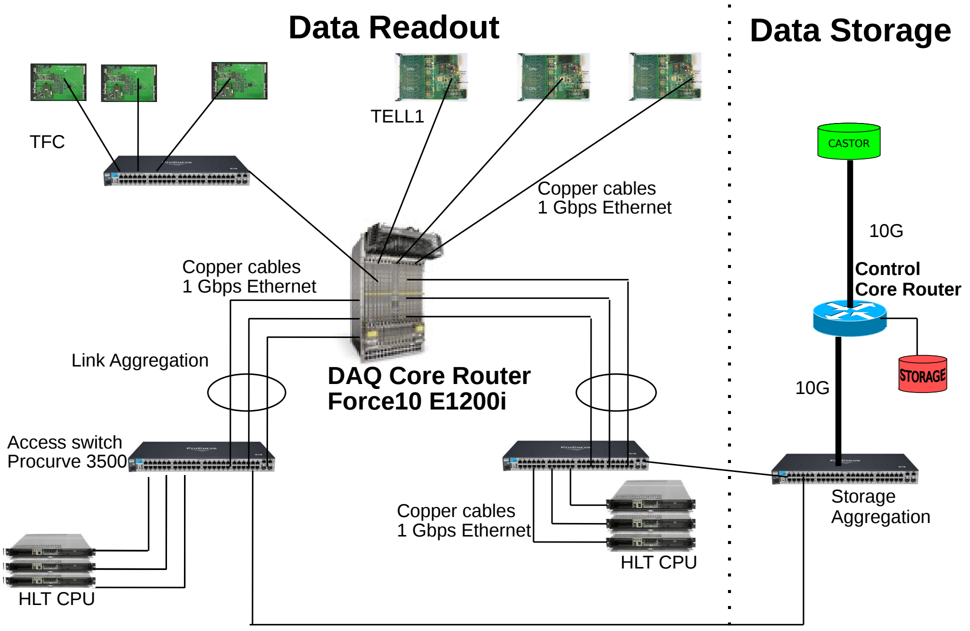


Figure 2.9: Tree Structure of the DAQ Network

As described in Section 2.1.2, each TELL1 has four GbE ports, but the actual number of links in use are calculated from the traffic, which is according to simulation at the beginning and can be adjusted for the real data during the operation of the experiment. By design, the load on the links shall be inferior to

60% of their capacity, in order to minimize the probability of packet loss due to buffer overflow during some temporary big bursts.

The E1200i switch is one of the highest density Gigabit Ethernet (GbE) switch available to date, it supports up to 14 linecards of 90 GbE ports, namely 1260 GbE ports in total. The E1200i switch has a 3.5 TeraBit/sec (Tbps) switch fabric capacity, and 2 Billion Packet/sec (Bpps) full-mesh forwarding capacity. There are two portpipes in each linecard. The portpipe is an unique concept for Force10 switch, it acts as a backplane communication channel that provides connectivity between slots and the switch fabric modules. The buffers for both input and output traffic in each portpipe are 256 MB. The ports number and bandwidth capacity of this switch completely fulfill the requirement of the LHCb readout network as shown in Table 2.1.

In the core switch E1200i, the event data from the TELL1s are routed to the edge switches; then the edge switches further forward the data to the HLT CPUs. For the data readout, the edge switches are operated at Layer 2. The CPU farm is physically installed in 50 racks, where a single edge switch HP Procurve 3500 is installed in each rack and connected to the core switch E1200i.

In the LHCb DAQ system, the dataflow is always unidirectional except for a very small amount of protocol traffic. The data readout is supervised by the central readout supervisor ODIN. When the TELL1s receive the destination information from ODIN, they will all send the traffic to the same HLT node at approximately the same time. Data from about 300 sources concentrate on one output port, the switch will have to buffer most of the packets from the same event, this will require a quite large buffer size in the switch. Even though the E1200i switch can work with full line rate, the input and output load should be balanced in each portpipe, hence the input ports gain more input buffer while output ports gain more output buffer, so that the performance of the DAQ network will be improved dramatically.

There are three methods to connect the core router and the edge switches in the HLT farm:

- 10 Gigabit Ethernet (10 GbE)

  The 10 GbE connection will need optical modules in both sides, which will introduce more cost, and it is not possible to interleave input ports and output ports because they use different linecards. So 10 GbE is not suitable.

- VLAN based concentration
  To avoid network loop, we have to create a VLAN for each uplink. Even though the performance is perfect (see the test report from the Tolly Group [44]), it is not easy to manage so many VLANs and it does not provide load balancing nor redundancy.

- link aggregation (LAG) [35]
  LAG, also called trunk, aggregates multiple links into one virtual link. It provides high bandwidth and increases the redundancy for higher availability with the same hardware, without extra cost. With proper configuration, the performance is almost as perfect as VLAN based concentration.

By comparing the three methods, the LAG connection is adopted because it has a high performance, and provides high redundancy without extra cost. The configuration and performance test will be described later in Section 2.3.2.3.

The events selected by HLT will be sent to LHCb local storage and then to the CERN CASTOR for permanent storage. For the data storage, the edge switches work at Layer 3, which route the data traffic to the storage aggregation switch as shown in Fig. 2.9. Through the core control routers underground and on the surface, the data will be transferred to the local storage cluster, where all the selected events are written into raw data files. The raw data files are registered in a run-database and transferred to the CERN mass-storage facility CASTOR as soon as they are closed.

2.3.2.3 Configuration and Performance of Link Aggregation

The LAG standard (IEEE 802.3ad) allows the aggregation of multiple link segments in order to obtain a higher bandwidth interconnection between devices. The actual benefits vary based on the load-balancing method used on each device and the traffic pattern. There is no standard mechanism defined for load balancing frames on the individual links within a LAG. The LAG hashing algorithms are very different in the devices from different vendors. The only requirement of the 802.3ad standard is to preserve the temporal order of the frames. Depending on the load balancing algorithm implementation, some links may be over-utilized, while others have spare capacity or are even idle.

There are 12 physical links between the core router and each edge switch in HLT farm, hence we could have up to 12 GbE uplinks. Since the maximum port member of a LAG is eight in the edge switch, the uplinks are divided into two LAGs. There are several hashing methods provided in the E1200i switch [33]:

- 5-tuple hashing: distributes IP traffic based on IP source address, IP destination address, Protocol type, TCP/UDP source port and TCP/UDP destination port.
- 3-tuple hashing: distributes IP traffic based on IP source address, IP destination address and IP protocol type.
- packet-based hashing: distributes IPV4 traffic based on the IP Identification field in the IPV4 header.

In the LHCb dataflow, all the traffic has the same protocol type, similar IP source address and IP destination address, so the 3-tuple and 5-tuple hashing method can not distribute the traffic very evenly. But the packet-based hashing is very suitable for our traffic. The data packets belong to the same collision have the same IP identification even they are from different TELL1s, because the IP identification is related to the collision event ID[1]. All the packets from the same event pass though the same link, and the total event size is quite constant for each collision, the destination links to the HLT nodes are assigned in a round-robin mode, so the traffic is balanced in the LAG very efficiently.

To verify the setup, the performance has been tested. In this test setup, six GbE links are aggregated in a LAG, and a throughput close to 6 Gb/s can be achieved. As illustrated in Fig. 2.10, we used the HLT nodes to perform the test with application iperf [45]. UDP traffic is generated in the HLT nodes, source1 to source6, and is sent to the destination HLT nodes through the LAG under test.

[1] This is a feature of the MEP protocol.

To achieve the desired traffic rate, the rate limit on the output ports are tuned in the Procurve 3500 switch sw1 to sw6.

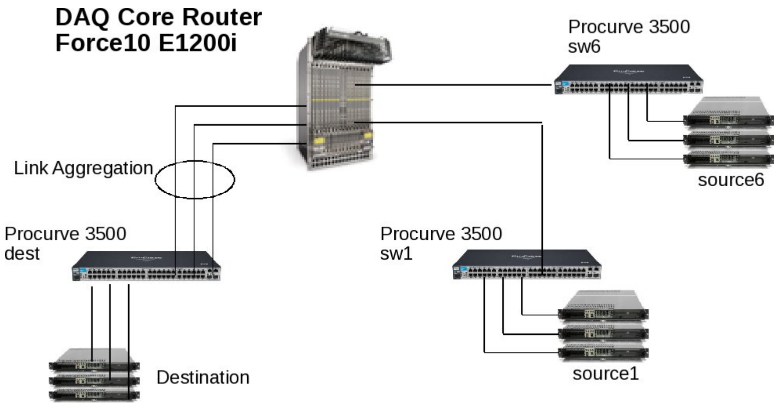


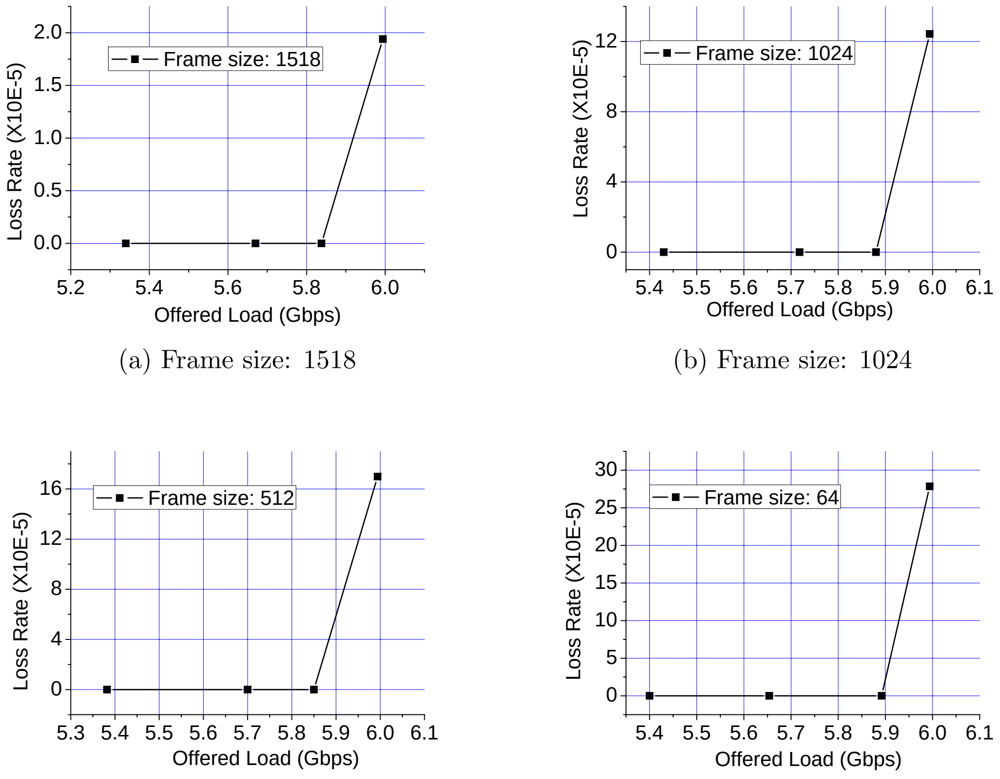
Figure 2.10: The setup for LAG performance test

We have tested the packet loss rate at different traffic rates with the frame size of 1518 Byte, 1024 Byte, 512 Byte and 64 Byte. There was no packet loss observed when the traffic rate reached about 5.83 Gb/s (97% of the capacity) for all frame sizes. Only when the traffic almost reaches the full capacity, packet loss was observed. The loss rates are 1.94E-5, 1.24E-4, 1.70E-4 and 2.78E-4 when the frame sizes are 1518 Byte, 1024 Byte, 512 Byte and 64 Byte respectively.

In this test, the IP identifications of the generated packets are consecutive with a increment of 1. In the real data, the IP identifications of the packets pass through the same LAG are not consecutive, this might affect the distribution of the traffic and slightly degrade the performance.

2.4 Summary

The Online system is one of the infrastructures for the LHCb experiment, providing all IT services. It consists of three major parts: DAQ, ECS and TFC, and it employs two large networks based on Gigabit Ethernet separately for DAQ and ECS. The architecture of the Online network is primarily finished in 2005. However, a lot optimizations have been done to improve the performance of the



(a) Frame size: 1518

(b) Frame size: 1024

(c) Frame size: 512

(d) Frame size: 64

Figure 2.11: LAG performance test result

networks, such as redundancy and flexibility, and these improvements has been test and verified.

Chapter 3

Network Monitoring with a SCADA System

As described in Section 2.2, LHCb uses a commercial SCADA system called PVSS for its experiment control system. SCADA tools are ideal for the management of large, heterogeneous industrial or experimental installations. The LHC experiments have based their entire control systems on this technology. For the efficient operation of the experiment it is crucial that all parts of the system are monitored and controlled through the same interface in a coherent way. The network monitoring is part of the LHCb ECS and it should be implemented in the same framework.

Modern networks are becoming increasingly fast and complex. Traditionally network management and monitoring is based on a few open standards, such as Simple Network Management Protocol (SNMP), Remote Network Monitoring (RMON) and sFlow. So far no integration of advanced network management and monitoring exists in any SCADA system, even though SNMP is supported but at least in PVSS not suitable for large scale network management. In this chapter, I will show the implementation of the network monitoring system using SCADA tools which are originally for industrial supervisory control.

3.1 Introduction to SCADA, PVSS and JCOP

3.1.1 Supervisory Control And Data Acquisition (SCADA)

SCADA stands for Supervisory Control And Data Acquisition. It is a technology to collect data from one ore more distant facilities and/or send limited control instructions to those facilities. SCADA systems are used extensively in industry for the supervision and control of industrial processes, and in some experiment facilities. A SCADA system gathers information (such as temperature, humidity, voltage), transfers the information back to a central site, then carries out necessary analysis and control, and displays the information in a logical and organized fashion [46].

A SCADA system normally consists of:

- data servers, usually Remote Terminal Units (RTUs), and/or Programmable Logic Controllers (PLCs), which interface to field sensing devices.
- A communications system used to transfer data between the data servers and the computers in the SCADA central host.
- A central host computer server called Master Terminal Unit (MTU), used to process the data.
- A collection of standard and/or custom software used to provide Human Machine Interface (HMI).

SCADA systems typically provide a flexible, distributed and open architecture to allow customization to a particular application area. The standard SCADA functionality can be summarized as follows [47]:

- Data acquisition
- Data logging and archiving
- Alarm handling
- Access control mechanism

- Human Machine Interface including many standard features e.g. alarm display, trending

A generic architecture of SCADA systems is shown in Fig. 3.1.

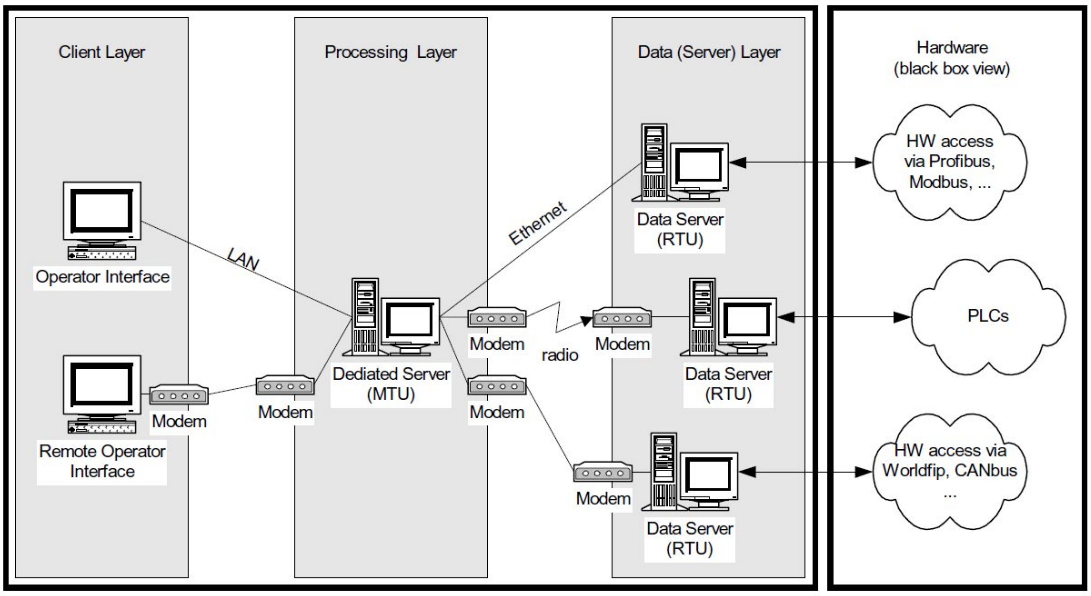


Figure 3.1: A generic architecture of SCADA system

3.1.2 PVSS

PVSS[1] is a commercial SCADA product from the Austrian company ETM. A PVSS application is composed of several processes, in PVSS nomenclature: Managers. These Managers communicate via a PVSS-specific protocol over TCP/IP. PVSS implements a client-server architecture with the event manager (EVM) acting as a server for all other managers. Managers subscribe to data and this is then only sent on change by the Event Manager, which is the heart of the system. PVSS has a highly distributed architecture as shown in Fig. 3.2 [29, 48, 49]

[1] A German acronym: Prozess Visualisierungs und Steuerungssystem, in English a system for visualization and control of processes

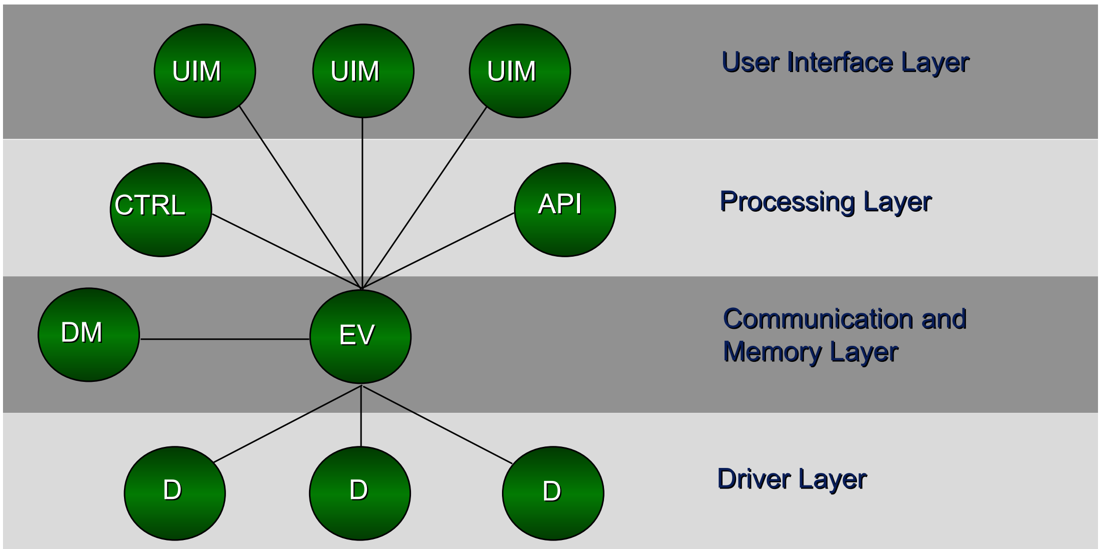


Figure 3.2: Architecture of PVSS system

- Event Manager (EVM) is responsible for all communications. It receives data from Drivers (D) and sends to the Database Manager to be stored in the data base.

- DataBase Manager (DBM) provides the interface to the (run-time) data base.

- User Interface Manager (UIM) is used for the visualization of process status information and for forwarding user input to the event manager.

- Ctrl Managers (Ctrl) provide for any data processing as “background” processes, by running a scripting language.

- API Managers (API) allow users to write their own programs in C++ using a PVSS API (Application Programming Interface).

- Driver Managers (D) provide the interface to the devices to be monitored / controlled.

PVSS allows users to design their own graphical user interfaces, which are called “Panels”. These will be created with the graphic editor GEDI. A Control program ("script") is used to control PVSS, either in panels or as stand-alone processes. A script is normally employed to specify the response of PVSS to an event. The Control program processes datapoints and controls the visualization of process states, and it is capable of multi-program operation (multi-threading). The Control syntax is based on the procedural programming language C.

Another essential concept of PVSS is its flexible data point concept. All data in the PVSS system is organized in so-called Data Points (DP) of a pre-defined Data Point Type (DPT). A DPT defines a data structure consisting of one or several typed data point elements (DPE), and are similar to structs in C. DPEs can themselves be structures and thus have subordinated elements. A DP designates a instance of a DPT, contains particular information. Finally, so-called configs can be inserted into DPEs and DPs. Configs are pre-defined sets of configuration information that are used for multiple purposes, e. g. for archiving, alert-handling and data smoothing.

3.1.3 The JCOP framework

The Joint Controls Project (JCOP) [50, 51] was setup as a collaboration between the CERN IT department and the LHC experiments, in order to reduce duplication, and hence reduce the overall manpower required for the development of the LHC experiment control systems. The JCOP Framework provides an integrated and coherent set of guidelines, devices and tools for the different teams to build their control applications in a coherent manner.

The JCOP framework is based on PVSS. It creates a layer on top of PVSS and the other tools to simplify their use for the application developer and also to provide customized components that can be used directly in an experiment control system. The JCOP framework provides the core components to build FSM of the experiment control. In addition it provides a set of tools to build the control system, e.g. archiving, trending, and it provides complete components for commonly used equipment.

3.2 Architecture of the network monitoring system

Network management refers to the activities, methods, procedures and tools that pertain to the operation, administration, maintenance and provisioning of networked systems. A *Network Monitoring System* (NMS) is a computer program that assists the network administrator in the typical tasks that are involved in running a computer network: configuration, health and performance monitoring, troubleshooting. According to the ITU standard model for network management, the following categories of functions (also known as FCAPS) that should be included in a management system [52]:

- Fault Management: alarm handling, trouble detection, trouble correction, test and acceptance, network recovery
- Configuration Management: system turn-up, network provisioning, autodiscovery, back up and restore, database handling
- Accounting Management: track service usage, bill for services
- Performance Management: statistical data collection, report generation, data analysis
- Security Management: security and access control

The idea of FCAPS is very useful for teaching network management functions, but a system that covers all the above categories is too complicated to be implemented. Most NMS, both commercial and open-source products, can only handle a part of the network management job. The main goal of our network management system is to monitor the health status and traffic/performance of the network, provide a limited set of interfaces to configure network devices. And it should be integrated into the LHCb experiment control system. Since the features of our network management system are most for monitoring, we call it a network monitoring system as well. Below is a list of functional requirements:

- Health monitoring (Listening to alarms and notifications)

- Traffic/performance monitoring
- Topology auto-discovery and monitoring
- Visualization of network states
- Integration with the LHCb ECS

The architecture of the network monitoring system is shown in Fig. 3.3. Similar to other SCADA system, the network monitoring system consists of several components: data collectors, data communication, data processing and user interfaces. All the components will be described in detail in the following sections.

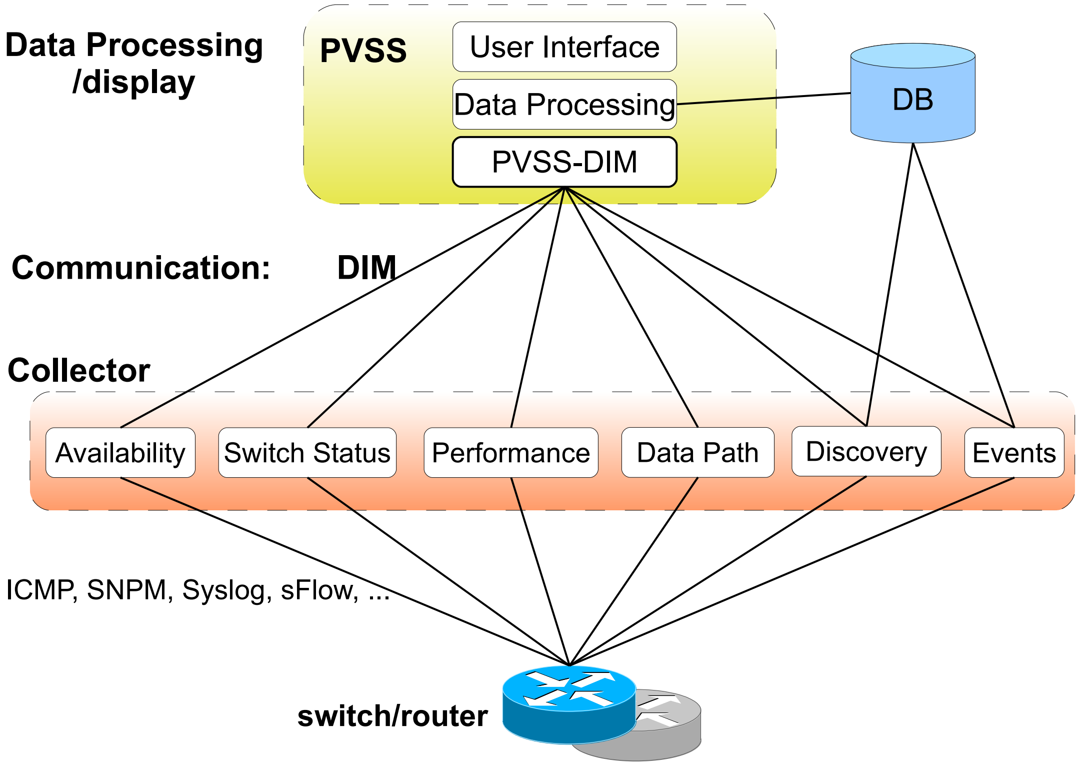


Figure 3.3: Architecture of the network monitoring system

3.2.1 Data communication: DIM

For the information exchange between the front-end data collectors and PVSS, we use Distributed Information Management (DIM) [53, 54], which is widely used in the LHC experiment control systems. DIM provides a network transparent interprocess communication layer in a distributed/mixed environments. As most of the communication systems, DIM is based on the client/server model: servers provide services to clients. The operation mechanism is shown in Fig. 3.4.

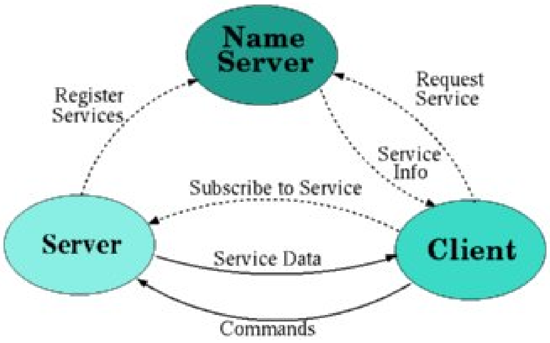


Figure 3.4: Mechanism of DIM

There are three components in the system: name server, server, client. The name server keeps an up-to-date directory of all the servers and services available in the system. Servers "publish" their services by registering them with the name server (normally once, at start-up). A service is recognized by a name. Clients "subscribe" to services by asking the name server which server provides the service and then contacting the server directly, providing the type of service and the type of update as parameters. Whenever one of the processes (a server or even the name server) in the system crashes or dies, all processes connected to it will be notified and will reconnect as soon as it comes back to life.

A service is normally a set of data (of any type or size). Services are normally requested by the client only once (at start-up) and they are subsequently automatically updated by the server either at regular time intervals or whenever the conditions change (according to the type of service requested by the client).

The PVSS-DIM [55] toolkit allows to interface PVSS to devices which do not provide any protocols supported by PVSS (such as OPC-OLE for Process Control). It provides a bridge between PVSS datapoints (either full structures or single items) and DIM services. PVSS communicates with DIM via a PVSS-DIM API Manager: `PVSS00dim`. `PVSS00dim` can be configured either via a configuration file or via a configuration datapoint. The PVSS - DIM package allows PVSS Data Points to behave as:

- Client Services - DP content is updated when the DIM Server updates the Service.
- Client Commands - DP content is sent as a DIM Command whenever it changes.
- Server Services - DP Content is published at start up and updated whenever it changes.
- Server Commands - DP Content is updated when a DIM Client sends a command.

With DIM and PVSS-DIM toolkit, the data from the collectors can be transferred to PVSS efficiently in a easy way in our network monitoring system.

3.2.2 Front-end data collectors

To monitor the network state, a lot information is collected from the network devices. There are several mechanisms to collect data for network management, including Simple Network Management Protocol (SNMP), sFlow, Internet Control Message Protocol (ICMP), command-line interface (CLI) and syslog. In the following sections, collectors based SNMP and sFlow will be explained in detail.

3.2.2.1 SNMP collector

SNMP [56, 57] is a UDP-based network protocol which is widely used for the network management. It is an application layer protocol that facilitates the exchange of management information between network devices. SNMP is based on the manager/agent model consisting of a manager, a SNMP agent, a database of management information, managed objects and the network protocol.

The SNMP agent resides in the managed device, provides the interface between the manager and the managed physical device. The SNMP manager and agent use an SNMP Management Information Base (MIB) and a small set of commands to exchange information. The SNMP MIB is organized in a tree structure with individual variables represented as leaves on the branches, an example is shown in Fig. 3.5. A long numeric tag or object identifier (OID) is used to distinguish each variable uniquely in the MIB and in SNMP messages.

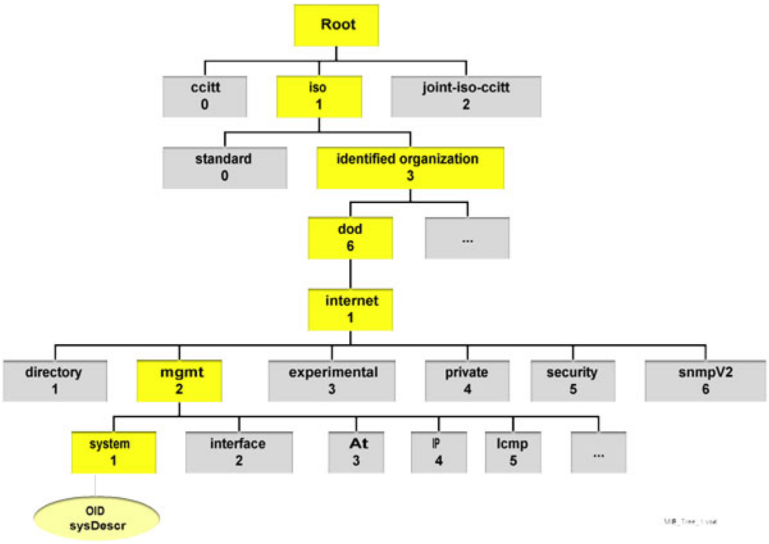


Figure 3.5: An example of MIB tree

Managed devices are monitored and controlled using the following basic SNMP commands: **Get**, **GetNext**, **GetBulk**, **Set**, **Trap**. The Get and GetNext operation is used by a NMS to obtain a single piece of information from a managed device. The GetBulk operation allows NMSs to obtain multiple pieces of information from a managed device in a single request. The Set command is used by a NMS to configure a managed device with a particular value. A Trap is initiated from a managed device to alert a NMS about the fact that a threshold has been reached or that an error/event of some type has occurred.

To implement a network monitoring system a lot information needs to be collected through SNMP. A SNMP driver is provided in PVSS, but the performance is insufficient for a large system. To improve efficiency, a custom front-end process called SNMP collector has been implemented. The SNMP collector not only collects data from the network devices, but also act as a DIM server publishing the information. It logically consists of two modules: the SNMP module and the DIM module.

The SNMP module deals with encoding and decoding of SNMP messages, and transports the SNMP messages over UDP. The SNMP module has been implemented based on the Net-SNMP [58] suite, which provides a suite of C library and application to facilitate the implementation of SNMP applications. The basics steps of SNMP operations are:

1. Initialize an SNMP session: a SNMP session structure will be initialized with default values

2. Define attributes for the session: e.g. the SNMP version, community name and target hostname.

3. Create a PDU (Protocol Data Unit) with SNMP operation type

4. Pack OIDs into the PDU

5. Send the request and wait for response

6. Process the returned values

7. Free the PDU

8. Continue with the other SNMP queries by repeating the above steps

9. Close the session

Each SNMP request packet includes a PDU, which can contain one or more OIDs. The type of request is specified by the type of PDU. If you simply needed to read more OIDs, you could just pack them into a single PDU. Once a PDU is prepared, it can be sent using the session handle returned earlier. The response

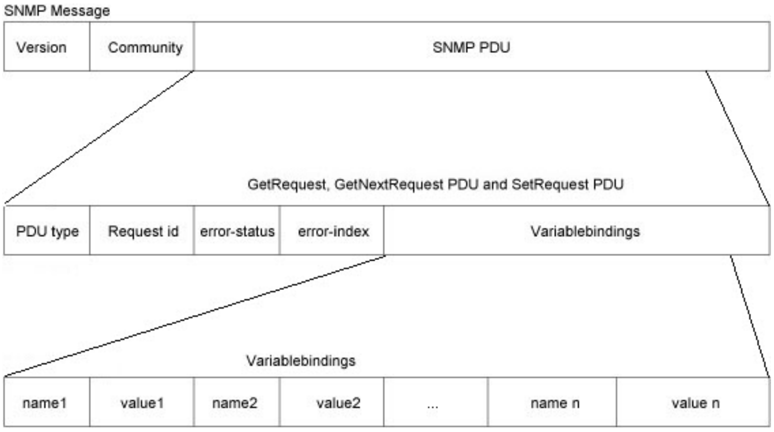


Figure 3.6: SNMP packet format

is put into a new PDU structure with both the OIDs and the values. The format of SNMP packet is shown in Fig. 3.6.

In our network monitoring system, all the information collected through SNMP are either integers (32 bit or 64 bit) or strings. First, a few fundamental library functions for SNMP basic operations have been implemented, e.g. snmpGetInt, snmpGetInt64, snmpGetStr snmpBulkWalkInt, snmpBulkWalkInt64, snmpBulkWalkStr and others. With these library functions, information is collected from the network devices, e.g. system description, all input/output traffic counters for all ports (octets, packets, errors, discards) and other information.

The SNMP collector is also a DIM server. To improve the efficiency and make the management of DIM service/client easier, the number of DIM services should be kept as small as possible. So the data of the same type will be packed into one service. For example, the traffic counters for all interfaces in the switch are provided in a single service. All the services are updated in a fixed interval or by the request from the clients.

3.2.2.2 sFlow

Because there are so many ports (1260) in the core switches, the SNMP query of interface counters (1260 × 10 counters) takes a long time and occupies a lot CPU and memory resource in the switch. The sFlow collector has been implemented to get the interface counters, while SNMP is still used for querying the other information of the core switches.

Originally developed by InMon, sFlow [59, 60] is a standard for monitoring high-speed switched and routed networks. sFlow specification (RFC 3176) [61] and its first implementation were both launched in 2001. sFlow is a sampling technology to collect statistics from the device based on hardware, hence it is applicable to high speed networks.

A sFlow system consists of sFlow agents and a sFlow collector. The sFlow collector is a central server which collects the sFlow datagrams from all agents and analyzes them. An sFlow agent is the implementation of the sampling mechanism on the hardware, it provides two types of sampling: statistical packet-based sampling of packets and time-based sampling of interface counters.

- Flow samples: Based on a defined sampling rate, 1 out of N packets is captured and sent to a collector server. The details will be described in Chapter 4.

- Counter samples: A polling interval defines how often the sFlow octet and packet counters for a specific interface are sent to the collector.

The architecture of the sFlow counter collector is shown in Fig. 3.7. The system consists of a number of software components. The first part is the sFlow collector, which is used to collect sFlow packets from network devices. The sFlow collector decodes the sFlow PDU, and forwards the counter samples to the sFlow counter collector. The packet structure of counter samples is shown in Fig. 3.8. Counter samples cover all the traffic counters we can get from SNMP. This will not consume much computing resources because the actual ingress/egress processing is implemented in the ASICs, rather than in the host CPU of the switch.

When the sFlow counter collector receives the counter samples, it further decodes the counter samples and passes the counters to the counter parser. The counters will be further processed, and stored in the shared memory based on the interface index (see Fig. 3.8). The DIM server retrieves the data from shared memory and updates the services periodically.

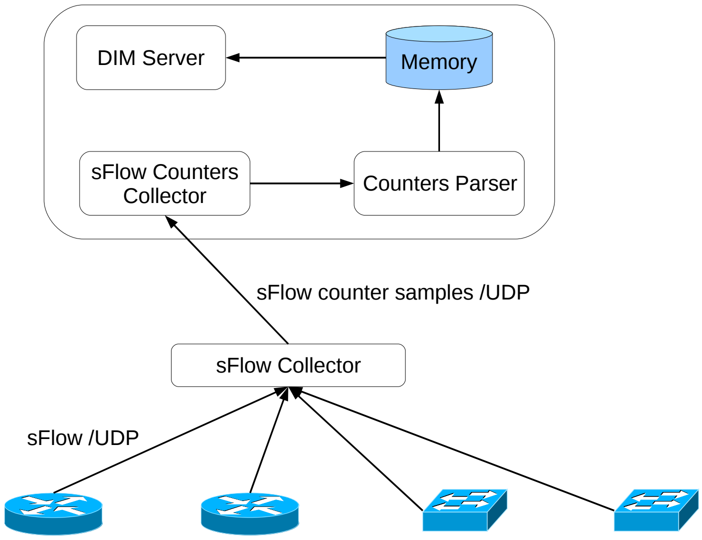


Figure 3.7: Architecture of the sFlow counter collector

3.2.3 Supervision layer in PVSS

As shown in Fig. 3.2, there are several layers in the supervision of PVSS. With the bridge provided by PVSS-DIM, PVSS receives data from the data collectors which also provide data as DIM servers, then processes the received data and represents to users by user interfaces. The whole supervision layer is implemented with the control script language and built-in features of PVSS.

3.2.3.1 Data modeling

As introduced in Section 3.1.2, PVSS uses datapoint type and datapoint to represent the process variables. In order to work with PVSS and create user interfaces for users, the bearer of the presented information must be defined. A datapoint type is a form of template for structured datapoints. Before you can create a structured datapoint as a representative of a device, a corresponding datapoint type must be created as a template. It is built up of one or several, almost arbitrarily structured datapoint elements, whose values present the current process information. Datapoint elements can represent quite different information, not only the common binary and analog variables but also higher and structured data

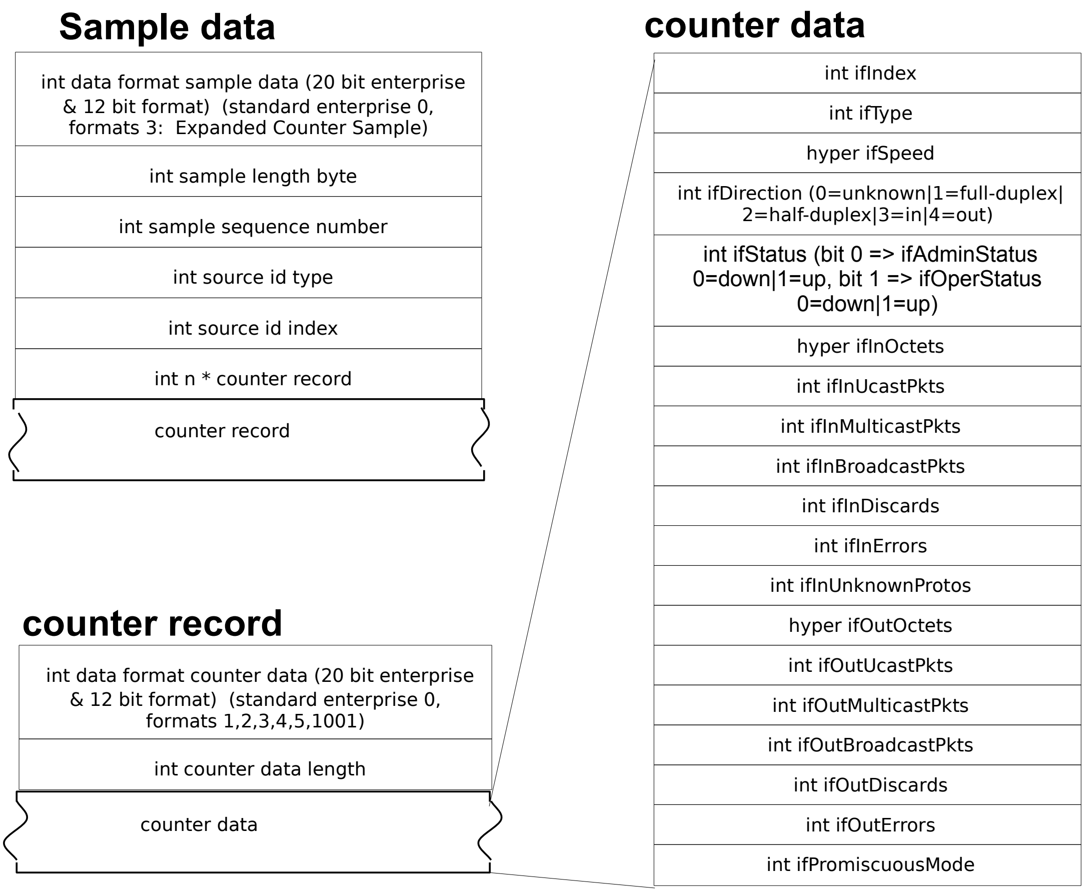


Figure 3.8: Packet format of sFlow counter samples

types. Datapoint types are merely a form of template for the datapoints, they alone cannot save any process information. Datapoints are the instances of the datapoint types, each datapoint can represent a real device or a logical combination of information. Datapoint types and datapoints always exist project wide. Fig. 3.9 shows one datapoint type and its datapoint (instance).

In addition to the value, a datapoint element normally possesses a number of other properties called “Configs” in PVSS. These are informative attributes but are used to define processing and alarming methods, e.g. value range, alarm handling, archiving and others. The setting of the config of datapoint elements is flexible, even the same elements on the datapoints of the same datapoint type

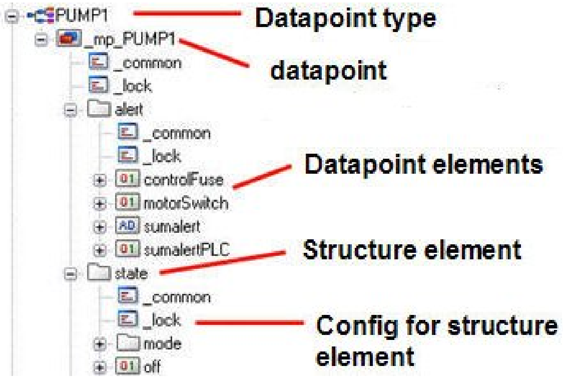


Figure 3.9: Example: datapoint type and datapoint

can have different configs.

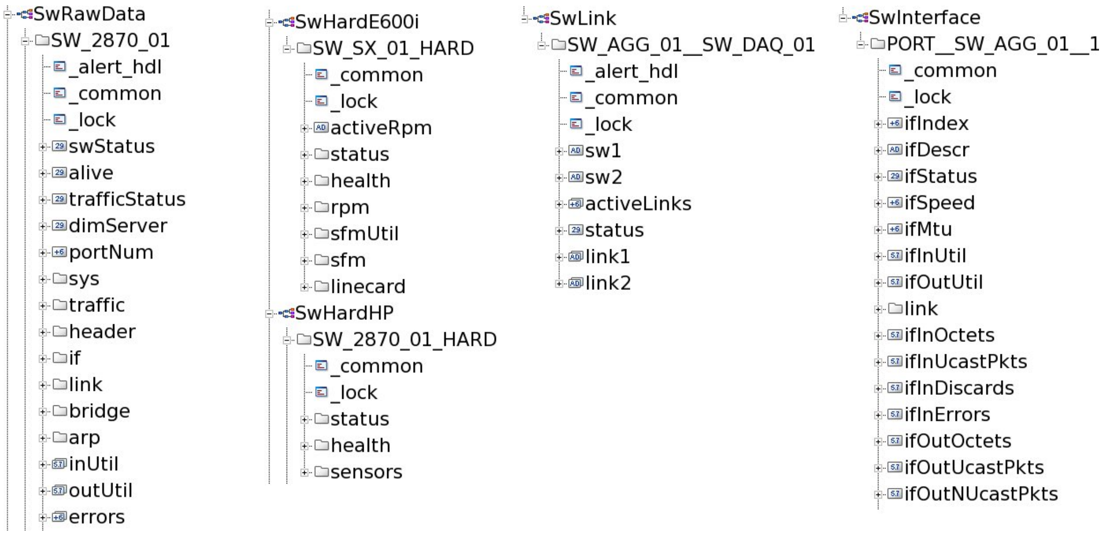


Figure 3.10: Instances of datapoint types and datapoints

There are several kind of switches in the LHCb Online network, and a lot of information is needed to build the network monitoring system. Some of the information is common to all devices, e.g. the system information, the interface counters and ARP table, while the others are quite hardware specific. Correspondingly, the common datapoint type "SwRawData" for all switches has been created to collect the generic information, and the hardware-oriented datapoint types have been created for each switch model, see some instances in Fig. 3.10. The datapoint type “SwRawData” is to represent the general information and the traffic of the switch, it consists of the the DPEs for: the generic information of the system, the interface information (description, speed, status.) and counters (input/output octet, input/output unicast packet, input/output error, input/output discard.). While the hardware specific datapoint type is to represent the hardware information and the health status, so it is build up of the DPE for temperature, power supply status, fan status, CPU load, memory usage, switch modules and others.

The datapoint type “SwInterface” has been created for a physical or logical interface. In most of the time, the network information, like traffic, are display to users based on interfaces. The datapoint type “SwInterface” is very convenient to represent the status of switch interfaces. The datapoint of interface receives the processed data which originally received from the switches and stored in the datapoint of “SwRawData”.

In order to represent the uplink between switches, a datapoint type “SwLink” has been created. This is based on switch-switch, instead of interface-interface. It shows the links between these two switches and the current status.

3.2.3.2 Data processing

With PVSS-DIM bridge, the project configures the datapoints as client services to receive data published by DIM servers, subscribing to the services provided by DIM servers. For the convenience of the data transfer and DIM service management, the data are always grouped based on their property. For example, all the counters of all interfaces are provided by a DIM service in specific format. Once the data are received in PVSS, they are decoded and further processed so that the data can be displayed in a user-friendly way.

As introduced in Section 3.1.2, the individual managers cooperate in a real client-server architecture. The servers execute their processing independently from the client and provide information. The client simply receives and consumes the information from the server. The processing of data between the individual managers in PVSS takes normally place event orientated. The value changes are processed immediately, vice versa there is no communication and processing load without value changes. This make the system work very efficiently.

The necessary structures are provided to build the connection between different managers. After successful connect, new values of the data sources in the server managers are forwarded automatically to the client manager and are processed, see Fig. 3.11 .

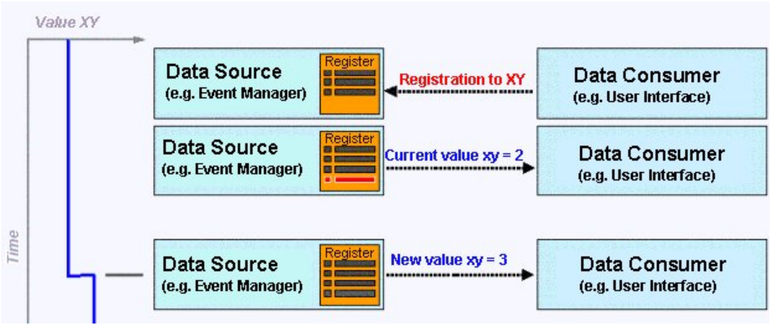


Figure 3.11: Process of an event orientated communication connection

The function **dpConnect()** in the script library allows a callback function to register to attributes of individual datapoint elements.

```
int dpConnect (string work, [bool answer,] string dpe1 [, string
dpe2 ...]);
```

The callback function work() is executed immediately once the values of the datapoint attributes dpe1 [, dpe2 ...] are changed. Most of the data processing take place in the registered callback functions, depending on the data sources.

For different models in the project, the methods to process data are varied, details will be described in each part in Section 3.3.

3.2.3.3 User interfaces

The user interfaces [62] (also known as human computer interface or man-machine interface) is the aggregate of means by which people (the users) interact with the system, a particular machine, device, computer program or other complex tool. The user interface provides means of input and output.

The network monitoring system needs to provide information at several levels of expertise:

- Critical conditions and alarms for the non-expert shift-crew
- long term health-monitoring for the network administrators
- Performance info for data acquisition experts

To provide these features, the project takes great advantage of the alarm handling, archiving and trending tools provided by PVSS and JCOP.

An alarm is a property that can be configured per datapoint element. A certain alarm state can be assigned to a range of values of the element. A number of properties are additionally assigned to each alarm range, for example: an alarm class comprises alarm priority and alarm message.

The value of DPEs can be archived in the Oracle database using relational database (RDB) archiving manager. Archiving saves the original value, source time and all status bits of a DPE. The archiving allows to display history value / events in trend displays or reports, which is essential to analyze the problem or to optimize the system performance.

PVSS provides a mechanism to plot process variables on a trend widget. And the JCOP framework extends this features by providing a simple framework oriented configuration of the displays, which provides a few templates that are instantiated at run time with a given set of parameters. The JCOP trending tool is widely used in the network monitoring system for traffic display and status display.

3.3 Modules of the network monitoring system

3.3.1 Topology discovery and monitoring

In large and constantly growing networks, knowledge of a network's physical topology (the physical connections between network devices) is useful in a variety of network scenarios and applications. The topology information is extremely important for many critical network management tasks, including network trouble shooting, resource management, new hardware installation and server replacement. Manual network mapping is becoming increasingly difficult (if not impossible) due to the size and dynamic behavior of networks. Accurate topology information, cannot be practically maintained without the aid of automatic topology discovery tools. Automatic topology discovery tools and algorithms will therefore play an important role in network security, management and administration.

There are three different levels at which to describe the network topology: the link layer topology, the network layer topology[1] and the overlay topology [63].

- Link layer topology: the physical layout of the network, i.e. the physical connections between switches and routers and between computers and switches. Link layer maps are essential for the management of the network.
- Network layer topology: the logical map of the network seen by the IP routing layer, containing only the connections between routers.
- Overlay topology [64]: an application layer virtual network topology. An overlay network is a computer network which is built on top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network. For example, many peer-to-peer networks are overlay networks

As described in the survey of Kamal A. Ahmat [63], many research efforts concerned with topology discovery have been carried out with different algorithms and methods. We do not attempt to study the topology discovery in full complexity, we only present the issues relevant to the LHCb Online networks.

As discussed in Chapter 2, the LHCb Online network has a quite plain architecture. In the control network, the backbone is formed with two core routers, while the other switches only works at Layer 2. The DAQ network consists of one core router and also 50 edge routers, but the topology is quite straightforward. So we will only focus on the link layer topology.

[1] sometimes referred to the internet topology

3.3.1.1 Data sources

The Ethernet standard does not offer the possibility of interrogating the switches about their neighbors. One way to discover the close neighbors is to use the local knowledge from each switch in order to find the inter-switch physical interconnections and node-switch connection. These information includes:

1. Link Layer Discovery Protocol (LLDP)

   For the discovery of network devices, IEEE has developed the Link Layer Discovery Protocol (LLDP) [65], now part of the 802.1AB-2005 standard [66]. LLDP is a vendor-neutral Data Link Layer protocol, specifically defines a standard method for Ethernet network devices such as switches, routers and wireless LAN access points to advertise information about themselves to other nodes on the network and learn the information they discover from their neighbours. Information gathered with LLDP is stored in the device as a MIB and can be queried with SNMP as specified in RFC 2922 [67]. The topology of an LLDP-enabled network can be discovered by crawling the hosts and querying this database. Information that may be retrieved include:

   - Remote chassis ID and description
   - Remote port ID and description
   - Remote management IP address

2. Media Access Control (MAC) address and forwarding database

   In computer network, a MAC address is an unique identifier assigned to network devices by the manufacturer for identification. MAC addresses function at the data link layer (layer 2 in the OSI model). They allow network devices to uniquely identify themselves on a network and communicate to each other.

   The FDB (forwarding database) table is used by a Layer 2 device (switch / bridge) to store the MAC addresses that have been learned and which ports that MAC address was learned on. The MAC addresses are learned through transparent bridging on switches and dedicated bridges. When a

Ethernet frame arrives at a Layer 2 device, the Layer 2 device will inspect the destination MAC address of the frame and look to its FDB table for information on where to send that specific Ethernet frame. If the specific MAC address is found in the FDB table, the Ethernet frame will be forwarded to the corresponding port, otherwise it will be flooded to all ports in the broadcast domain.

3. Address Resolution Protocol (ARP) table

   ARP is used to resolve an IP address into a MAC address. An IP address is the address of a host at the network layer and can be configured by software. To send a network layer packet to a destination host, the device must know the MAC address of the destination host. So the IP address must be resolved into the corresponding MAC address.

   When a Layer 3 device has an IP packet which needs to deliver, it will look to the ARP table to figure out what MAC address to put into the packet header. If there is no ARP table entry for the destination IP address, the Layer 3 device will broadcast an ARP request message in the broadcast domain. Once it gets the ARP response, the MAC address for that specific IP address will be learned, and the ARP entry will be used to forward the packets. The ARP table is populated as devices issue ARP broadcasts looking for a network devices MAC address.

3.3.1.2 Discovery of switches

In the LHCb Online network, all the switches support LLDP, which make it much easier to discover all the uplinks between the switches. We use the three core routers as seeds, then discover their neighbors by querying the SNMP MIB for LLDP, with the information of remote chassis ID (normally is MAC address depends on the ID type), remote port ID (normally interface index) and remote management IP address, the neighbor switches can be easily identified. Then the neighbor is used as a new seed to discover its neighbor, and so on until all devices in the network are discovered. Based on the LLDP information, the map of the network topology can be learned accurately.

3.3.1.3 Discovery of end-nodes

We use passive method to discover the end-nodes without sending any active probe message. With the MAC address and ARP entries, the connection for end-nodes should be able to found out, but it is not so straightforward. A simplified example is shown in Fig. 3.12.

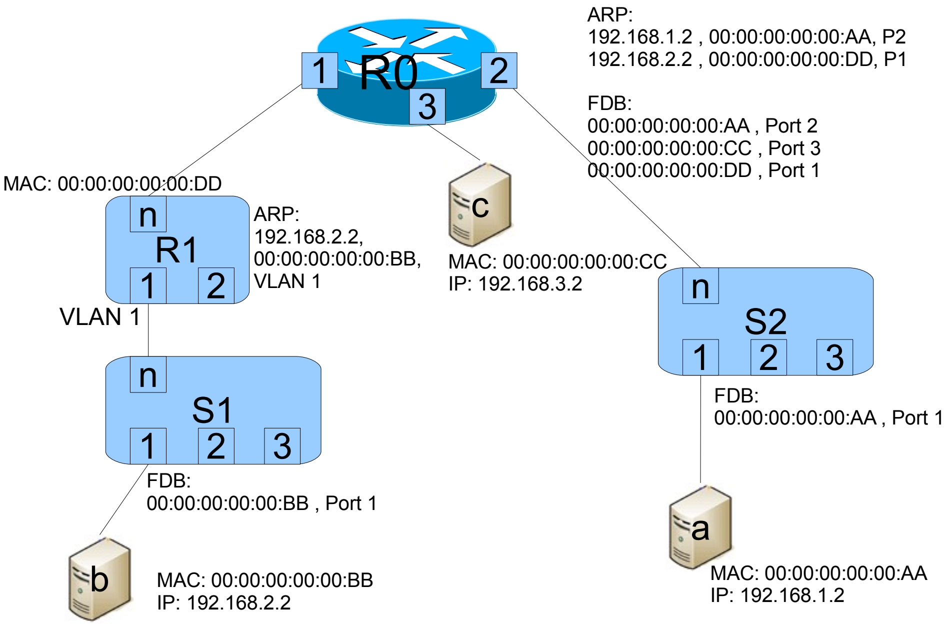


Figure 3.12: Example of the topology discovery

The basic rule to discover the end-nodes is get the MAC address from the FDB table in the switch, then look up the ARP table to find the IP address for it, the information of the node will be learned in this way, but there are several problems to solve:

1. The ARP table in the switch provides the information of the IP, MAC address and interface, but the interface is virtual interface like VLAN, instead of physical port, so we can not use the interface information.
2. The MAC addresses learned on the uplink port should not be recognized as directly attached nodes.

3. For a node, the MAC address entry is learned in the directly connected switch (normally it is a Layer 2 switch), while the ARP entry is learned in a router or Layer 3 switch. Normally these two are not the same switch.

4. For a node, the ARP entry are usually learned in variable switches, the right switch which is closest to the node should be selected.

5. For the MAC address that relates to logical interface, we needs to find out the physical port.

The main procedure for a switch to scan the attached nodes and finds the connections is presented in Table 3.1. First, we collect all information from LLDP, ARP table and FDB, and then traverse all the forwarding entry learned in the FDB. If the port of the forwarding entry is an uplink, it means this can not be a direct attached node, so the entry will be discarded. Otherwise, we will search the matching ARP entry, which possesses the same MAC address, in the local ARP table, if not found then search in the global ARP table, which contains all ARP entries learned in the known routers. After the traversal of the FDB, the current switch will be marked as known switch, while the new discovered switch will be added to to-do list. And the local ARP table entries will be added into the global ARP table if the MAC addresses don't belong to the neighbor switches. If the MAC address belongs to a neighbor switch X, it means the neighbor switch X is also a router, the host is closer to the neighbor switch X, so the ARP entry in current switch is not an accurate one.

For the scanning of the global network, the three routers are appended to the global variable *To-do* list as the seeds, and the procedure above is performed for them, and then for each discovered switch. For example to discover the topology of the network in Fig. 3.12, the scan starts from R0, it will find the neighbor switch: R1, S2. And the MAC address {*00:00:00:00:00:CC*} matches the ARP entry {*192.168.3.2 , 00:00:00:00:00:CC, P3*}, so a node is discovered {*192.168.3.2, P3*}. And MAC address {*00:00:00:00:00:AA*} does not belong to any switch, so the entry {*192.168.1.2, 00:00:00:00:00:AA, P2*} is appended to *Global-ARP-Table.* After the procedure is finished for the core router R0, the scan of S2, R1 will start. S2 is merely a Layer 2 switch, the MAC address

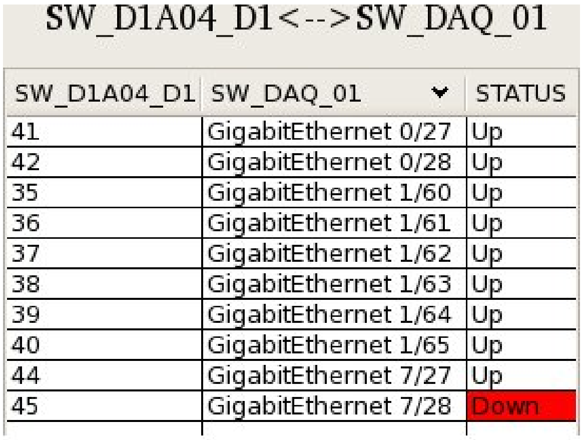
```
Topology discover for switch S
Global variables: Global-ARP-Table, Found-Sw, To-do
Discover the neighbor switches: Neighbor and the uplink port UL
Gather ARP table: ARP-Table
Gather forwarding table: FDB
for f in FDB do
    if f.port not in UL then
        if f.port is logical-port and physical-port in UL then
            continue
        end if
    if f.mac in ARP-Table then
        build the entry {f.port, IP(hostname)}
        Remove the ARP entry from ARP-Table
    else if f.mac in Global-ARP-Table then
        build the entry {f.port, IP(hostname)}
    end if
end for
Append S to Found-Dev
for newSw in Neighbor do
    if newSw not in Found-Sw then
        Append newSw to To-do
    end if
end for
for arp in ARP-Table do
    if arp.mac not in neighbors then
        Append arp to Global-ARP-Table
    end if
end for
```

Table 3.1: Procedure to discover the end-nodes

{*00:00:00:00:00:AA*} is found and matches the entry in *Global-ARP-Table*, so the node is discovered {*192.168.1.2, S1-P3*}. The procedure for the remaining switches are the same until the end.

3.3.1.4 Topology monitoring

Topology monitoring is an important part of the network monitoring system, any unintentional change of topology always means something is wrong with the network. The monitoring of topology is based on LLDP. As discussed in Section 3.2.3.1, datapoint type "SwLink" is created for the uplink between two switches. First, a link table between two switches is built using information of LLDP, and maybe some need to be added manually if the link is down initially. This is the complete table for the uplinks. When the LLDP information of the switches is updated, a callback function will check all the uplinks of the datapoint. If any of the uplink is not present in the current switch, which means the link is down, then an alarm will be issue. If some new links are found, they will be appended to the uplink table. Fig. 3.13 is the example of an uplink table.



SW_D1A04_D1<-->SW_DAQ_01

SW_D1A04_D1	SW_DAQ_01	STATUS
41	GigabitEthernet 0/27	Up
42	GigabitEthernet 0/28	Up
35	GigabitEthernet 1/60	Up
36	GigabitEthernet 1/61	Up
37	GigabitEthernet 1/62	Up
38	GigabitEthernet 1/63	Up
39	GigabitEthernet 1/64	Up
40	GigabitEthernet 1/65	Up
44	GigabitEthernet 7/27	Up
45	GigabitEthernet 7/28	Down

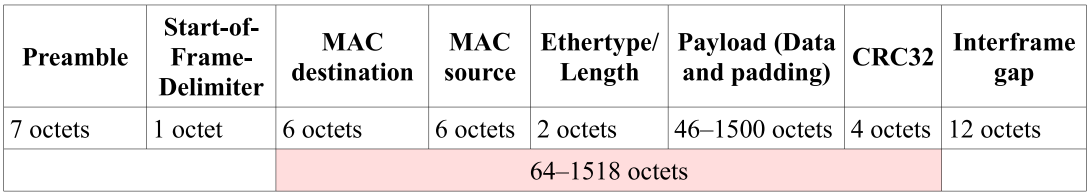
Figure 3.13: Uplink table

3.3.2 Network traffic monitoring

In the LHCb Online cluster, the DAQ network is dedicated for data acquisition which is extremely performance critical. It is essential for the operation of the DAQ to monitor the bandwidth usage and network performance sophisticatedly.

SNMP is the most popular method of gathering bandwidth and network usage data. It can be used to monitor bandwidth usage of routers and switches port-by-port. The following information is collected from the switches via SNMP [68]:

- **ifInOctets/ifOutOctets**: the total number of octets received / transmitted on the interface, including framing characters.
- **ifInUcastPkts/ifOutUcastPkts**: the number of unicast packets received / transmitted on the interface.
- **ifInNUcastPkts/ifOutNUcastPkts**: the number of broadcast and multicast packets received / transmitted on the interface.
- **ifInDiscards/ifOutDiscards**: the number of inbound / outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol.
- **ifInErrors/ifOutErrors**: the number of inbound / outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.



Preamble	Start-of-Frame-Delimiter	MAC destination	MAC source	Ethertype/ Length	Payload (Data and padding)	CRC32	Interframe gap
7 octets	1 octet	6 octets	6 octets	2 octets	46–1500 octets	4 octets	12 octets
		64–1518 octets					

Figure 3.14: 802.3 Ethernet MAC frame

With the counters above, the bandwidth utilization can be calculated with the consideration of the structure of the Ethernet MAC frame [69], which is shown in Fig. 3.14. The ifInOctets/ifOutOctets is only the octet number in the MAC frames, but the preamble, start-of-frame-delimiter[1] and the inter-frame gap[2] are not included. The formula used to calculate the bandwidth utilization is:

$$Utilization = \frac{(\Delta Pkts * (12 + 7 + 1) + \Delta Octets) * 8}{Interval * ifSpeed} \tag{3.1}$$

After the calculation, the bandwidth utilization will be stored in the individual datapoint for each interface and be archived for further analysis and trending display.

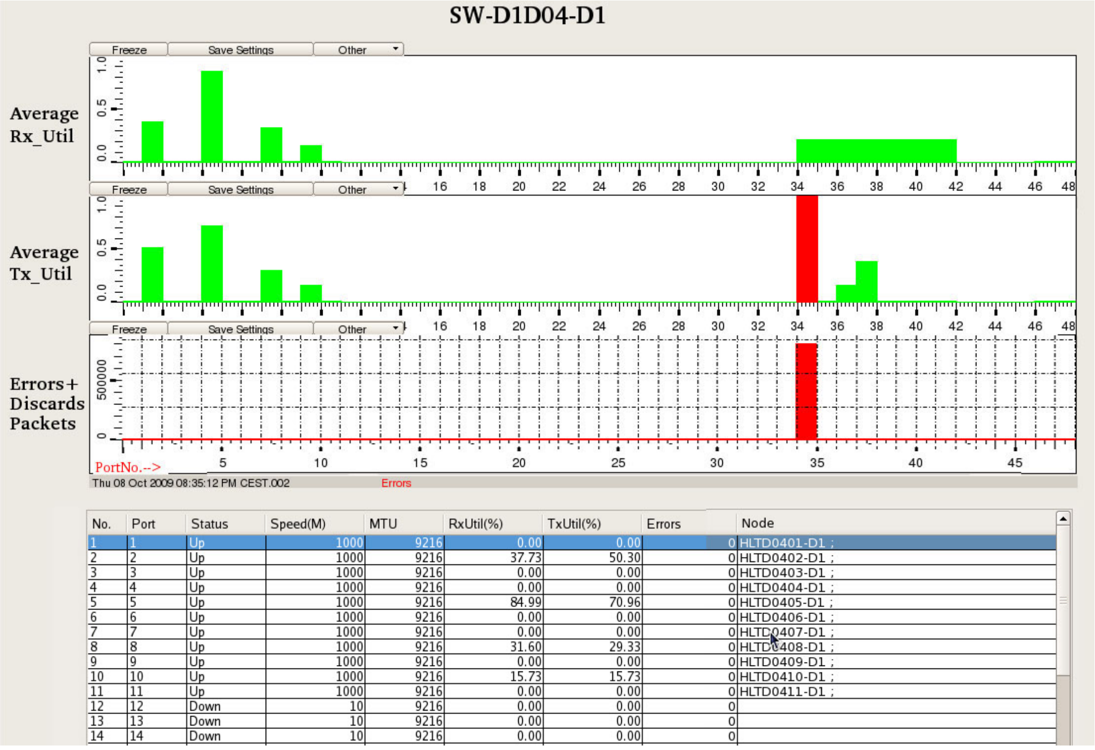
The network monitoring system provides real-time bandwidth monitoring of all ports in the switch, see the panel in Fig. 3.15. The graphical representation of the bandwidth usage gives a clear visibility into how much bandwidth is consumed on a specific switch. The panel also presents the error and discard packet counters in the switch, which can monitor the packet loss and is critical for DAQ network. When the bandwidth utilization exceeds the threshold or any error/discard packet occurs, the bar will become red which indicates a problem. In addition to the graphical view, the table in the lower part of the panel provides the accurate values.

With all the value archived, the network monitoring system can provide the trending of bandwidth utilization for an individual port, see Fig. 3.16, which shows the usage patterns and network traffic trends across an individual period. With such statistics, the administrators can analyze how bandwidth is used in the network, and decide whether an increase in bandwidth or a modification of network configuration is required. The panel also provides the history counters in detail, including all the counters mentioned above, which provides important information to isolate and troubleshoot network problems. In addition to the trending for an individual port, the trending plot is provided for the global throughput in switches as shown in Fig. 3.17.

The graphical view clearly represents the status of traffic, but it is unpractical to watch the panels all the time, these panels are used normally when we

[1] The preamble is a 64-bit (8 byte) field that contains a synchronization pattern consisting of alternating ones and zeros and ending with two consecutive ones. After synchronization is established, the preamble is used to locate the first bit of the packet.

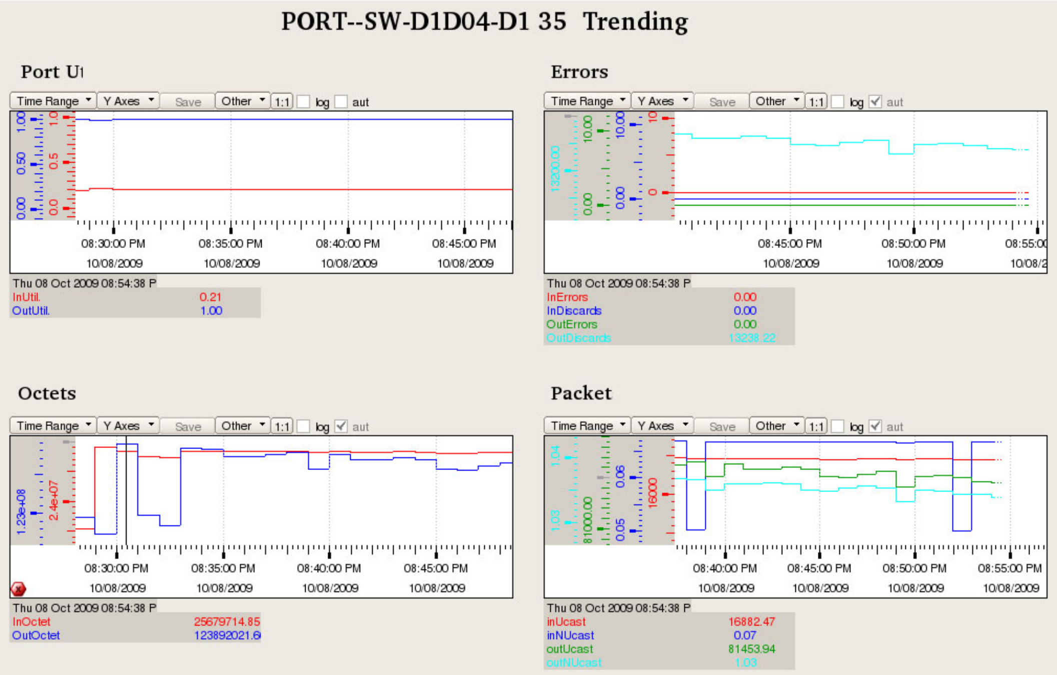
[2] A brief recovery time between frames allows devices to prepare for reception of the next frame.



No.	Port	Status	Speed(M)	MTU	RxUtil(%)	TxUtil(%)	Errors	Node
1	1	Up	1000	9216	0.00	0.00	0	HLTD0401-D1 ;
2	2	Up	1000	9216	37.73	50.30	0	HLTD0402-D1 ;
3	3	Up	1000	9216	0.00	0.00	0	HLTD0403-D1 ;
4	4	Up	1000	9216	0.00	0.00	0	HLTD0404-D1 ;
5	5	Up	1000	9216	84.99	70.96	0	HLTD0405-D1 ;
6	6	Up	1000	9216	0.00	0.00	0	HLTD0406-D1 ;
7	7	Up	1000	9216	0.00	0.00	0	HLTD0407-D1 ;
8	8	Up	1000	9216	31.60	29.33	0	HLTD0408-D1 ;
9	9	Up	1000	9216	0.00	0.00	0	HLTD0409-D1 ;
10	10	Up	1000	9216	15.73	15.73	0	HLTD0410-D1 ;
11	11	Up	1000	9216	0.00	0.00	0	HLTD0411-D1 ;
12	12	Down	10	9216	0.00	0.00	0	
13	13	Down	10	9216	0.00	0.00	0	
14	14	Down	10	9216	0.00	0.00	0	

Figure 3.15: Real-time bandwidth utilization and error monitoring

troubleshoot network problems. For the normal operation, the alarm system is more important, which informs the operator and network administrator once a problem is detected. In the network monitoring system, thresholds are set for bandwidth utilization and error/discard packet. When the bandwidth utilization or error/discard packet rate exceeds the prescribed limits, an alarm will be triggered with a priority (e.g. Info, Warning, Error, Fatal) depends on the value. This allows a quicker isolation of the problem in the network and hence a quicker action. For example, if the non-unicast packet rate become abnormally high and bandwidth utilization almost reach the line rate of the port, this usually means a network loop in Layer 2.



PORT--SW-D1D04-D1 35 Trending

Port Ut

```
Thu 08 Oct 2009 08:54:38 P
InUtil.                  0.21
OutUtil.                 1.00
```

Errors

```
Thu 08 Oct 2009 08:54:38 P
InErrors                 0.00
InDiscards               0.00
OutErrors                0.00
OutDiscards              13238.22
```

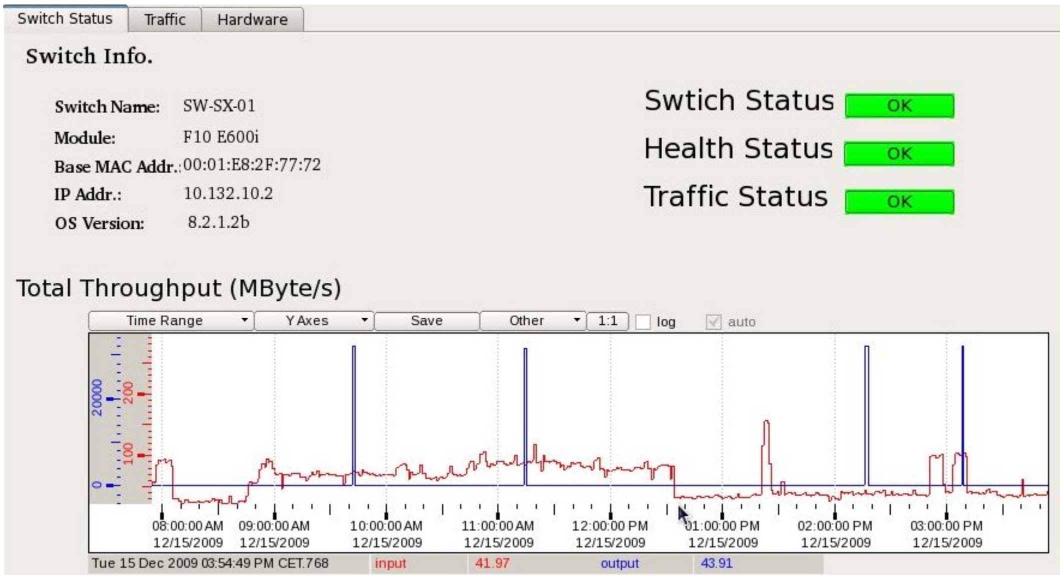
Octets

```
Thu 08 Oct 2009 08:54:38 P
InOctet                  25679714.85
OutOctet                 123892021.6
```

Packet

```
Thu 08 Oct 2009 08:54:38 P
inUcast                  16882.47
inNUcast                 0.07
outUcast                 81453.94
outNUcast                1.03
```

Figure 3.16: Trending of traffic for an individual port



Switch Info.

Switch Name: SW-SX-01

Module: F10 E600i

Base MAC Addr.:00:01:E8:2F:77:72

IP Addr.: 10.132.10.2

OS Version: 8.2.1.2b

Swtich Status OK

Health Status OK

Traffic Status OK

Total Throughput (MByte/s)

```
Tue 15 Dec 2009 03:54:49 PM CET.768   input   41.97   output   43.91
```

Figure 3.17: Trending of traffic throughput for switches

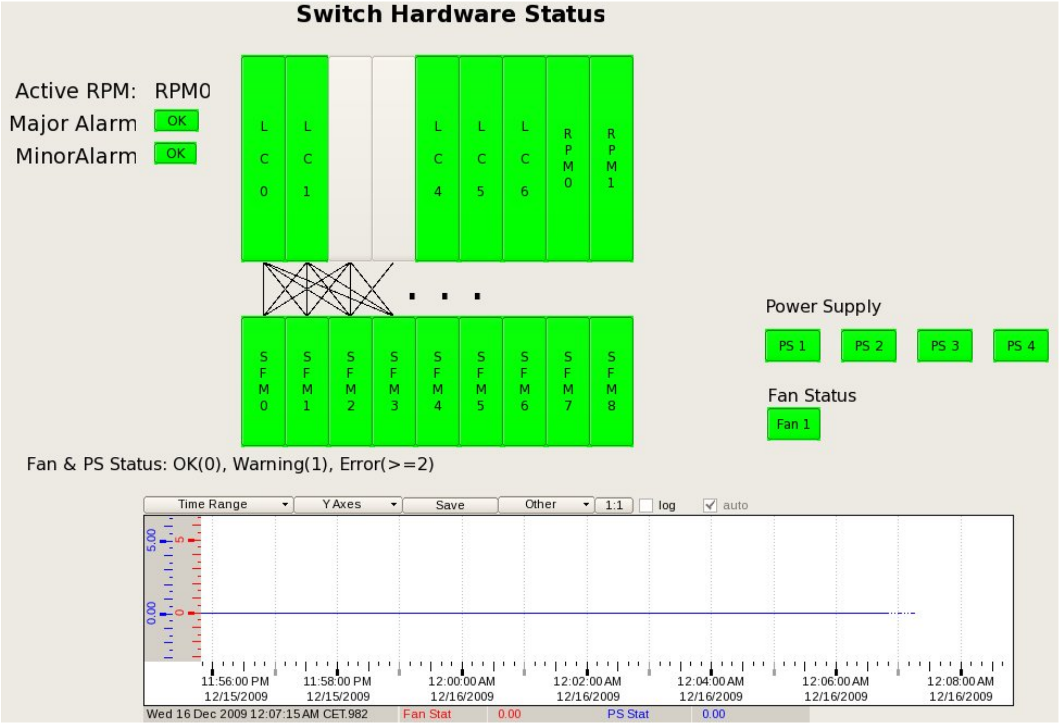
3.3.3 Network device status monitoring

For a long term operation, it is important to monitor the health status of network switches, e.g. power supply, fan status, temperature, CPU, Memory and so on. Compared with the traffic monitoring, the monitoring of health status is quite hardware oriented. Different devices have different hardwares and sensors, and of course different OIDs and values. The network switches in LHCb Online are most from two vendors: HP and Force10 Networks. Those switches provided by the same vendor, will have quite similar hardware structure and sensors, despite that the models are different. For the HP switches, the items to be monitored are: power supply status, fan status, temperature, usage of CPU and Memory, while the Force10 switches have much more complex structure hence more specific monitoring is required. Force10 switches [33] consist of three major function components: Route Processor Modules (RPM), Switch Fabric Modules (SFM) and Line Card (LC). RPM has three CPU and LC has one CPU inside the module, and they both have temperature sensors inside as well. So in addition to the global items like power suppliers and fans, CPU/Memory usage and temperature in RPM and LC need to be monitored, Fig. 3.18 is a snapshot of the monitoring of Force10 switches.

3.3.4 Monitoring of data path for DAQ and interfaces to external network

It is critical to monitor the routing status in each stage for the DAQ. In the operation of 2008, some edge switches stopped routing traffic after a power-cut due to a bug in the operating system. As discussed in Chapter 2, there are three stages from the networks point of view for the LHCb DAQ: event data from the readout boards to the HLT farm, selected event data from the HLT farm to the LHCb Online storage, raw data files from the LHCb Online storage to CERN CASTOR. In all these stages, the data transfers are at Layer 3, and the traffics are routed through varied paths. A simple tool to monitor availability is ping[1],

[1] ping is a program that uses the IP/ICMP protocol to send echo request packets and expects to receive back echo replies. It is used to test IP connectivity


Switch Hardware Status

Active RPM: RPM0
Major Alarm OK
MinorAlarm OK

Power Supply: PS 1 PS 2 PS 3 PS 4

Fan Status: Fan 1

Fan & PS Status: OK(0), Warning(1), Error(>=2)

Wed 16 Dec 2009 12:07:15 AM CET.982 Fan Stat 0.00 PS Stat 0.00

Figure 3.18: Monitoring of health status: F10 switch

which can tell if a host is reachable in an IP network. We can get the routing status by analyzing the responses to ping from the hosts.

In the readout board (TELL1), full features of ICMP are not implemented, ping can not be launched for the test. To monitor the data path from TELL1s to the HLT farm, a monitoring server is used to simulate TELL1, and it is connected to the same VLAN as TELL1s in the core switch. The monitoring server scans the availability of the HLT nodes with ping since both ends fully support ICMP. A multi-threaded program is launched from the monitoring server to ping all the HLT nodes. The purpose here is not to check the status of the HLT nodes but the status of the routing, we will consider the status to be good if more than half of the HLT nodes in the same sub-net reply to ping.

For monitoring the routing from the HLT nodes to the LHCb Online storage, the method is similar. The monitoring server has an interface connected to the storage subnet, and the same program is launched to scan the availability of the

HLT nodes for the storage nodes.

For monitoring the routing from the LHCb storage to CERN CASTOR, the method is similar except for the different destination nodes. For this scanning, the destinations are a set of nodes in the CERN computing center, and they are allowed to access the LHCb Online storage by the firewall. The list of trusted nodes are fetched dynamically from the CERN LANDB, which is a network database used for the management of CERN campus network.

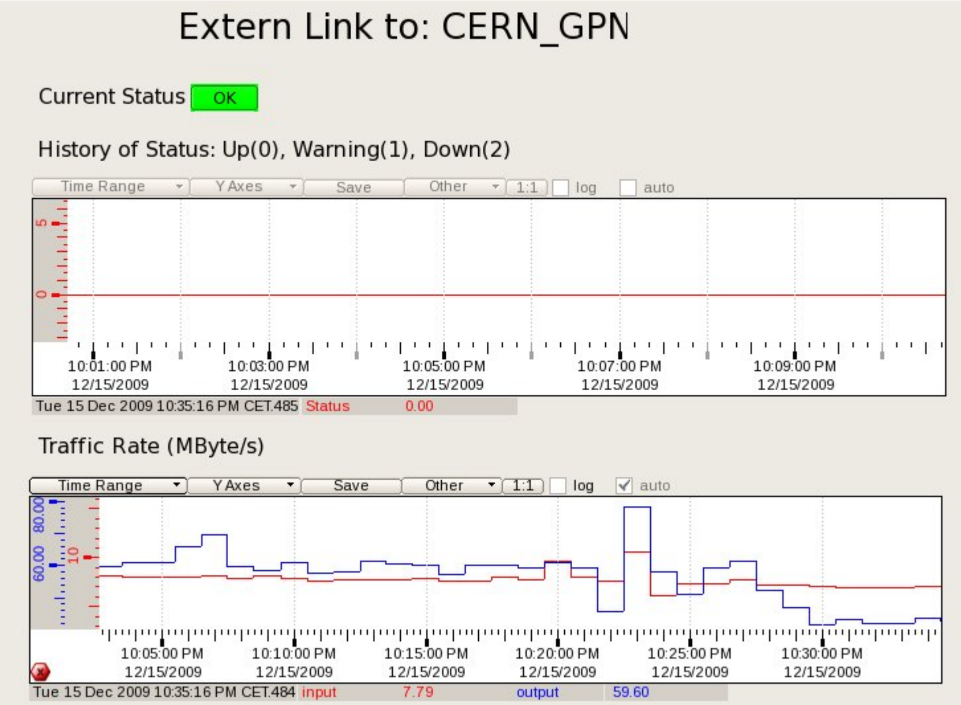
The scanning program is implemented in Python. For each stage, the result is summarized and published via DIM using PyDIM [70]. PyDIM is a Python interface to DIM, and allows to create DIM clients and servers, using an API very similar to the one used for C. In the DIM service, the status of routing are published for every stage. The DIM clients of the network monitoring system subscribe to these DIM services, and will receive the updates for each scan. The current status and history status of data paths are shown in the PVSS panel.

The interfaces to external networks, namely CERN networks, are critical for experiment control and data moving, and also important for user to access the LHCb Online cluster. As discussed in Section 2.3.1.2, the LHCb Online network have three interfaces to CERN GPN, TN and LCG network, the status and traffic rate of these three interfaces are under monitor, as shown in Fig. 3.19, both current values and history trending.

3.3.5 Syslog message monitoring

When network switches run into problems, they will generate a log in the local storage, and in the mean time send a syslog [71] and/or SNMP trap [56] to the designated server (if configured) to inform the network administrators. For syslog and SNMP trap, most of the information is duplicated, but there are events where a device will only generate a trap and others where it only generates a syslog entry, normally the different part is not critical. But SNMP Trap has a big and ever increasing MIB, it is easier to monitor the message with syslog.

Syslog is a standard used by devices and applications to send log messages in a IP network. It is typically used for computer system management and security auditing. The syslog packets are sent in clear text, and have three distinct parts:



Extern Link to: CERN_GPN

Current Status OK

History of Status: Up(0), Warning(1), Down(2)

Tue 15 Dec 2009 10:35:16 PM CET.485 Status 0.00

Traffic Rate (MByte/s)

Tue 15 Dec 2009 10:35:16 PM CET.484 input 7.79 output 59.60

Figure 3.19: Status and traffic rate of external links

priority, header and message. The priority part represents both the facility and severity of the message. A facility in syslog is a class of messages, the facility NEWS is assigned to the network devices in the LHCb Online cluster. The facilities and severities of the messages are numerically coded with decimal values, the message severities are shown in table 3.2. With the severities, it is easy to judge if it is a critical message.

A syslog server to receive remote syslog message has been implemented using Python. The syslog server is setup to listen on port 514 for receiving the syslog messages from the network switches, while the network switches have been configured to send the syslog message to the designated server. When there is any activity or change, or switches run into problems, a syslog messages will be generated and sent to the syslog server as configured. Once the syslog messages is received in the server, the syslog server will log all the messages in a text file. For those messages with higher priority above warning (Numerical Code <= 4),

Numerical Code	Severity
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

Table 3.2: syslog Message Severities

the syslog server logs the message into database, and publishes it in the DIM server. the PVSS network monitoring system subscribes to these DIM services, and it will issue an alarm if a syslog message is received.

The syslog is a complementarity to the SNMP query in the network monitoring system. With the syslog server, we can always get the message if some errors occur, e.g. high temperature, fan failure. This helps isolate the network device problem, and allows the network administrators to take some proactive actions before a real network problem occurs.

3.3.6 FSM for network monitoring: the user point of view

The LHCb ECS control system is a hierarchical, reactive system built as a tree of interconnected control nodes which is implemented using Finite State Machines (FSM), the network monitoring system uses the same framework as well.

FSM is used to model the behavior of a system by means of a limited number of states, transitions between these states, actions and events. And the concept of FSM can be applied to a wide range of applications in both hardware and software. The FSM component is provided by the JCOP framework based on the functionality of PVSS and the State Management Interface (SMI++) [31]. Using the JCOP FSM toolkit, a hierarchical control system can be build easily.

At the higher levels of the FSM tree, the control nodes called control units (CU), allow an operator to take the control of the associated sub tree of the system, and at the very lower level, the control nodes called devices units (DU) connect to the real hardware components that they supervise. In the hierarchical architecture, commands are sent from the top of the tree down to bottom DU, while the state changes propagate along the tree from the hardware component that changes its status, up to the control node CUs. The FSM represents the state of each sub-system, provides convenient mechanism to model the functionality and behavior of a component.

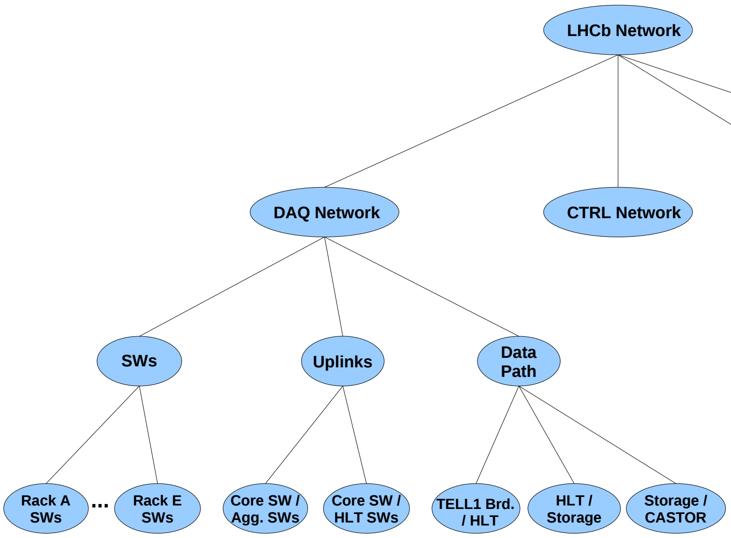
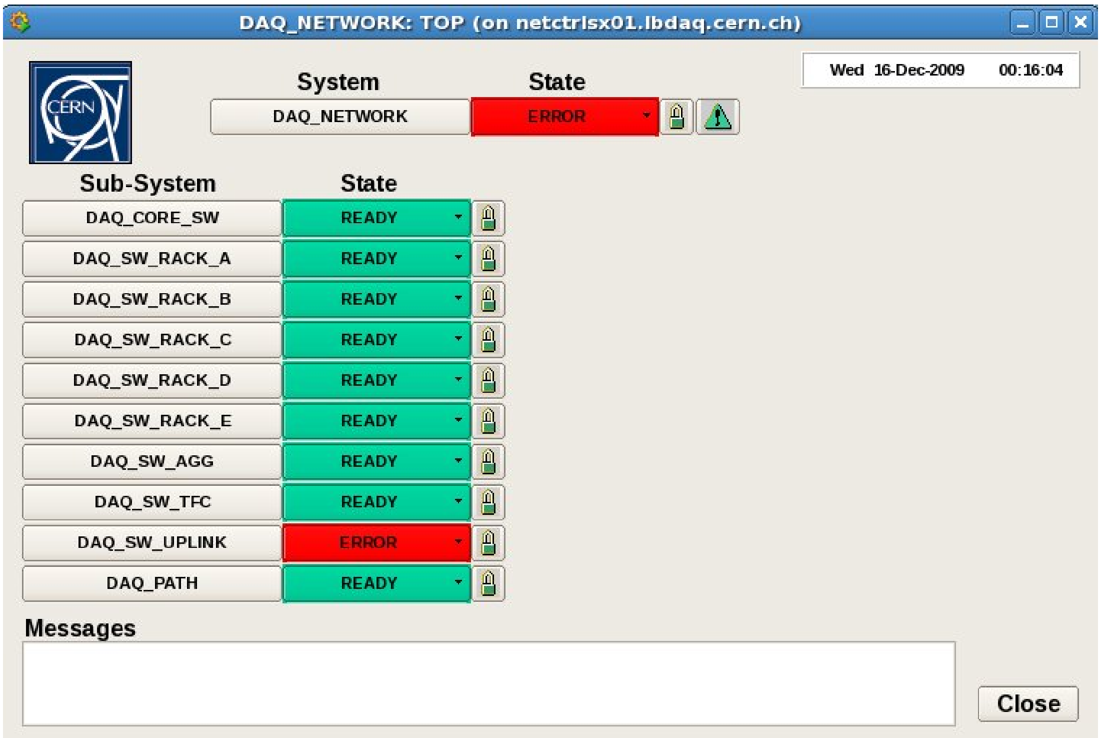


Figure 3.20: FSM tree of the network monitoring system

In the network monitoring system, all states and behaviors are modeled as FSM, the architecture is shown in Fig. 3.20, all the models discussed in this section will be integrated into this tree. According to the guide for the FSM design [72, 73] and the properties of the readout network, three states (Ready, Error and Off) are defined to indicate the state of the LHCb Online network. The top layer is composed of four CUs: DAQ network, CTRL network, Storage network and external uplinks to CERN network. The former three CUs have almost the same structure, so only the DAQ network is shown. In the DAQ network, there are three groups of objects under monitor: the switches, the uplinks between switches and the data path for DAQ. The state of the switch includes the hardware status and the traffic throughput status. The FSM for the DAQ network is shown in Fig. 3.21.



DAQ_NETWORK: TOP (on netctrlsx01.lbdaq.cern.ch)

Wed 16-Dec-2009 00:16:04

System	State
DAQ_NETWORK	ERROR

Sub-System	State
DAQ_CORE_SW	READY
DAQ_SW_RACK_A	READY
DAQ_SW_RACK_B	READY
DAQ_SW_RACK_C	READY
DAQ_SW_RACK_D	READY
DAQ_SW_RACK_E	READY
DAQ_SW_AGG	READY
DAQ_SW_TFC	READY
DAQ_SW_UPLINK	ERROR
DAQ_PATH	READY

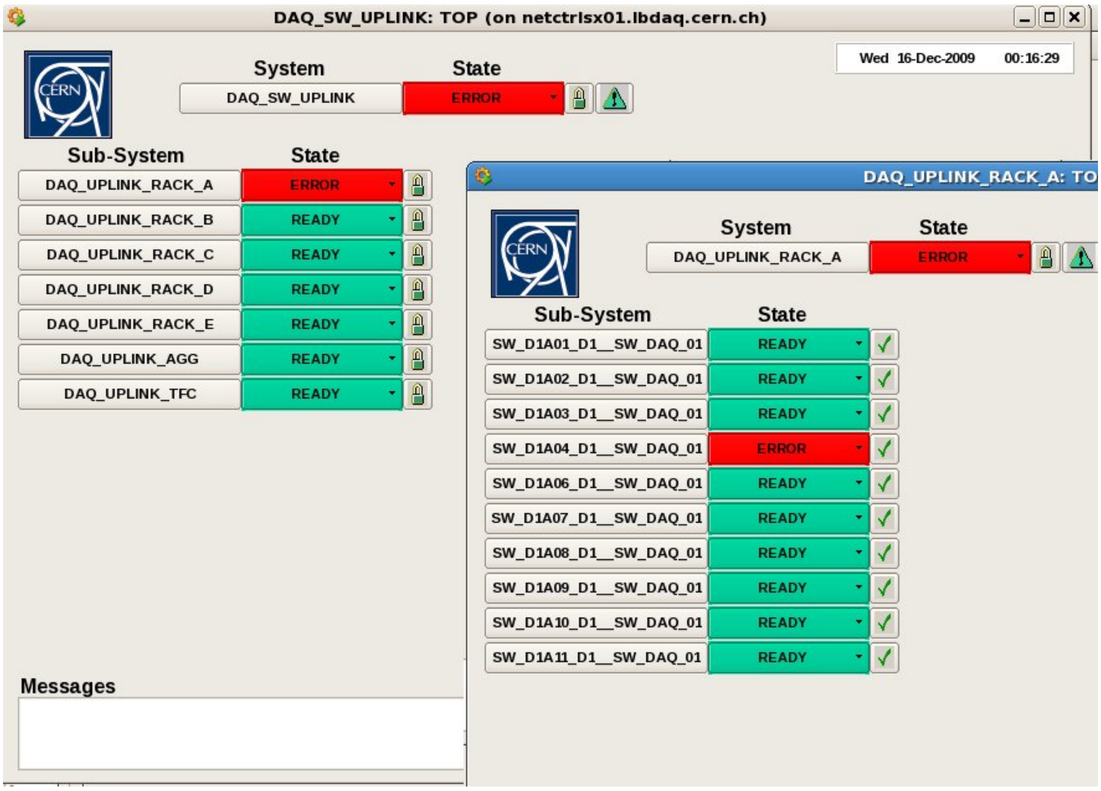
Messages

Figure 3.21: FSM of the DAQ network

The FSM provides a global overview of the system and an intuitive user interface for experiment operators. User can easily isolate the problem by tracing down to the bottom DU. For example, SW_UPLINK is in error state, the root cause can be easily found out by investigating into the tree as shown in Fig. 3.22.

3.4 Summary

The LHCb Online network is a large scale LAN, a sophisticated monitoring at all levels is essential for its successful operation. The custom network monitoring system has been implemented using the same SCADA tool PVSS and the



DAQ_SW_UPLINK: TOP (on netctrlsx01.lbdaq.cern.ch)

Wed 16-Dec-2009 00:16:29

System	State
DAQ_SW_UPLINK	ERROR

Sub-System	State
DAQ_UPLINK_RACK_A	ERROR
DAQ_UPLINK_RACK_B	READY
DAQ_UPLINK_RACK_C	READY
DAQ_UPLINK_RACK_D	READY
DAQ_UPLINK_RACK_E	READY
DAQ_UPLINK_AGG	READY
DAQ_UPLINK_TFC	READY

Messages

DAQ_UPLINK_RACK_A: TO

System	State
DAQ_UPLINK_RACK_A	ERROR

Sub-System	State
SW_D1A01_D1__SW_DAQ_01	READY
SW_D1A02_D1__SW_DAQ_01	READY
SW_D1A03_D1__SW_DAQ_01	READY
SW_D1A04_D1__SW_DAQ_01	ERROR
SW_D1A06_D1__SW_DAQ_01	READY
SW_D1A07_D1__SW_DAQ_01	READY
SW_D1A08_D1__SW_DAQ_01	READY
SW_D1A09_D1__SW_DAQ_01	READY
SW_D1A10_D1__SW_DAQ_01	READY
SW_D1A11_D1__SW_DAQ_01	READY

Figure 3.22: FSM of the uplinks in the DAQ network

framework JCOP as the LHCb ECS. A large amount of information is collected through SNMP, sFlow, syslog and other sources by custom front-end processes efficiently, and consolidated by PVSS. This monitoring system provides to the users the network status: performance, health status, critical conditions, alarms and others.

Chapter 4

Advanced Network Monitoring Based on sFlow

The LHCb Online network needs constant monitoring during its operation. In addition to the monitoring of the bandwidth occupancy and the health status of devices, we have to check the content of the traffic flow at some levels to monitor the network behavior, which can be used to troubleshoot the network problem, detect network anomalies and network intrusions. The traditional technology for network management, like SNMP and RMON, do not allow looking deep into the content of the traffic.

One way to investigate the content of packets is to capture the traffics and analyze them. Modern switches provide a feature called "port-mirroring" to "copy" all traffic from the source port to the destination port, which is normally attached with a PC with a software sniffer[1] (like tcpdump [74], wireshark [75]). This is quite useful to debug a known problem related to a specific switch port, but it is impractical even impossible to examine in this way every packet of the whole network, especially high speed network due to processing and storage constraints. A solution is to sample the traffic stream and examine only a small subset from all the packets using statistics method, this make it possible to monitor the behavior of the whole network.

[1] Sniffer (also known as a network analyzer, protocol analyzer or packet analyzer), is a computer software or computer hardware that can intercept and log traffic passing over a digital network or part of a network

In this chapter, we will first introduce the packet sampling theory and one of the technologies, namely sFlow, then we will describe in the following sections the applications of sFlow in our network, like network trouble-shooting, monitoring of DAQ, misbehavior and security monitoring in control network.

4.1 Packet Sampling and sFlow

As introduced, packet sampling is the most important technology for traffic measurement and monitoring in today's high speed network. There are two major industrial protocols of packet sampling: sFlow and Netflow[1] [76]. In the LHCb Online network, the switches support sFlow only, so all the discussion will focus on sFlow. sFlow employs a "1 out of N" static sampling method to reduce overhead in flow statistics collection. The foremost and fundamental concern regarding packet sampling is its accuracy. Because the network traffic fluctuates dynamically and unpredictably, inaccurate packet sampling may lead to wrong decisions by network operators. Another important question is: how many packets need to be sampled in order to produce traffic measurements with a pre-specified error bound? We explain the theory behind packet sampling and answer the questions in this section.

Suppose during a given time period, there are a total number of packets (N) and a total number of samples (n) of which a certain number (c) belongs to a particular class (S) of interest, then the number of packets in the class S is (N_S), the main notations are summarized in Table 4.1. The estimated value ($\hat{N_S}$) is thus given by:

$$\hat{N_S} = N \cdot \frac{c}{n} \tag{4.1}$$

This has been demonstrated in [77, 78, 79].

Considering the number of interesting samples c, Jedwab et. al. have demonstrated in [78] that c has a hypergeometric distribution[2] [80], the probability of c

[1] NetFlow is a network protocol developed by Cisco Systems to run on Cisco IOS-enabled equipment for collecting IP traffic information.

[2] In probability theory and statistics, the hypergeometric distribution is a discrete probability distribution that describes the number of successes in a sequence of n draws from a finite population without replacement.

Symbol	Description	Comments
N	Total number of packets during the given period	Measured
n	Number of sampled packets	Measured
c	Number of sampled packets belonging to class S	Measured
N_S	Total number of packets which belong to class S	Unknown

having a certain value is:

$$P_{hyp} = \frac{\binom{N_S}{c} \cdot \binom{N-N_S}{n-c}}{\binom{N}{n}} \tag{4.2}$$

If N is large, this approximates to the binomial distribution[1] [80]:

$$P_{bin} \approx \binom{N}{c} \cdot p^c (1-p)^{n-c} \tag{4.3}$$

where $p = N_S/N$, is the population proportion of the interesting packets which is unknown in our problem. Assume the number of samples n is large, according to the Central Limit Theorem [80], c approximates to a Normal distribution [80] with mean and variance as below:

$$\mu_c = np \tag{4.4}$$

$$\sigma_c^2 = np(1-p) \tag{4.5}$$

Using this approximation, the maximum likelihood estimate of the population proportion p is:

$$\hat{p} = \frac{c}{n} \tag{4.6}$$

Thus the estimate for the total number of packets in class S is:

$$\hat{N_S} = \hat{p} \cdot N = \frac{c}{n} \cdot N \tag{4.7}$$

[1] The binomial distribution is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each experiment yields success with probability p

Therefore, the approximate $100(1-\alpha)\%$ confidence interval for N is $[\hat{N}_S - Z_\alpha\hat{\sigma}, \hat{N}_S + Z_\alpha\hat{\sigma}]$, where $1-\alpha/2 = \Phi(Z_\alpha)$ (Φ is the distribution function of a standard Normal), and

$$\hat{\sigma}^2 = \frac{\hat{p}(1-\hat{p})}{n-1} = \frac{c(1-\frac{c}{n})}{n(n-1)} \tag{4.8}$$

is an unbiased estimate of σ^2.

Take 95% confidence interval as example ($[\hat{N}_S - 1.96\hat{\sigma}, \hat{N}_S + 1.96\hat{\sigma}]$), the relative error as a percentage is :

$$error(\%) = 100 \cdot \frac{1.96\hat{\sigma}}{\hat{N}_S} = 196\sqrt{(\frac{1}{c}-\frac{1}{n})(\frac{n}{n-1})} \tag{4.9}$$

As assumed, n is large, so the relative error can be further approximated:

$$error(\%) <= 196\sqrt{\frac{1}{c}} \tag{4.10}$$

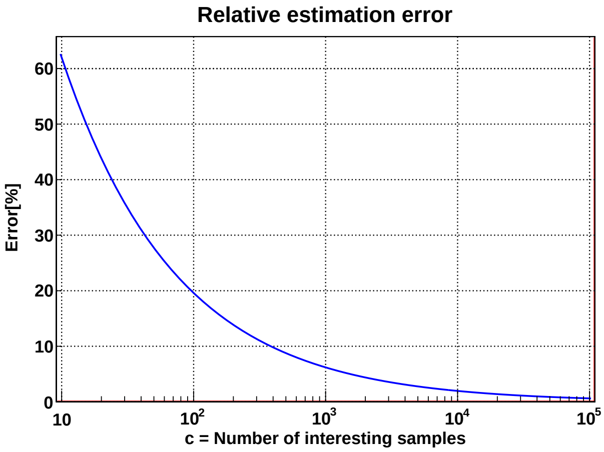


Figure 4.1: Relative estimation error

Fig. 4.1 shows the plot of the relative estimation error. As shown in the plot, the accuracy of the measurement increases along with the number of interesting samples rises. Even though packet sampling does not provide a 100% accurate result, it does provide a result with quantifiable accuracy if the number of interesting sampled packets is large enough. Furthermore, the applications of packet sampling in our system are network trouble shooting, anomaly detection and security monitoring, so we do not need to monitor the traffic flow accurately. The accuracy is not that critical as long as we can get enough samples.

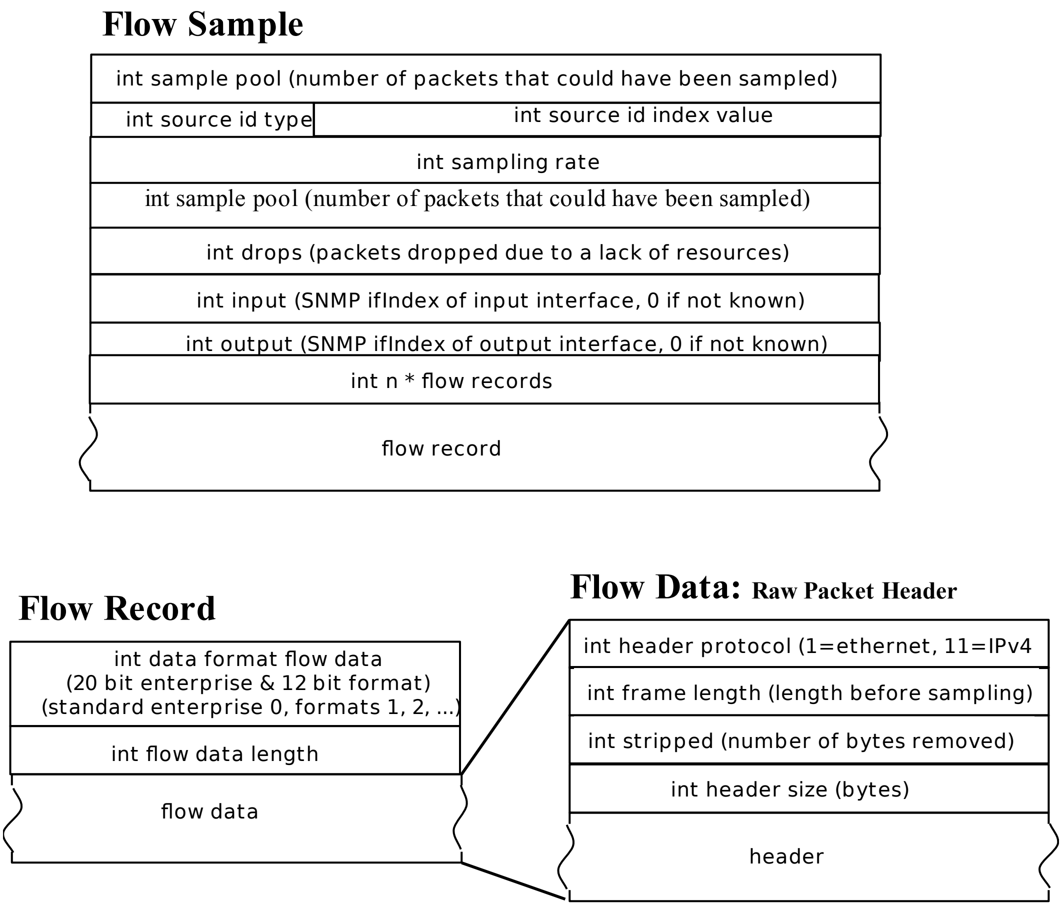

Flow Sample
int sample pool (number of packets that could have been sampled)
int source id type
int source id index value
int sampling rate
int sample pool (number of packets that could have been sampled)
int drops (packets dropped due to a lack of resources)
int input (SNMP ifIndex of input interface, 0 if not known)
int output (SNMP ifIndex of output interface, 0 if not known)
int n * flow records
flow record
Flow Record
int data format flow data
(20 bit enterprise & 12 bit format)
(standard enterprise 0, formats 1, 2, ...)
int flow data length
flow data
Flow Data: Raw Packet Header
int header protocol (1=ethernet, 11=IPv4
int frame length (length before sampling)
int stripped (number of bytes removed)
int header size (bytes)
header

Figure 4.2: Packet format of flow samples

As mentioned in Section 3.2.2.2, there are two kinds of samples in sFlow, the flow sample and the counter sample. The counter sample has already been introduced, here we are going to introduce the flow sample. The flow sample is sampled based on the rate (1 out of N) configured in the sFlow agent and sent to a collector server. As shown in Fig. 4.2, sFlow packets of flow samples contain at least one flow sample at each packet, and each flow sample contains the information of input port, output port and a flow record. Depends on the traffic type and the configuration in the sFlow agent, the record could contain different kinds of flow data, e.g. raw packet header, Ethernet frame data, IPV4 data, IPV6 data, extended switch data and others. In our applications, we use the flow data "raw packet header". The "raw packet header" contains the packet header and a few bytes copied from the payload. The header for an IP packets contains:

- Layer-2 protocol information: source and destination MAC addresses, VLAN tags
- Internet Protocol (IP) headers: source and destination IP addresses, IP protocol, time to live (TTL)[1], IP options and others
- TCP/UDP[2] header: source and destination TCP/UDP port numbers

we can use these information to study the flows and network behaviors.

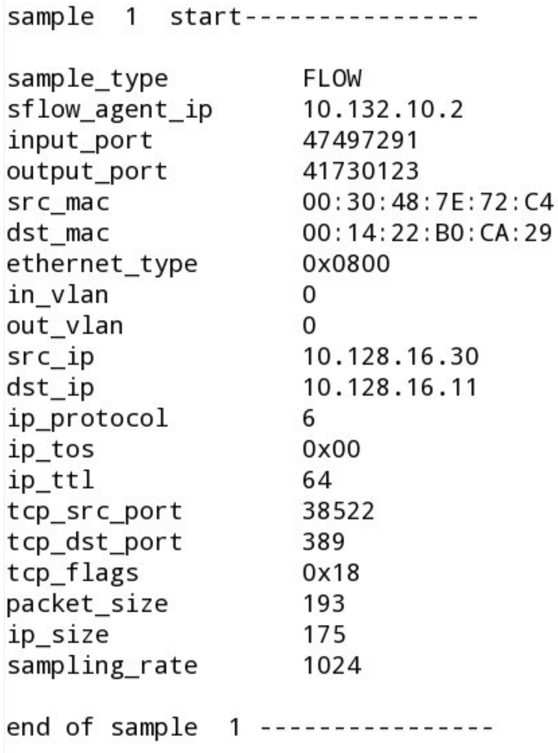
4.2 Network debug tool based on sFlow

During the DAQ commissioning, the sub-detectors sometimes found their data not arriving at the HLT farm for a number of reasons. To find out the cause, we need to look into the packets. The traditional way is to copy the traffic to a sniffer using port mirroring and analyze the traffic, but it is difficult to determine the originating port and there is no way to have this done without the assist from network administrators. Furthermore this can only be done on one input port at a time, not for the entire sub-detector at once. We found the problems were typically caused by wrong configuration of the TELL1 (sender) boards:
- wrong destination IP address: packets will never arrive at the right HLT farm.
- wrong destination MAC address: the router will not accept frames not addressed to itself.

[1] TTL is a value in an IP packet that tells a network router whether or not the packet has been in the network too long and should be discarded.

[2] TCP is a transport layer protocol with re-transmission mechanism to guarantee delivery, while UDP is a connectionless transport layer protocol without guaranteed delivery

- wrong source IP address: packets are denied by access control list (ACL)[1].
- TTL is zero: the router will not forward dead packets.



```
sample  1  start----------------

sample_type          FLOW
sflow_agent_ip       10.132.10.2
input_port           47497291
output_port          41730123
src_mac              00:30:48:7E:72:C4
dst_mac              00:14:22:B0:CA:29
ethernet_type        0x0800
in_vlan              0
out_vlan             0
src_ip               10.128.16.30
dst_ip               10.128.16.11
ip_protocol          6
ip_tos               0x00
ip_ttl               64
tcp_src_port         38522
tcp_dst_port         389
tcp_flags            0x18
packet_size          193
ip_size              175
sampling_rate        1024

end of sample  1 ----------------
```

Figure 4.3: Output of “sFlowSampler”

To ease the troubleshooting, we have developed a tool (call “sFlowSampler”) based on sFlow to investigate a subset of traffic. We use the sflowtool provided by InMon Corp [60] to collect sFlow data. sFlowtool can generate either a simple-to-parse tagged-ASCII output, or binary output in tcpdump format. A Python[2] script has been implemented to parse and display the interesting samples as shown in Fig. 4.3. The output of “sFlowSampler” shows clearly the header of the packet so that we can check if anything is wrong within the packet, otherwise we will further trace the root cause. With this tool, there is no need to configure “port

[1] An ACL is essentially a filter containing some criteria to match (examine IP, TCP, or UDP packets) and an action to take (permit or deny).

[2] Python is a general-purpose high-level programming language.

mirroring" and use a complicated packet analyzer each time, which allow users to investigate the traffic without intervention of network administrators. According to our experiences, most of the problems were caused by misconfiguration on the users side and this tool greatly assisted the troubleshooting.

4.3 DAQ monitoring using sFlow

Similar to "sFlowSampler", we implemented a tool called "daqprobe" to monitor the anomaly of DAQ traffics during the operation of the experiment. If any data source assembles IP packets with a wrong configuration, the problem can be found out quickly, so that DAQ experts fix it by themselves.

The traffic in the LHCb DAQ network is quite simple:

- There are only two kinds of traffic: MEP and MEP request. MEPs are the packets sent from TELL1 boards to HLT farms, and they use the IP protocol 242. While MEP requests are sent from HLT farms to TFC/ODINs, and they use the IP protocol 253.

- There are strict rules for IP addressing:
  - HLT farms use the IP pool 192.168.0.0/16
  - TELL1 boards of each sub-detector belong to the same subnet[1] 192.169.xx.0/24 (xx is the number assigned to each sub-detector), and have the MAC address 00:cc:bb:xx:yy:0z (*yy* is the board number in the sub-detector, and *z* is the port number in this TELL1 board and could range from 0 to 3.)

Based on the features of DAQ traffic and our experiences, we have made a few strict rules for monitoring:

1. All traffic are only between the subnets: 192.168.0.0/16 and 192.169.0.0/16.

2. TTL must be greater than 1

[1] A subnet (short for "subnetwork") is an identifiably separate part of a network. A subnet may represent all the machines at one geographic location, or on the same local area network (LAN). In IP networks, computers in the same subnet have a common, designated IP address routing prefix.

3. The traffic from 192.169.0.0/16 to 192.168.0.0/16 are MEP packets, the protocol number is 242, and the destination MAC address is the MAC address of the core router.

4. The traffic from 192.168.0.0/16 to 192.169.5.0/24 are MEP requests, the protocol number is 253.

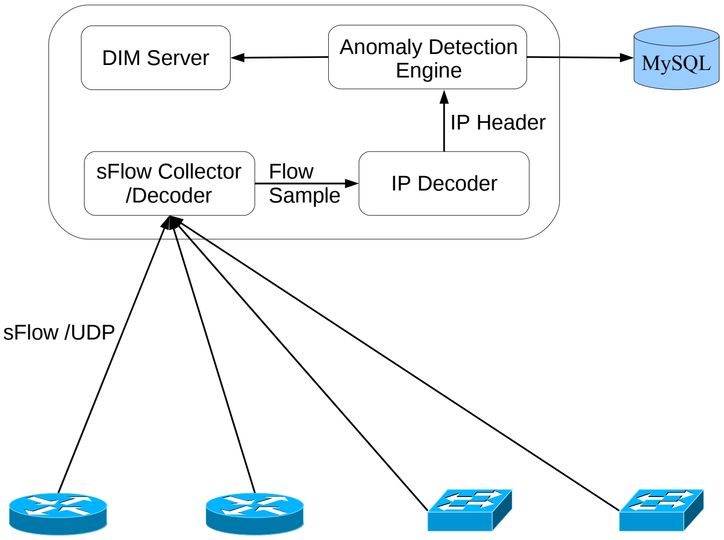


Figure 4.4: Architecture of daqprobe

The architecture of “daqprobe” is shown in Fig. 4.4. The sFlow collector receives sFlow packets from the sFlow agents and decodes them to get the IP headers. The unwrapped IP headers are further decoded to get the IP information which will be sent to the anomaly detection engine. Then the anomaly detection engine will check the IP information with the rules introduced above. If any of the rules is violated, a message will be sent through DIM server to PVSS to inform the operator. In the meantime, the packet will be logged into MySQL [81] database, which is very useful for maintaining history data, generating reports and analyzing information. The database schema is shown in Fig. 4.5, and the tables are:

- Signature: a list of alarm/signature and priorities

- Event: the time stamp, alarm, data packet and the switch interface.
- MAChdr: the Layer-2 frame information
- IPhdr: all the IP information
- Switch_if: a list of the SNMP index of interfaces

With all these information in database, we can display the protocol analysis through PVSS panels.

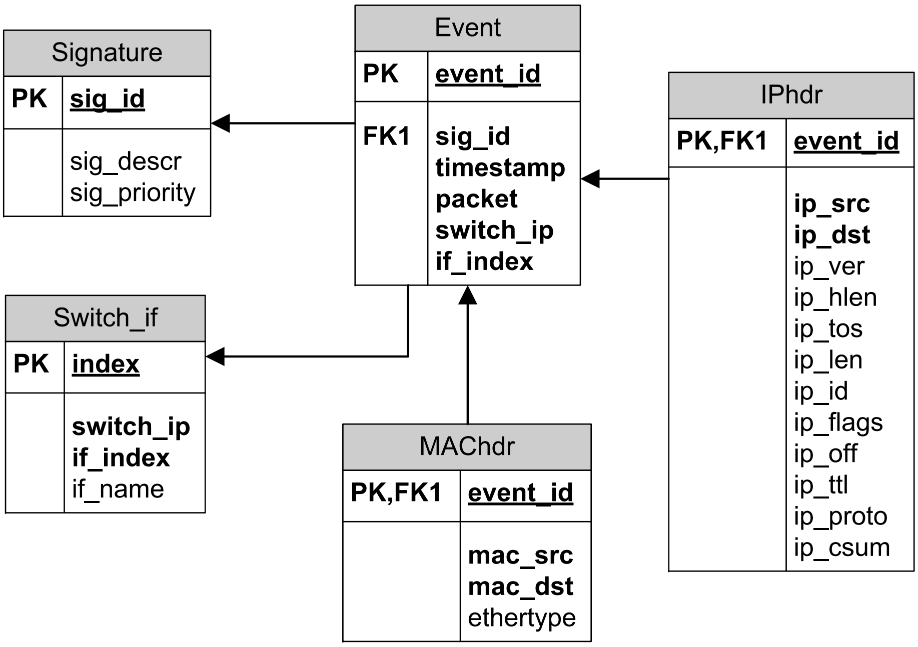


Figure 4.5: Database schema of "daqprobe"

4.4 Monitor security threats and anomalies using sFlow and snort

Nowadays computer networks are becoming bigger and more complex. And networks are being faced with increasing security threats and malicious network service misuse, which could cause serious disruption to network services. The LHCb Online network is a private network, except for the web services, users can only access the LHCb Online cluster from CERN network through a few gateways where firewalls are deployed. Even though it is less exposed directly to Internet, the network security is still a critical issue. The threats may originate externally or internally, and may occur at any time. And studies have shown that insiders do the most damage, normally attackers first break into a computer in your network or crack an account, then launch the attacks to your network. To detect and respond promptly to this situation, an intrusion detection system (IDS) has been deployed to monitor the internal and the external security threats. The external monitoring are only for the gateways and the web server, while the internal monitoring requires much broader coverage.

In order to detect network attacks promptly, an intrusion detection system has to meet certain requirements [82, 83]:

- Network-wide, continuous surveillance
- Reliable, timely availability of data, especially during network overload, which is common during an attack
- Interpretation of traffic patterns
- Alerts to violations or threats
- Provision of sufficient information to take action

Initially packet monitoring was performed by specially designed probes installed on the network. These probes copied the entire contents of every packet for further analysis. This approach allowed relatively easy monitoring of whole network traffic on shared medium networks like the early Ethernet. However it is impossible to log and process the entire contents on today's high-speed and large-scale networks. Packet sampling technology like sFlow is another way to monitor the network. Many studies have been carried out for anomaly detection based on packet sampling [84, 85, 86]. sFlow captured only the IP header and a very limited number of bytes from the payload, but the IP header has a great interest in network monitoring and can be used to detect most of the network attacks.

Our network intrusion detection system is based on sFlow which collect the packet samples and an open source IDS called Snort [87], which will be introduced in the following sections.

4.4.1 Introduction to Snort

Snort is an open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. It is a cross-platform, lightweight tool that can be deployed to monitor small TCP/IP networks and detect a wide variety of suspicious network traffic as well as outright attacks.

Snort can be configured to run in four modes [88]:

- Sniffer mode, which simply reads the packets off of the network and displays them.

- Packet Logger mode, which logs the packets to disk.

- Intrusion Detection System (IDS) mode, which allows Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees.

- Inline mode, which obtains packets from iptables[1] instead of from libpcap and then causes iptables to drop or pass packets based on Snort rules that use inline-specific rule types.

Snort consists of multiple logical components: Sniffer, Preprocessor, Detection Engine and Alerts/Logging. These components work together to detect particular attacks and to generate output in a required format from the detection system. Fig. 4.6 shows the architecture of Snort and how its components are arranged:

- **Sniffer**: which captures and decodes the packets from the network, and then send them to preprocessors or the detection engine.

[1] Iptables is used to set up, maintain, and inspect the tables of IP firewall in the Linux kernel. Each table contains a number of chains, which are a list of rules that match a set of packets. Each rule specifies what to do with a matched packet.

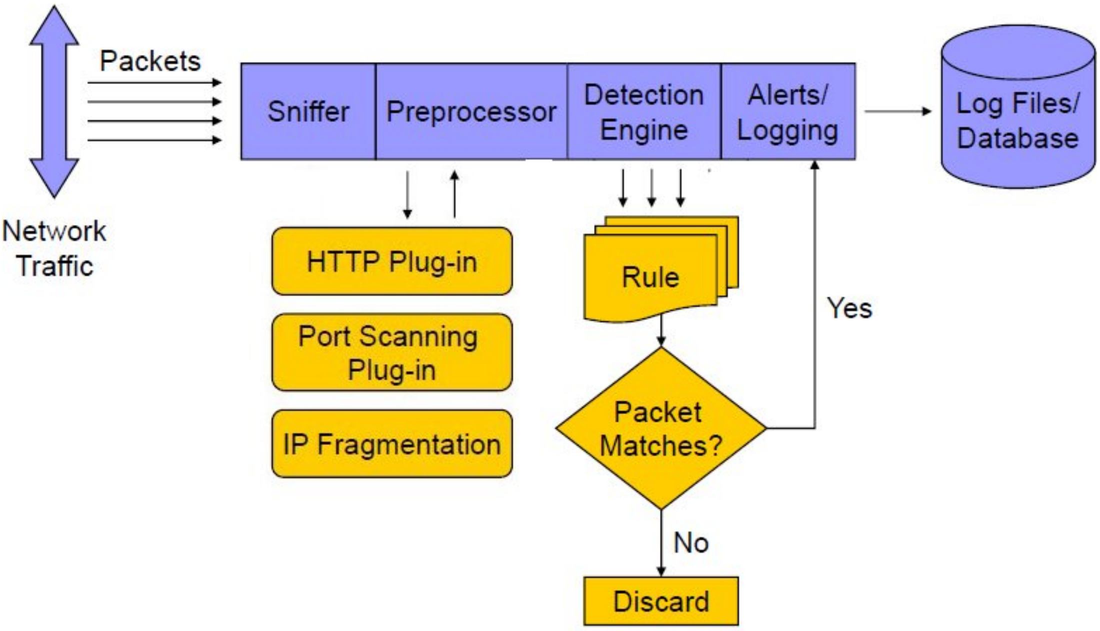


Figure 4.6: Architecture of Snort

- **Preprocessor**: which allows users and programmers to drop modular plugins into Snort to extend the functionality fairly easily. Hackers use different techniques to fool an IDS in different ways, for example hackers can fragment the packet or insert in the web Uniform Resource Identifier (URI) hexadecimal characters or Unicode characters. Before sending data packets to the detection engine, preprocessors can defragment packets, decode HTTP URI, re-assemble TCP streams and so on. These functions are a very important part of the intrusion detection system.

- **Detection Engine**: which is the most important part of Snort. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. If a packet matches any rule, the packet is sent to the alert/logging processor, otherwise the packet is dropped.

- **Alerts/Logging**: which logs the data packet in a log file or database and triggers an alert. Alerts may be sent in the several formats: syslog, SNMP

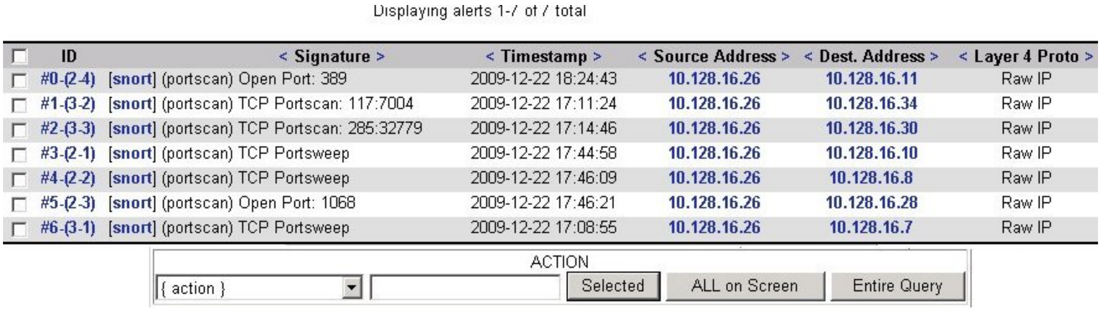
Trap, Windows Server Message Block (SMB) and so on.

Snort rules are used to detect a wide variety of hostile or merely suspicious network traffic. The rules describe a state of the network and list an action to perform if the state is true. The rules can be obtained in the Snort homepage `www.snort.org`, but we still need to customize or develop some rules for our network. Snort rules are divided into two logical parts: the rule header and the rule options. The rule header contains the rules action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken.

4.4.2 IDS based on sFlow and Snort

In the system, sflowtool is used to collect sFlow data packets, the packets are output to the standard output (STDOUT) in binary tcpdump format. While Snort listens and reads the tcpdump-formatted file from the standard input (STDIN), it is the same as listening to network traffic on a normal Ethernet interface. The command is :

```
sflowtool -t | snort -Afull -r - -c snort.conf
```

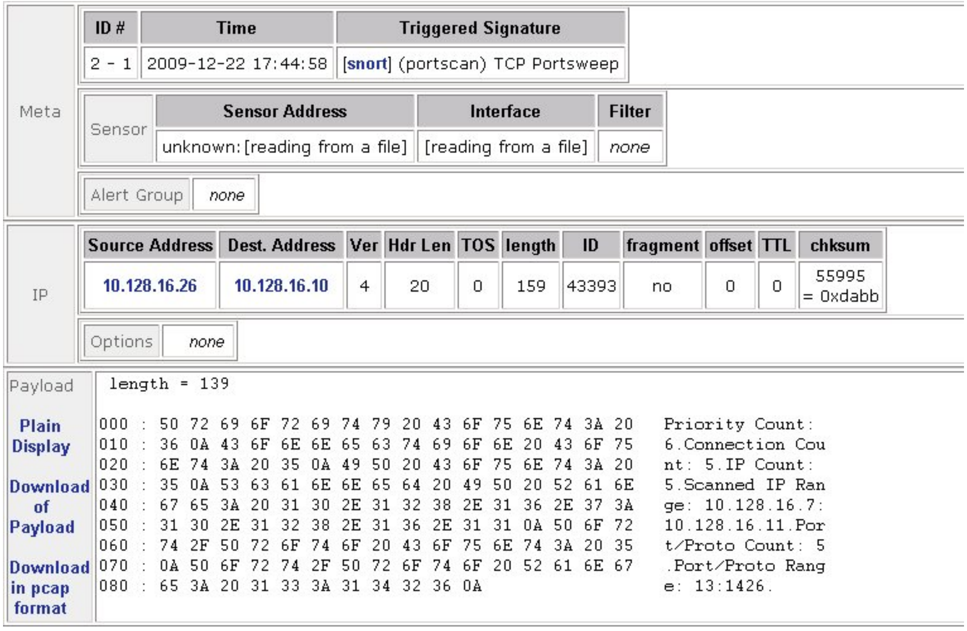


Displaying alerts 1-7 of 7 total

□	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
□	#0-(2-4)	[snort] (portscan) Open Port: 389	2009-12-22 18:24:43	10.128.16.26	10.128.16.11	Raw IP
□	#1-(3-2)	[snort] (portscan) TCP Portscan: 117:7004	2009-12-22 17:11:24	10.128.16.26	10.128.16.34	Raw IP
□	#2-(3-3)	[snort] (portscan) TCP Portscan: 285:32779	2009-12-22 17:14:46	10.128.16.26	10.128.16.30	Raw IP
□	#3-(2-1)	[snort] (portscan) TCP Portsweep	2009-12-22 17:44:58	10.128.16.26	10.128.16.10	Raw IP
□	#4-(2-2)	[snort] (portscan) TCP Portsweep	2009-12-22 17:46:09	10.128.16.26	10.128.16.8	Raw IP
□	#5-(2-3)	[snort] (portscan) Open Port: 1068	2009-12-22 17:46:21	10.128.16.26	10.128.16.28	Raw IP
□	#6-(3-1)	[snort] (portscan) TCP Portsweep	2009-12-22 17:08:55	10.128.16.26	10.128.16.7	Raw IP

Figure 4.7: Snort alert list

We deployed the Basic Analysis and Security Engine (BASE) to display the alert in Snort. BASE is a web interface to perform analysis of intrusions that snort has detected on your network. It displays the list of the alerts (see Fig. 4.7), and the detail of the alert and the data packet as shown in Fig. 4.8.



Meta

ID #	Time	Triggered Signature
2 - 1	2009-12-22 17:44:58	[snort] (portscan) TCP Portsweep

Sensor	Sensor Address	Interface	Filter
	unknown: [reading from a file]	[reading from a file]	*none*

Alert Group	*none*

IP

Source Address	Dest. Address	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
10.128.16.26	10.128.16.10	4	20	0	159	43393	no	0	0	55995 = 0xdabb

Options	*none*

Payload

Plain Display

Download of Payload

Download in pcap format

```
length = 139

000 : 50 72 69 6F 72 69 74 79 20 43 6F 75 6E 74 3A 20   Priority Count:
010 : 36 0A 43 6F 6E 6E 65 63 74 69 6F 6E 20 43 6F 75   6.Connection Cou
020 : 6E 74 3A 20 35 0A 49 50 20 43 6F 75 6E 74 3A 20   nt: 5.IP Count:
030 : 35 0A 53 63 61 6E 6E 65 64 20 49 50 20 52 61 6E   5.Scanned IP Ran
040 : 67 65 3A 20 31 30 2E 31 32 38 2E 31 36 2E 37 3A   ge: 10.128.16.7:
050 : 31 30 2E 31 32 38 2E 31 36 2E 31 31 0A 50 6F 72   10.128.16.11.Por
060 : 74 2F 50 72 6F 74 6F 20 43 6F 75 6E 74 3A 20 35   t/Proto Count: 5
070 : 0A 50 6F 72 74 2F 50 72 6F 74 6F 20 52 61 6E 67   .Port/Proto Rang
080 : 65 3A 20 31 33 3A 31 34 32 36 0A                  e: 13:1426.
```

Figure 4.8: Snort alert detail

4.4.3 Discussions on the accuracy of anomaly detection

A monitoring system based on packet sampling will not provide absolutely accurate data by its nature. So far, packet sampling have been mainly used for network traffic accounting and billing. As discussed in Section 4.1 and proved in many research, the desired accuracy of various traffic estimations can be achieved (see the survey in [89]). The accuracy of packet sampling depends on the sampling rate, application and periodicity of measured metric. The result could be also useful for intrusion detection technology when estimated thresholds are needed. Periodic events are very likely to be detected by packet sampling.

There is a wide range of common network anomalies that only require a single sample, for instance protocol violations: a connection to an IP address from a blacklist. If a packet sample matches the rule that identifies one of this kind of anomalies, then it is likely that the packet was with malicious intent. It is enough to detect the existence of certain types of traffic for this kind of anomalies. Most of security threats generate enough traffic to be seen by sampling and hence the existence will be reported. This kind of detection provides an accuracy of 100%,

There are some network anomalies which cannot be detected by observing only a single sample, this normally need a threshold between normal and abnormal. For this case, we need to study a large amount of samples in order to get a reasonably high accuracy of the threshold. For instance, the first phase of a network attack (as known as reconnaissance) is to determine what types of network protocols or services a host supports by a portscan. The attacking host has no prior knowledge of what protocols or services are supported by the intended target, most queries sent by the attacker will be negative (the service ports are closed). In the legitimate network communications, negative responses from hosts are rare. Multiple negative responses within a given amount of time, indicates a portscan/portsweep which could be one to one, one to many, or many to one scan. To detect this kind of threats, we need to establish a threshold. Because the network traffic data is sampled, the threshold must be much lower than the normal traffic depends on the sampling rate.

There are still some network anomalies, for instance the application attack, which need to examine the application protocol or the content of the packet. Since the sampled packets carry very few bytes of the payload, this kind of anomalies can not be detected by packet sampling based IDS.

In order to test the IDS based on sFlow and Snort, we have performed portscans with nmap[1], and performed SYN-flood attack[2] with hping[3], the IDS detected the attacks without error. In general, the anomaly detection based on sFlow packet sampling is reliable for most of the anomalies if the rules are created properly, but it can not detect those threats attack in the application layer.

[1] Nmap ("Network Mapper") is a free and open source utility for network exploration or security auditing

[2] SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system

[3] Hping is one of the de facto tools for security auditing and testing of firewalls and networks

4.5 Summary

Nowadays packet sampling technologies are widely used to characterize network traffic. It provides a quantifiable accuracy, even not 100%, but it is sufficient for some applications. In this chapter, we introduced the basis of packet sampling theory and one of the mechanism sFlow. A few applications of sFlow have been demonstrated, such as the network trouble-shooting, the DAQ monitoring and the network anomaly detection.

Chapter 5

Evolution of the LHCb DAQ Network

5.1 LHC upgrade

The peak luminosity of LHC is designed to be $10^{34} cm^{-2} \cdot s^{-1}$. The performance of LHC is planned to evolve to the design peak luminosity around the year 2013. This will open a new field of physics, allowing the study of rare physics with the highest accuracy ever. To establish TeV scale physics firmly after possible discoveries, it is important to make precise measurements of the discovered new phenomena. Thus, it is planned to upgrade LHC to super LHC (sLHC). The peak luminosity will be increased to $10^{35} cm^{-2} \cdot s^{-1}$, which is ten times higher than the original LHC design. LHC luminosity upgrade is expected to go in two stages [90]:

- Phase I [91]: obtains the luminosity of 2 to 3 $\times 10^{34} cm^{-2} \cdot s^{-1}$, mostly based on the current infrastructure.
- Phase II [92]: reaches the ultimate luminosity of $10^{35} cm^{-2} \cdot s^{-1}$

The time frame is not yet established, but according to the presentation of Lyn Evans in SLHC-PP annual meeting "SLHC, the high-luminosity upgrade" in February 2009 [93], Phase I will be implemented in 2013-2014, Phase II will be in 2016-2017 and eventually deliver the ultimate luminosity at about 2020. The plan of LHC is shown in Fig. 5.1 (extracted from [93]).

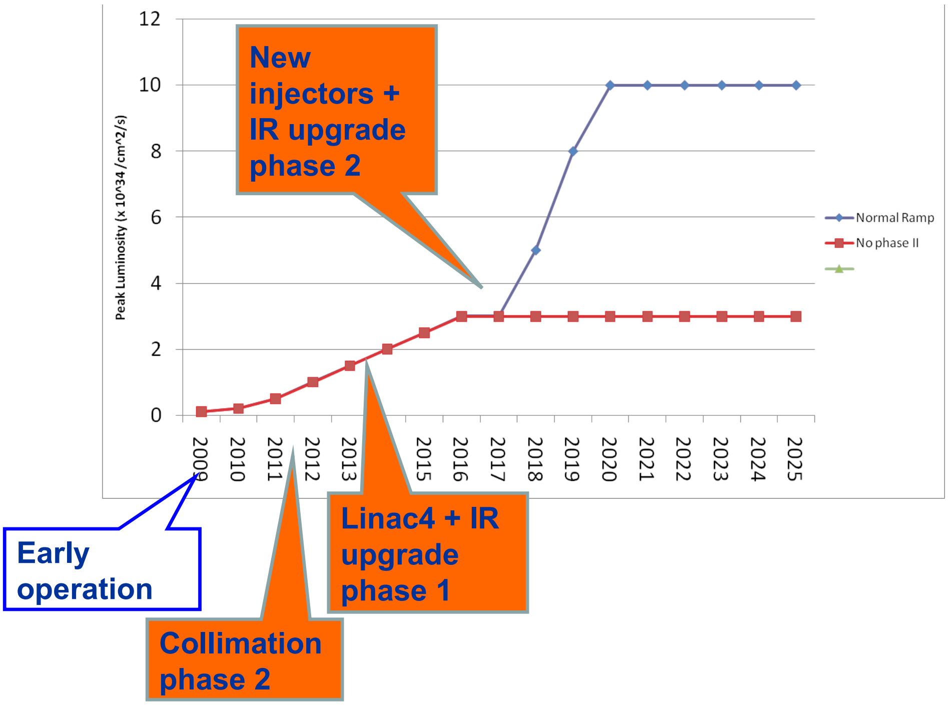


Figure 5.1: Estimated time frame for LHC luminosity upgrade

5.2 LHCb upgrade

LHCb will start taking physics data in 2010 and be expected to accumulate a data sample of ~10 fb^{-1} during the following five years. The large cross section[1] will allow LHCb to collect a large amount of data samples of B mesons to improve the precision on the CKM angle γ by a factor of five. At that time, the statistical precision on measurements increases very slowly if the LHCb experiment continues operating at current peak luminosity beyond 10 fb^{-1} without an upgrade.

So LHCb has planned an upgrade [94] to enable operation at 10 times of the current design luminosity, i.e. at about $2 \times 10^{33} cm^{-2} \cdot s^{-1}$, to improve the trigger efficiency for hadronic decays by a factor of two and to collect an integrated luminosity of about 100 fb^{-1}. This will increase the data samples by a factor

[1] 500 μb for $b\bar{b}$-quark production in pp collisions at 14 TeV centre-of-mass energy

of 10 and 20 for leptonic and hadronic decay modes respectively. The original upgrade plans were for the general purpose detectors (ATLAS and CMS) [95], however LHCb can be made compatible [96]. The bunch crossing rate at LHCb given by the LHC machine is to be 40.08 MHz, while 2622 out of the theoretically possible 3564 crossings have protons in both bunches. Hence, the maximum rate of crossings with at least one pp interaction is ∼30 MHz [94].

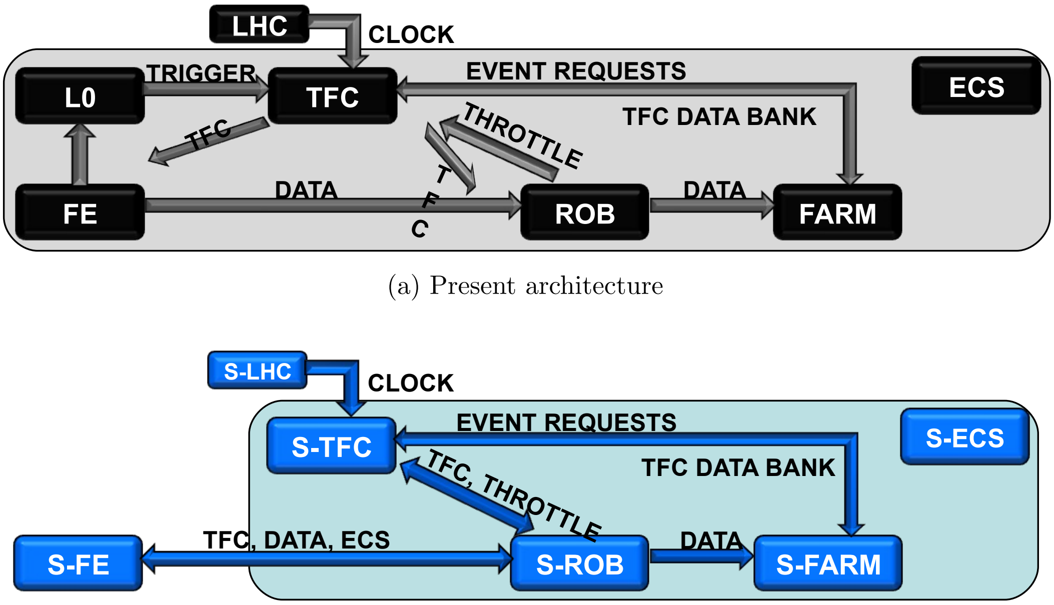
5.2.1 The LHCb trigger upgrade

As introduced in Section 1.3.5, The current LHCb trigger has two levels, L0 and HLT. The L0 trigger rate is maximum at 1 MHz, which is limited by the hardware of the L0 trigger. L0 reconstructs the highest E_T hadron, electron and photon, and the two highest p_T muons. The typical threshold for E_T^{hadron} is $E_T^{hadron} > 3.5$ GeV. Even only for a $E_T^{hadron} > 3.5$ GeV, the hadron trigger (L0-h) rate at $2 \times 10^{32} cm^{-2} \cdot s^{-1}$ would exceed 1 MHz, hence L0 imposes additional cuts on the number of interactions per event and the track multiplicity to reduce the L0-h rate to 700 kHz [94].

The main purpose of the LHCb upgrade is to improve the trigger efficiency for hadronic decays. To efficiently select B decays, the latency required by the algorithms is far longer than what is possible with the present architecture (4 μs). Hence, the LHCb upgrade has opted for a new FEE architecture which requires all data readout at the full 40 MHz rate of the LHC machine. The data should be transmitted to a large Event Filter Farm (EFF), where the trigger algorithm would be executed, just like the present HLT.

5.2.2 Architecture of trigger free DAQ system

Since all data need to be readout at 40 MHz and transported to EFF, the entire FEE and DAQ system must be replaced and all event selection is done in software in a large CPU farm. A new readout architecture [97, 98] is shown in Fig. 5.2b (the current readout architecture is also shown in 5.2a for comparison). In the new architecture, all modules are preceded by “S-” (Super-) which indicates upgraded modules.



(a) Present architecture

(b) Upgrade architecture

Figure 5.2: Readout architecture: new vs old

Comparing to the current architecture, there is no L0 trigger anymore. Instead, a “rate control trigger” [94] is introduced to cope with the overflow caused by:

- A staged DAQ system, which can not yet handle the full rate.
- Unexpectedly high occupancies, which exceed capacity of the readout system.
- Insufficient CPU power in the event-filter farm.

The “rate control trigger” mechanism is likely based on local trigger decisions computed in the Readout Boards (ROB) and also possibly overflow monitoring in the output stage of the Readout Boards.

The push protocol with a passive pull for the data readout will be kept [99]. The farm nodes still declare themselves as ready to receive the next events for processing as the mechanism today. The readout supervisor is still needed to assign an destination IP address for an IP packet. The scheme avoids the risk of sending events to non-functional nodes and provides a level of load balancing as well as a rate control possibly.

Another big change is the TFC links to FEE. The direct TFC links to FEE are eliminated. Instead the new Readout Boards will provide the interface to FEE for TFC, and as well the ECS information for the configuration and monitoring of FEE. This benefits from bidirectional capability of the CERN GigaBit Transceiver (GBT) which will be used for the data transfer between FEE and the readout boards. The synchronous TFC information is thus relayed onto a set of GBT links together with the asynchronous ECS information, which significantly reduces the number of links to the FEE boards.

Fig. 5.3 [100] shows the architecture of the electronics. FEE is required to readout at 40 MHz, although ∼10 MHz of the bunches are actually empty. All sub-detectors need to replace or adapt their FEE to the new 40 MHz read-out scheme, and drive their data over the GBT link to the "New Readout Board" called TELL40. In order to reduce the GBT links, the zero-suppression will be performed in the FEE boards instead of the readout boards. The readout boards will not do a lot of data reduction but rather data formatting, and then send the data to the high-speed readout network [101] via possible 10 Gb/s interfaces. The event size and the event rate are not yet established, but an estimation of 100 kB@30 MHz is adopted in [94], and the number of GBT and 10-Gb/s links required is given in [100]: 9532 GBT links and 3556 10-Gb/s links from the readout boards to the readout network. For the high-speed readout network, currently there are two potential technologies, 10 Gb/s Ethernet (10GbE) or possibly 40/100 Gb/s Ethernet) and 40 Gb/s InfiniBand [102], which will be compared in the next section.

5.3 High speed networking technology

10GbE and InfiniBand are the two most popular technologies for high speed interconnection today. We will introduce both technologies and then give a comparison.

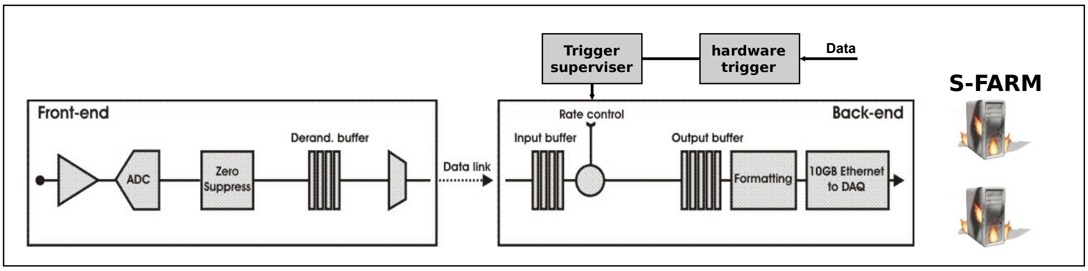


Figure 5.3: Architecture of electronics

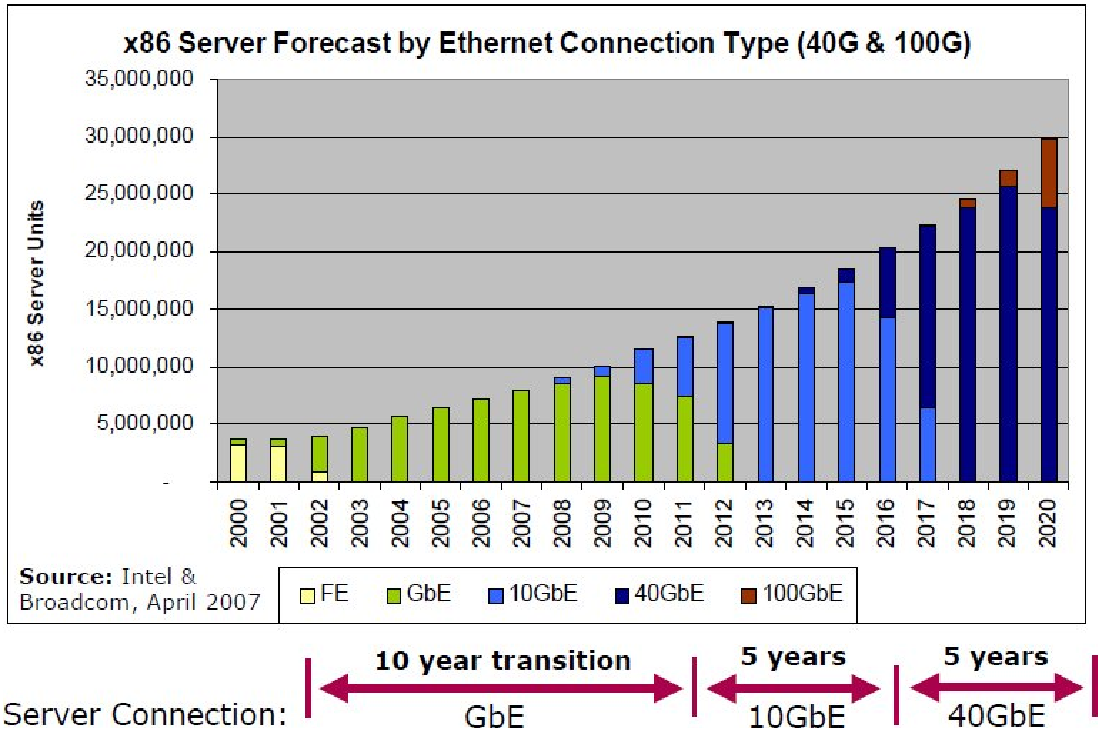
5.3.1 10/40/100 Gb/s Ethernet

The 10GbE standard was first published in 2002 as IEEE Std 802.3ae-2002 and is the fastest of the Ethernet standards so far. In almost every respect, 10GbE is fully compatible with previous versions of Ethernet. It uses the same frame format, Media Access Control (MAC) protocol and frame size, and network managers can use familiar management tools and operational procedures.

Shipments of 10GbE have rapidly increased since it emerged. Today, although most desktops/workstation or servers are still shipped with the 1GbE Networks Interface Card (NIC), 10GbE has gained more and more acceptance as more and more bandwidth intensive applications (such as Web 2.0, Virtualization, High-Performance Computing (HPC) and Network Attached Storage (NAS)) are deployed in enterprises, data centers and service provider networks.

The 10GbE standard encompasses a number of different physical layer (PHY) standards, based on optical fibers or copper cables. Fiber cabling is typically used for long-distance communications and environments that need protection from interference, such as manufacturing areas. Copper cabling possesses the advantage of low cost, easy installation and flexibility. The copper standards for 10GbE include 10GBASE-CX4, SFP+ Direct Attach and 10GBASE-T [103]. 10GBASE-CX4 and SFP+ Direct Attach can reach a distance of 15 m and 10 m respectively, while 10GBASE-T can reach 100 m over cat 6a cables and 55 m over cat 6 and cat 5E cables. 10GBASE-T allows a gradual upgrade from 1000BASE-T using auto-negotiation to select which speed to use.

There are plenty of 10GbE devices available from variant vendors in the market with inexpensive price nowadays. In the past few years, the cost per switch port has dropped from about US$12,000 to as low as US$500 today according to Cisco[1] in [104]. And the price of NIC has dropped to about US$500 today, strong players such as Chelsio, Intel and Broadcom are providing stable, reliable products. Now server vendors are starting to add an Ethernet chip to the motherboard, known as LAN-on-Motherboard (LOM). This will further reduce the cost of 10GbE NIC.



Source: Intel & Broadcom, April 2007

Figure 5.4: X86 server forecast by Ethernet connection type

As for the evolution of 10GbE, the IEEE 802.3ba group was named in July 2007 to study new Ethernet standards with higher speed at 40 Gb/s for local server applications, and 100 Gb/s for internet backbone. According to the

[1] Cisco is the leading network equipment maker in the world

roadmap, the new standard is expected to be ratified by June 2010. Mellanox[1] has released a 40GbE NIC called "ConnectX-2 EN 40G" with a single 10GBASE-CR4 port, which uses standard Quad Small Form-factor Pluggable (QSFP) connectors. And some opto-electronics manufacturers like Finisar, Opnext, Reflexphotonics demonstrated their 40/100G CFP modules at ECOC09[2]. We can expect the market of 40GbE will increase as the demands of bandwidth increase. Robert Hays from Intel and Howard Frasier from Broadcom predicted the market potential of 40GbE (as shown in Fig. 5.4) in IEEE 802.3 Higher Speed Study Group Interim Meeting, April 2007.

5.3.2 InfiniBand

InfiniBand is an industry standard that defines a new high-speed switched fabric subsystem. It is designed to connect processor nodes and I/O nodes to form a system area network. And it is used mainly in HPC and large enterprise server installations. The specification of InfiniBand can be found in the homepage of the InfiniBand Trade Association (IBTA)[105]. This new interconnect method moves away from the local transaction-based I/O model across buses to a remote message-passing model across channels.

The InfiniBand architecture (IBA) provides lower latency and higher bandwidth than traditional networking solutions. InfiniBand provides OS-bypass features such as Remote Direct Memory Access (RDMA) operations, which enables true "zero copy"[3] between systems. RDMA allows remote host to place data directly into the buffer of an application program without any involvement from the operating system. This mechanism dramatically reduces latency and CPU load. The switch chips have a latency of ∼200 nanoseconds (ns).

InfiniBand fabrics use high-speed, bi-directional serial interconnects between devices through either copper cable, optical fiber, or printed circuit on a back-

[1] Mellanox is a leading supplier of end-to-end connectivity solutions for servers and storage. http://www.mellanox.com

[2] ECOC09 stands for European Conference on Optical Communication 2009, which was held in Vienna, Austria.

[3] Zero-copy describes computer operations in which the CPU does not perform the task of copying data from one memory area to another.

	SDR (Gb/s)	DDR (Gb/s)	QDR (Gb/s)
1×	2.5	5	10
4×	10	20	40
12×	30	60	120

Table 5.1: InfiniBand interconnect bandwidth

plane. The bi-directional links contain dedicated send and receive lanes for full duplex operation. The signaling rate is 2.5 Gb/s for single data rate (SDR) in each direction per lane. InfiniBand also supports double (DDR) and quad data rate (QDR) speeds, for 5 Gb/s or 10 Gb/s respectively, at the same data-clock rate. Each InfiniBand link could contain 1×, 4×, or 12× lanes, of which the 4× is the most popular configuration today. But we have to keep in mind that InfiniBand links use 8B/10B encoding (every 10 bits sent carry 8bits of data), which make the useful data transmission rate four-fifths the signal rate. The bandwidth of InfiniBand continues increasing, the roadmap is shown in Fig. 5.5.

IBA is divided into multiple layers where each layer operates independently [106] as TCP/IP. As shown in Fig. 5.6, InfiniBand is divided into the following layers: Physical, Link, Network, Transport and Upper Layers. IBA provides five transport service types:

- Reliable connection (RC) - data transfer between two entities using receive acknowledgment
- Unreliable connection (UC) - same as RC but without acknowledgment
- Reliable datagram (RD) - data transfer using RD channel between RD domains
- Unreliable datagram (UD) - data transfer without acknowledgment
- Raw packets (RP) - transfer of datagram messages that are not interpreted

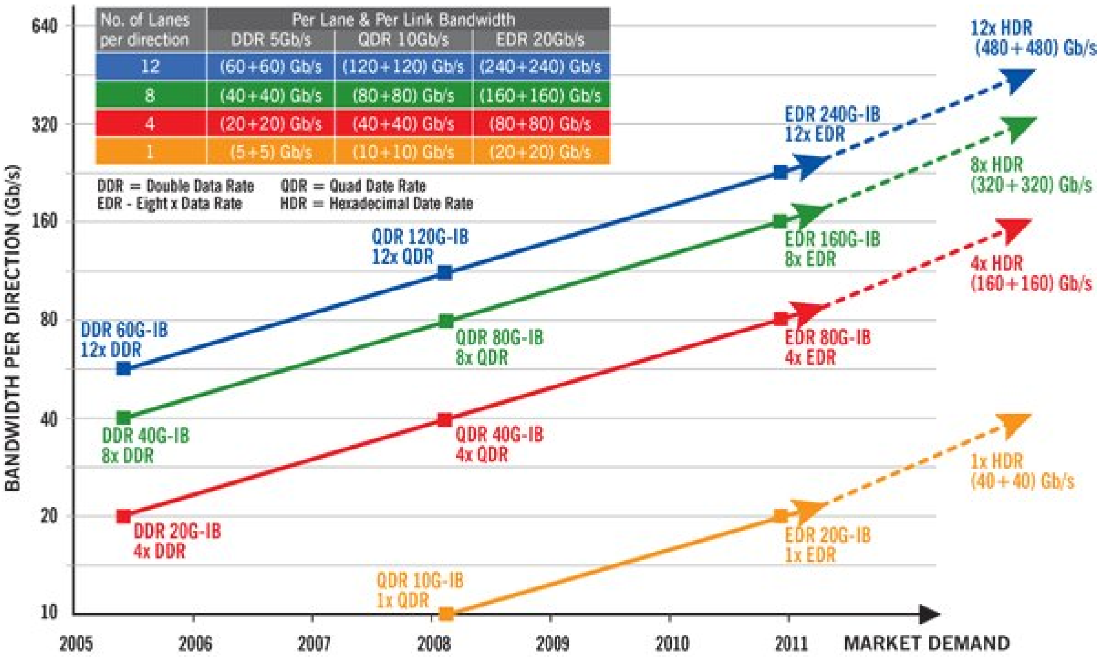


Figure 5.5: InfiniBand roadmap

5.3.3 Comparison of 10/40/100 GbE and InfiniBand

As high speed interconnection technologies, high speed Ethernet and InfiniBand are the most promising candidates, they both have their own advantages in bandwidth intensive applications. Considering the application in LHCb DAQ upgrade, we will compare 10GbE and InfiniBand in the following aspects: performance, market, difficulty of implementing, maturity and the quality of service (QoS) mechanisms.

5.3.3.1 Performance

40G InfiniBand has more than 3 times the throughput of 10GbE and 5-6 times less the latency. There is no doubt that InfiniBand gains much better performance over 10GbE in those applications that need the lowest possible latency or the highest bandwidth. For those applications that are not latency sensitive, both 10GbE and InfiniBand are appropriate solutions. Cisco has demonstrated that the performance of 10GbE was very close to that of 4×DDR (20G) InfiniBand:

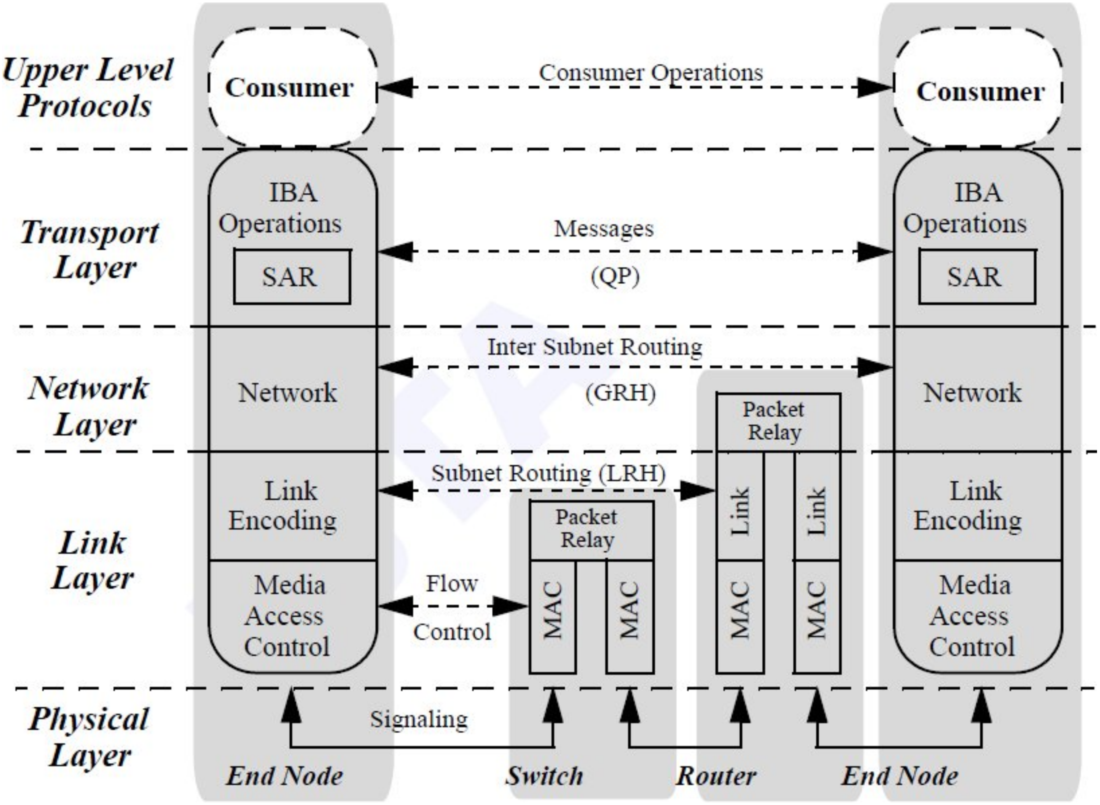


Figure 5.6: InfiniBand architecture layers

less than or equal to 2.5 percent [104]. With the deployment of TOE (TCP/IP Offload Engine) technology in 10GbE NIC, CPU utilization for data movement drops to a very low level which make the host system become extremely efficient. iWARP[1] supports RDMA and OS bypass. Coupled with TOE, it can fully eliminate the host CPU involvement in an Ethernet environment.

With TOE and iWARP, host systems based on Ethernet are almost as efficient as the ones based on InfiniBand, even though the latency is still higher, but our applications are not sensitive to latency.

[1] iWARP is a set of standards enabling RDMA over Ethernet.

5.3.3.2 Market

Ethernet technology is the most deployed technology for high-performance LAN environments. Enterprises around the world have invested cabling, equipment, processes, and training in Ethernet. The cost of 10GbE has dropped significantly drop with the development of 10GbE-based technologies, there is no doubt that 10GbE will gain more and more acceptance. Comparing to Ethernet, shipment of InfiniBand is still very tiny, but InfiniBand is gaining more and more commercial acceptance especially in the HPC area. According to the statistics of top500 supercomputer sites [107], the number of site using InfiniBand for interconnect has increased from 78 to 181 in the past three year (see Fig. 5.7). The competition in HPC will become harsher and harsher.

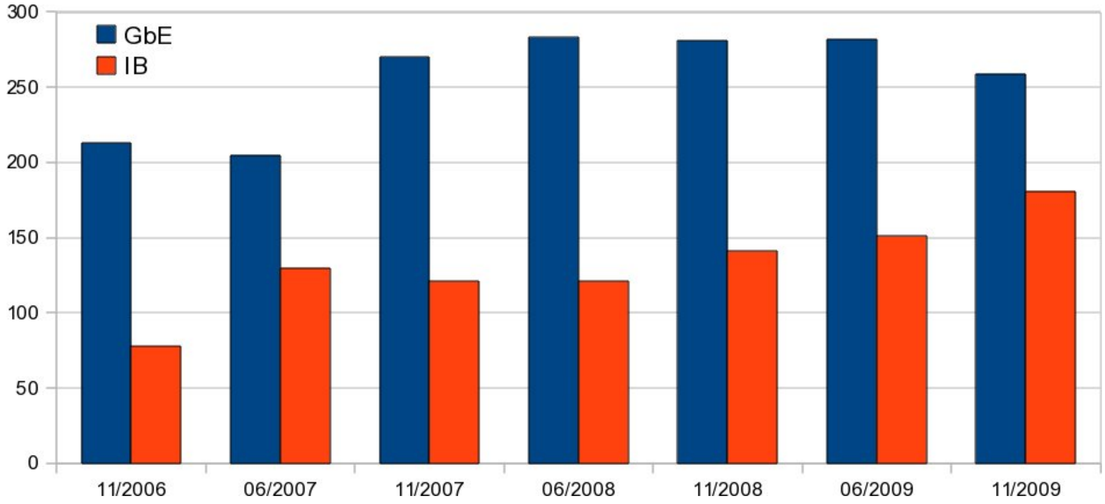


Figure 5.7: Top500 supercomputer interconnect: Ethernet vs InfiniBand

There are many vendors provide 10GbE devices, including chips, switches, NICs and others. However, there are only very few manufacturers (Mellanox and QLogic) provide chips for InfiniBand switches and channel adapters, and only a few vendors (such as Mellanox, QLogic, Voltaire, Cisco and others) provide InfiniBand switches and channel adapters. Obviously 10GbE has a wider support from the industry.

5.3.3.3 Difficulty of implementing

Except for the custom readout boards, the entire LHCb DAQ system is built out of commercial off-the-shelf (COTS) components. The readout boards are based on FPGA, so the intellectual property for DAQ network interfaces is quit critical for the implementation. With wide support from industrial, 10GbE hence has more solutions available, Altera has even provided the solution for 40/100 GbE. However, InfiniBand only gained very little support from FPGA vendors, and the implementation is more complex than Ethernet. We might need an embedded host-processor with a Real-Time Operating System (RTOS) to implement InfiniBand protocol. This is an obstacle to implement the DAQ system based on InfiniBand.

5.3.3.4 Maturity

To construct a network, the major hardware elements we need are the switches for the core layer, the top-of-rack (TOR) switches, NIC/HCA. For both 10GbE and InfiniBand, TOR switches and NIC/HCAs are mature enough for production, but the port density and bandwidth of a single core switch are still not enough so that we may need several core switches to construct the core layer of the readout network. For 10GbE, Voltaire 8500 Ethernet switch provides 288 non-blocking 10G ports. This is the highest density 10GbE switch today, but it works at Layer 2 only. For InfiniBand, QLogic 12800-360 supports the highest density of ports, up to 864 4×QDR ports with a switch capacity of 51.8 Tb/s in a single chassis.

With the existing switches today, InfiniBand switches are mature enough to construct such a DAQ network which demands high bandwidth. On the contrary, 10GbE switches have not provided enough bandwidth and port density. However, some Ethernet network equipment manufacturers are working on the next generation Ethernet switch toward 40/100GbE with high bandwidth and high density of 10GbE ports.

5.3.3.5 Quality of Service

In order to obtain a good performance and avoid packet loss, we have to design the whole system carefully. But the traffic pattern is unpredictable, so we also need to take advantage of the quality of service (QoS) mechanism to improve the network performance. In the field of computer networking, QoS refers to resource reservation control mechanisms rather than the achieved service quality. QoS is the ability to provide different priorities to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. There are two basic mechanisms for QoS: flow-control and traffic classification. In our DAQ system all traffics have the equal priority, so we will discuss flow-control only.

In IBA, the foundation operation is the ability of a consumer to queue up a set of instructions that the hardware executes. This facility is referred to as a work queue. Work queues are always created in pairs, called a Queue Pair (QP), one for send operations and one for receive operations. In general, the send work queue holds instructions that cause data to be transferred between the consumer's memory and another consumer's memory, and the receive work queue holds instructions about where to place data that is received from another consumer [106]. In order to prevent the loss of packets due to buffer overflows, IBA uses a credit-based link-level flow-control scheme [108, 109]. In such a scheme, the receiver on each link sends credits to the transmitter on the other end of the link. Credits are per Virtual Lane (VL)[1] and indicate the number of data packets that the receiver can accept on that VL. The transmitter does not send data packets unless the receiver indicates it has room. At regular intervals the receiver sends credit availability to the transmitter. The update interval depends on the load: high loads increase the frequency of updates, while low loads reduce the frequency. The mechanism ensures that packet loss is only a result of transmission errors and hence the available link bandwidth is used effectively.

In Ethernet, flow-control is a mechanism for temporarily stopping the transmission of data on a network on the data link layer. A situation may arise where a sender may be transmitting data faster than some other part of the network (including the receiver) can accept it. Flow-control is time-based mechanism. The overwhelmed network element will send a PAUSE frame to halt the transmission of the sender for a specified period of time so as to avoid buffer overflow. The pause-time is defined in the PAUSE frame, The current Ethernet flow-control is

[1] InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels on the same physical link. VLs provide the ability to support QoS.

on a per-port basis. A few amendments of “802.1Q IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks” are under working, such as “802.1Qau - Congestion Notification” [110] and “802.1Qbb - Priority-based Flow Control” [111].

5.3.3.6 Conclusion

InfiniBand and 10GbE are the two promising candidate technologies for the super-LHCb DAQ network. InfiniBand has a higher bandwidth and lower latency over 10GbE, however our application is not latency sensitive. The data rate of each CPU can process will be about 10Gb/s according to the moore's law at the time of the deployment, so the bandwidth of 10Gb/s should be enough for each CPU, even if it is not enough, another link can be added to form a aggregation link. The advantage of higher bandwidth of InfiniBand is only in the core layer of the network.

In addition to the performance, it is also critical to consider the cost to build a system. As mentioned in Section 5.3.3.2, InfiniBand is gaining more acceptance in HPC, but it is limited in the HPC area, and the number of the device suppliers is quite small, we can foresee that the price of InfiniBand will not drop dramatically comparing to the price of today. However, 10GbE is becoming more and more popular in the servers, and will also dominate the network of the desktop PC in the future. We can expect that 10GbE will replace 1GbE in the future and the price will drop to the level of 1GbE of today. Another consideration is the cost of cables. The maximum length of copper cables supported by InfiniBand is only 15m, which is not long enough for most of our connections, but the fiber with active optical modules is very expensive. However, 10GBase-T supports 100m in cat 6e cables, and 55m in cat 5e and cat 6 cables, this meets our requirement, and we could even reuse the installed cat 6 cables.

To sum up, InfiniBand has a better performance than 10GbE, but it is not obvious in our application; the cost of 10GbE is foreseen to be much lower than InfiniBand in the future. It is not necessary to decide which technology is the winner yet, some R&Ds need to be done before we jump to the conclusion. However, we assume 10GbE will be adopted preferably and the discussion of the following section is based on 10GbE.

5.4 Solution of the DAQ network

As mentioned in Section 5.2.2, the DAQ network needs to provide a bandwidth of ~30 Tb/s and 3556 10-Gb/s port for the readout boards, and we assume about 4000 CPUs will be needed in EFF. To achieve the given goal, we follow a number of basic principles in our design:

- The DAQ network should use a simple dataflow protocol, a push mode is adopted as mentioned in Section 5.2.2.
- We use industrial equipment and exist standards as much as possible.
- According to the comparison of 10GbE and InfiniBand, we use 10GbE in the design, but it is also applicable to InfiniBand with slight modification.
- The network connections for the readout boards and the HLT CPUs will be mixed together. Because the dataflow is unidirectional from the readout boards to the HLT CPUs, such a scheme can make use of the bisection bandwidth as much as possible.

In this section, we will present three possible schemes to implement the DAQ network for super-LHCb, and then focus on the most promising one which is based on low cost pizza-box [1] switches with scale-out approach.

5.4.1 Solution based on high-end switches

This solution is based on high-end switches which form the core switching layer. As shown in Fig. 5.8, the architecture is similar to the current design (see Fig. 2.9). In this scheme, we need several high-end switches in the core layer, which transfers the traffic from different edge switches. The core switches must provide very high bandwidth and port density. In the edge switch, the readout boards and the HLT CPUs are mixed together with a proper ratio which depends on the total number of the readout boards and the HLT CPUs we need. This is a f difference to the current design, which make use of the bisection bandwidth of the ports in the

[1] This kind of switch is 1U chassis, and it looks link a pizza box, so it is called pizza-box switch

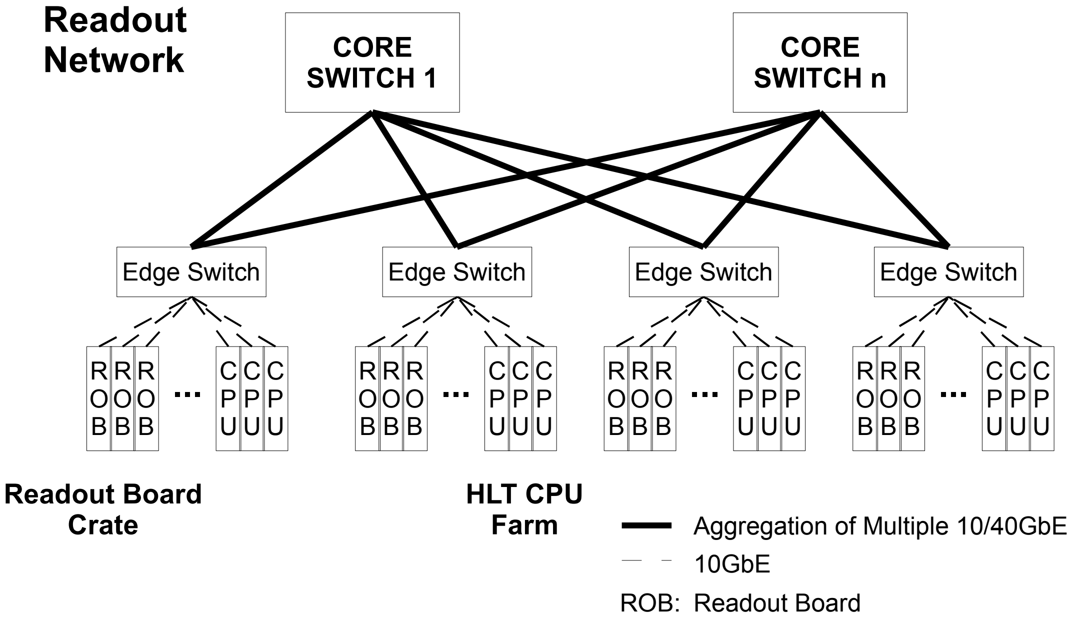


Figure 5.8: Network architecture based on high-end switches

core switches. And the edge switches have connections to each core switch, which enable the traffic from all readout boards to reach any HLT CPU.

This approach has a great simple architecture with typically high reliability and easy network management. However, this solution has a few key limitations:

- **Expense**: Larger chassises cost much more than multiple, smaller chassises with the same aggregate capacity. Normally the price per port of the high-end switch is 4 times higher than the small chassis.

- **Flexibility**: It is expensive and complex to re-scale the network after it is deployed.

5.4.2 Solution based on commodity pizza-box switches

In this solution, we use only the commodity low-cost pizza-box switches to build the DAQ network. We assume the port numbers of the switches are 24 or 48 as today. This is a 2-tier 4-stage network. In this scheme, the network is composed of a number of modules. The module is basically the two tiers (edge layer and aggregation layer) with a fat-tree topology [112]. The module is shown in Fig. 5.9, there are four 48-port edge switches, each contains 15 ports for the readout boards, 18 ports for the HLT CPUs and 15 ports for the uplinks to the aggregation layer. The aggregation layer is composed of two 48-port and one 24-port pizza-box switches, it provides 60 links to the edge layer and 60 links for the interconnection between modules. As shown in Fig. 5.10, the interconnection between modules is a full mesh topology[1] There are 61 modules in total, there is one link between any two modules. In total, this scheme provide 3660 ports for the readout boards and 4392 ports for the HLT CPUs.

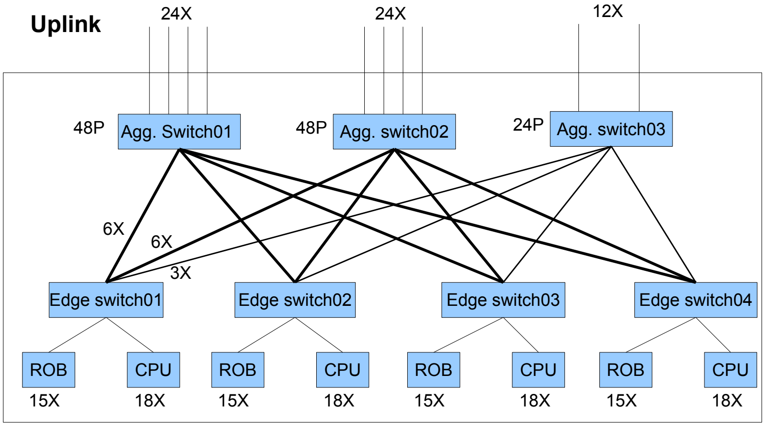


Figure 5.9: Network architecture based on pizza-box switches: diagram of basic modules

This scheme uses only the pizza-box switches to build the entire network, even though it needs more ports for the interconnection of the modules, the cost is still much lower than the solution based on high-end switches. However, the architecture of this scheme is more complicated, so a central management station is needed to monitor and configure all network components in real-time.

[1] A full mesh topology is a network topology in which there is a direct link between all pairs of nodes.

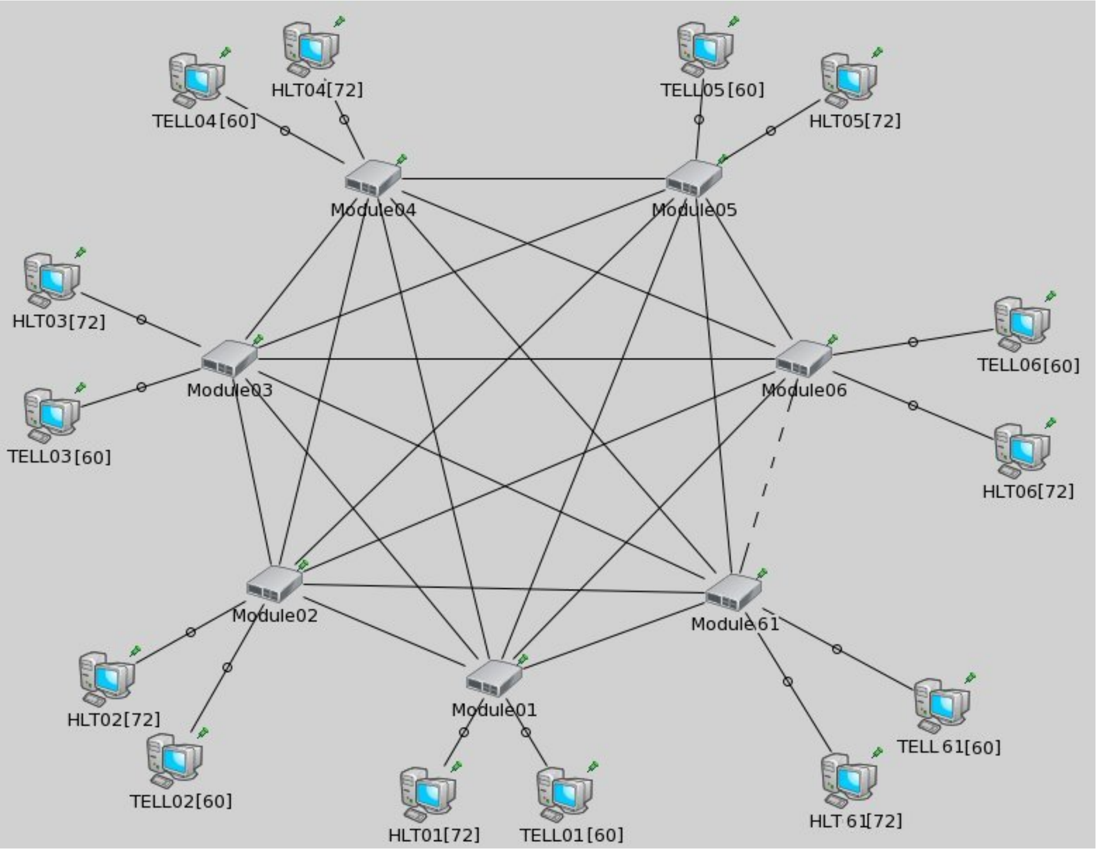


Figure 5.10: Network architecture based on pizza-box switches: topology of modules

5.4.3 Solution based on merchant networking ASIC

The last decade, some networking Application-Specific Integrated Circuits (ASIC) have been produced for the mass market of network equipments. there are currently at least three separate makers of 24-port 10GbE switch ASICs: Broadcom[1], Fulcrum[2] and Fujitsu[3]. These makers provide not only the ASIC, but also the design reference and software. Several companies have used these ASICs to build 10GbE TOR switches with custom software.

[1] Broadcom: www.broadcom.com

[2] Fulcrum: www.fulcrumnetwork.com

[3] Fujitsu: www.fujitsu.com

A solution based on merchant networking ASICs is to use the same architecture as the solution based on commodity pizza-box switches, however the modules are constructed using merchant ASICs. A diagram of the module based on merchant networking ASICs is shown in Fig. 5.11[113], the number of ASICs depends on the scale of the module.

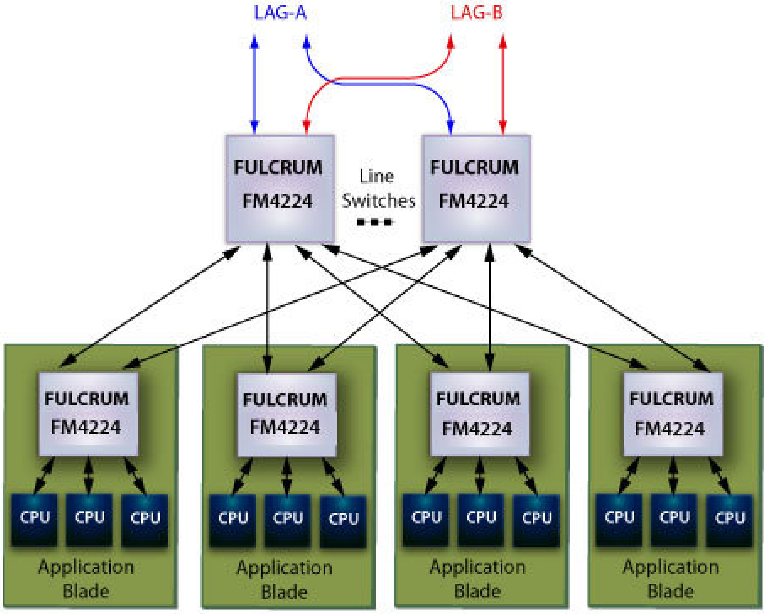


Figure 5.11: Network module based on merchant networking ASICs

The center for networked systems from University of California San Diego has implemented a 3456-port 10GbE system using merchant networking ASICs [114]. This solution provides flexibility to build a large network, and reduces the power, cost and cables. However this solution involves effort to the implementation of hardware and software of the modules, which need lot of experience and manpower to maintain such a system over the lifetime of an experiment. The trend is not to build but to buy COTS!

5.4.4 Conclusion

Comparing the features of the three approaches (include cost, flexibility, performance and difficulty of implementing), we consider the solution based on commodity pizza-box switches as the most promising one, the cost is low and the main efforts needed are mainly the network design and management.

5.5 Study of the solution based on commodity pizza-box switches

This is a large scale LAN, the design needs to be studied and verified. Network simulation is one of the approaches to evaluate the performance and other properties for study and engineering. In the following section, we will present the simulation results and optimized scheme, and the management of such a complex network is also discussed.

5.5.1 Network simulation

In the DAQ of the high energy physics experiments, all the sources send traffic to the same destination at one time. Unavoidably, there will be contention on the output ports at each stage. Buffers are required to temporarily store packets that lose the contention. When it runs out of buffers, the packets may get dropped randomly depends on the QoS strategy deployed in the switch.

Some simulations have been done to study the performance of the scheme. The simulation is based on the framework OMNET++ [115]. OMNET++ is an extensible, modular, component-based C++ simulation library and framework, with an Eclipse-based IDE and a graphical runtime environment, for more detail see Appendix A. Buffer occupancy and rate of packet drops are the most interesting parameters to study.

In the simulation, we have used the following assumptions or parameters:

1. All switches use a shared-memory queuing strategy [116], and total buffer size is 48 MB. The watermark is set to 90%, the the buffer occupancy exceeds the watermark, the packet will get drop randomly. We record not only the total buffer level, but also the buffer for each port, so the results are also applicable for the other memory strategies such as output queuing and combined input and output queuing.

2. The packet size from all readout boards are quite even, a uniform distribution between 1100 and 1300 is adopted.
3. The packet drop caused by bit-error is ignored, only the drop caused by buffer overflow is recorded.
4. The switching latency is 2μs.
5. We use Layer 2 switching only, and only unicast traffics are generated.

5.5.1.1 Simulation results of the scheme without optimization

Because of the nature of the LHCb experiment, all the fragments belong to the same collision are read-out synchronously in all FEE boards. In the readout boards, the re-assembled packets are sent out as soon as they are ready. Even though the time needed for the zero-suppression varies in different FEE boards, they are still quite synchronous to some extent. In such a scheme, we can foresee that there will be a lot of contention at the output port of each stage.

In the simulation, we assume the packets are sent out simultaneously to the same HLT CPU at one time. The simulation results are shown in Fig. 5.12. The maximum buffer occupancy is shown as a function of the relative input load[1] on the network. For both the edge switch and the aggregation switch, Fig. 5.12(a) shows the maximum occupancy of the total buffer, Fig. 5.12(b) shows the maximum occupancy of the buffer of an uplink port, Fig. 5.12(c) shows the maximum occupancy of the buffer of an edge port. For the edge switch, the uplink port is the port connects the aggregation switch, the edge port is the port connects the readout boards or the HLT CPUs. For the aggregation switch, the uplink port is the port connects the other modules, the edge port is the port connects the edge switches in the same module.

We assume that the input load is about 70%, and analysis the result at this load. The total buffer needed for each aggregation switch and edge switch is more than 17 MB; the requirements to the uplink ports are quite low (less than 100 kB); however the maximum buffer occupancy for the edge port are quite high, about 1.7 MB and 2.6 MB for the aggregation switch and the edge switch, which

[1] Load is the traffic rate in a switch or a switch port.

is a challenge for the output queuing strategy. Fig. 5.13 and Fig. 5.14 show the buffer occupancy of the aggregation switch and the edge switch as a function of time at the input load of 70%.

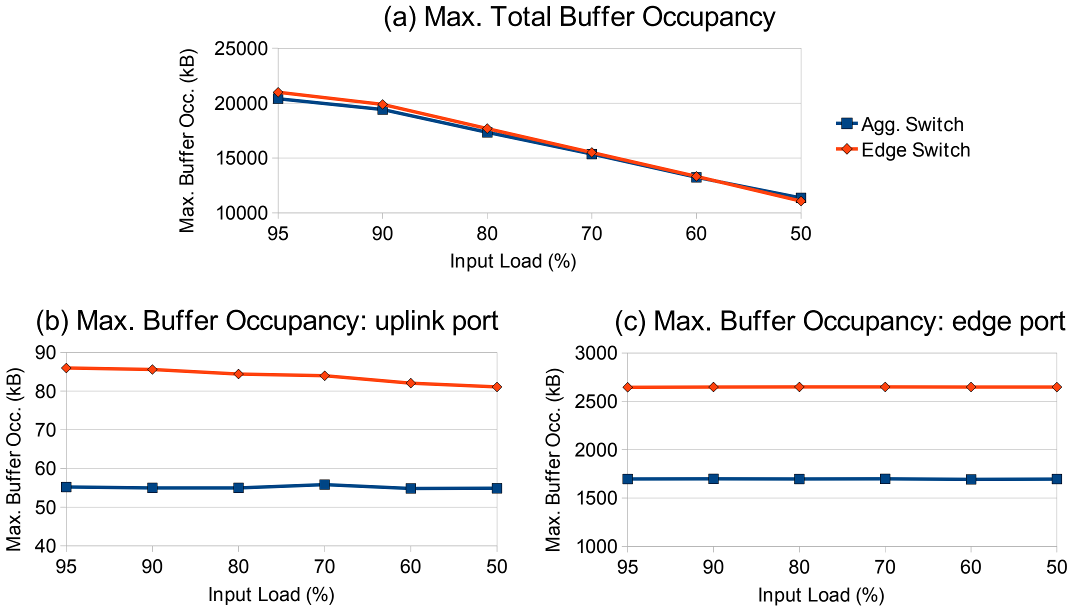


Figure 5.12: Maximum buffer occupancies

5.5.1.2 Simulation results of the scheme with optimization

Today, most 10GbE TOR switches use cut-through switching method (In general, there are three switching methods: Store-and-Forward Switching, Cut-Through Switching and Fragment-Free Switching, for more detail see Appendix B). Usually cut-through switches have only a small buffer, so it is difficult to buffer the whole event data in the switch. A time schedule for each readout board to send data is necessary. Each readout board sends data in different fixed slots to avoid congestion in the output port of the edge switch. A preliminary scheme is shown in Fig. 5.15, A, B, C, D and E are different collision events, each readout board sends out the event data at the assigned time slot.

In the section, we will present the simulation results with the above optimization scheme. Similar to Section 5.5.1.1, Fig. 5.16 shows the maximum buffer

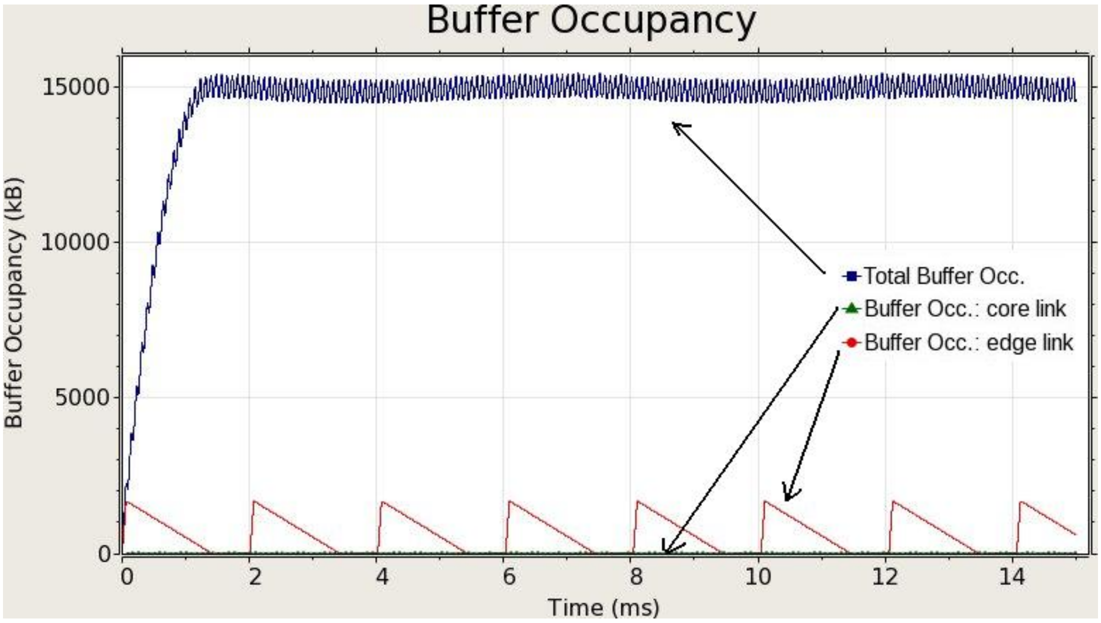


Figure 5.13: Buffer occupancy of aggregation switches

occupancy of the whole switch, the uplink port and the edge port for both aggregation switches and edge switches. Fig. 5.17 and Fig. 5.18 show the buffer occupancy of the aggregation switch and the edge switch as a function of time at the input load of 70% after using the optimization scheme.

At the input load of 70%, the maximum buffer occupancy drops dramatically from 17 MB to about 700 kB with the help of the optimization. For the edge port, the maximum buffer occupancy also decreases by a factor of 20, from 1.7 MB drops to ∼50 kB for the aggregation switch, from 2.6 MB drops to ∼65 kB for the edge switch.

5.5.1.3 The rate of packet drop

In the simulations, there is no packet drop in both cases (with and without optimization), because the buffer size is big enough to store the traffic temporally. However, the switch needs a big buffer if no optimization is implemented, especially for those with the buffer strategy of output queuing.

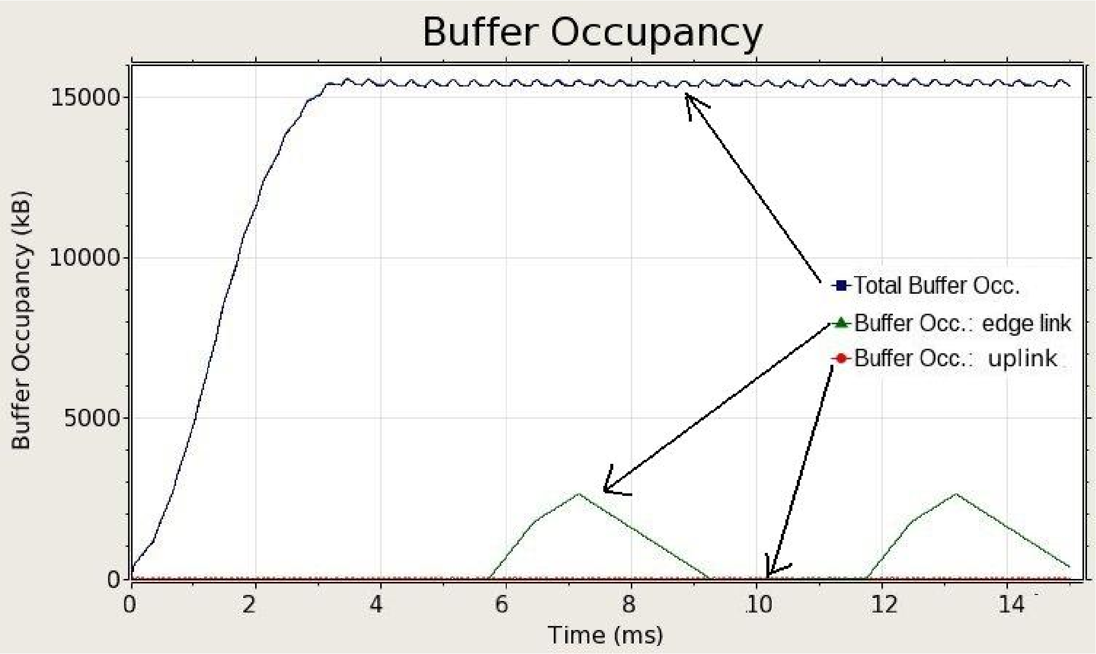


Figure 5.14: Buffer occupancy of edge switches

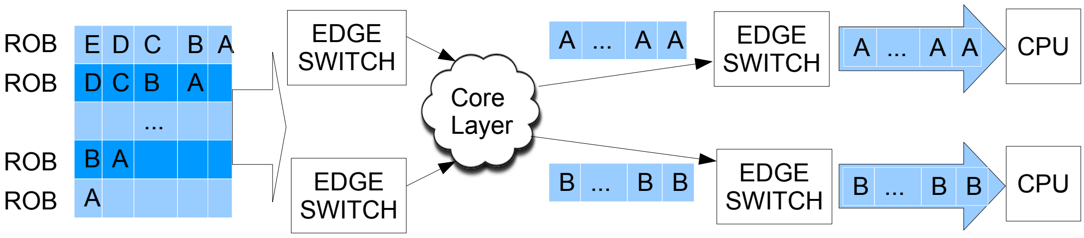


Figure 5.15: A preliminary scheme of slot schedule to send data for readout broads

5.5.2 Network management

Such a network provides high flexibility and redundancy, but its operation relies on the network management. In this circumstance, the network management becomes extremely important, it includes not only the management tasks introduced in Chapter 3 but also the routing table/forwarding table in all switches. Because all the elements have full connectivities in the network, any misconfigu-

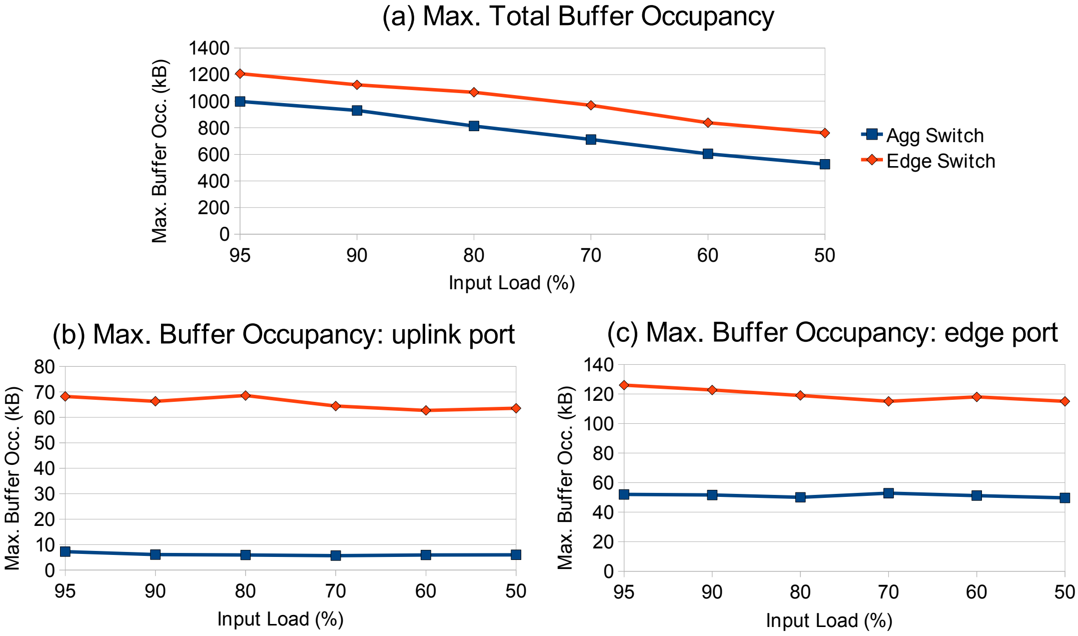


Figure 5.16: Maximum Buffer Occupancy after optimization

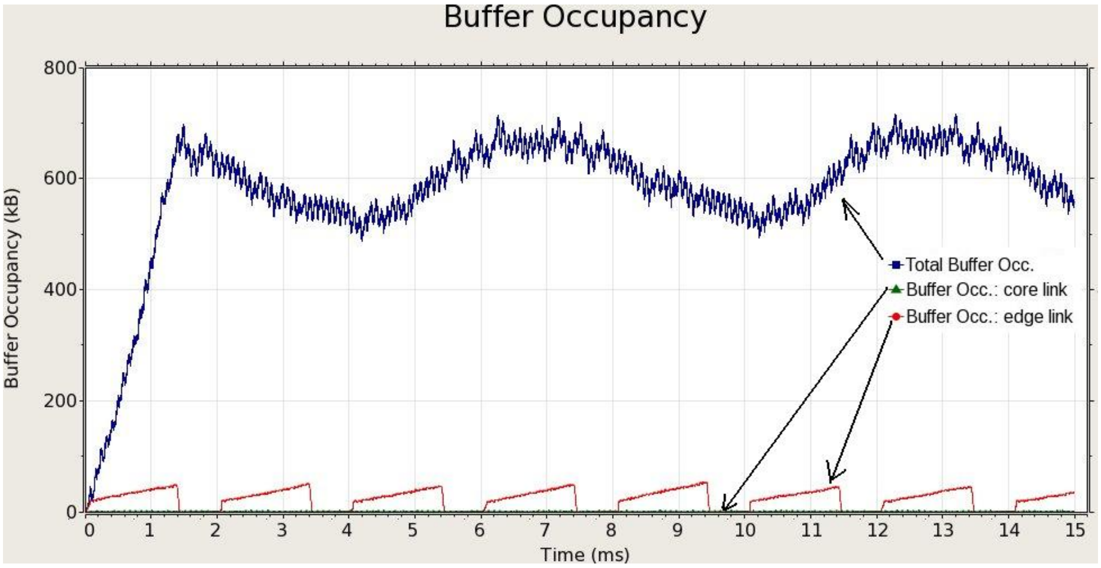


Figure 5.17: Buffer occupancy of aggregation switches after optimization

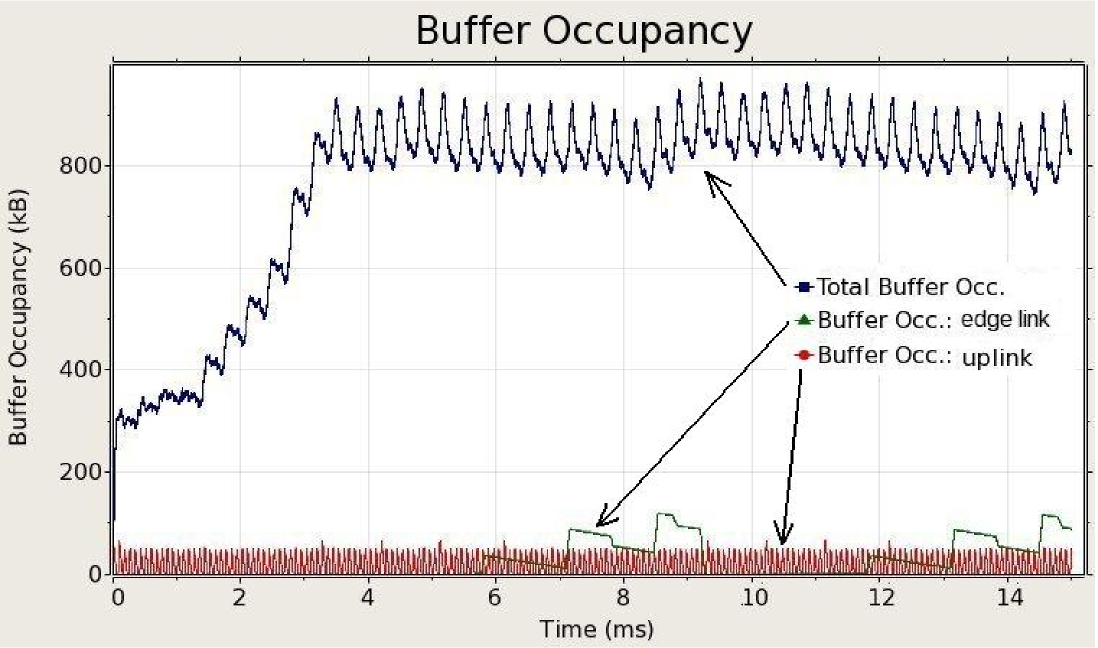


Figure 5.18: Buffer occupancy of edge switches after optimization

ration or misbehavior could easily lead to a loop. So the status of routing needs to be monitored centrally. For the operation and further provide fault tolerance, the status of all switch elements and links need to be monitored in real-time. If any link or switch failure is detected, a set of involved switches should be reconfigured immediately in order to recover from the situation.

5.6 Summary

In this chapter, we have introduced the plan of the LHC upgrade and the upgrade proposal of the LHCb experiment. To improve the trigger efficiency, the LHCb upgrade has opted for a trigger scheme that the trigger system is implemented in software applications run in a large event filter farm. This requires that all data is readout at the full 40 MHz rate of the LHC machine and transmitted to the event filter farm. The high readout rate results in the requirement of a high performance DAQ network. Two promising candidate technologies 10GbE and InfiniBand were studied and compared in several aspects: performance, market, difficulty of implementing, maturity and QoS. It is not yet decided which technology is the winner. Some R&D need to be done and development of these two technologies needs to be considered. However we assume 10GbE is more likely the technology we will adopt, and we proposed the possible solutions for the DAQ network based on 10GbE and have some simulations on the solution based on pizza-box lost cost switches. The solution can completely fulfill the requirement and provides high redundancy and flexibility with low cost.

Chapter 6

Conclusions

The LHCb Online system relies on the Online network, employing two large networks based on Gigabit Ethernet for its DAQ and ECS. The architecture of the Online network is primarily finished in 2005. However, there were still a lot of detail work to be finished since then. A few optimizations have been done to improve the performance of the networks. For example:

- The solution for redundancy is provided for the critical services to increase the high availability (see Section 2.3.1.3).
- The link aggregation scheme is adopted for the connection between the core switch and the edge switches. This scheme provides high redundancy for DAQ and increases the flexibility (see Section 2.3.2.3).

The LHCb Online network is a large scale LAN, a sophisticated monitoring is needed for its successful operation. The custom network monitoring system has been implemented using the same SCADA tool PVSS and the framework JCOP as the LHCb ECS. A large amount of information is collected through SNMP, sFlow and other sources by custom front-end processes efficiently. All the information is consolidated and processed by the PVSS supervisory layer to provide information at several levels of expertise:

- critical conditions and alarms for the non-expert shift-crew
- long term health-monitoring for the network administrators
- performance info for data acquisition experts

In the network monitoring system, all states and behaviors are modeled as finite state machine, and provides homogeneous interfaces for the operators as the LHCb ECS.

We have introduced a packet sampling mechanism called sFlow which allows to investigates the traffic flow statistically. In addition to the monitoring of the bandwidth occupancy and the health status of devices, more advanced monitoring of the network is necessitated. sFlow captures the IP header of sampled packets, this make it possible to investigate the content of the traffic. sFlow has been used for the trouble-shooting and the advanced monitoring of the DAQ network. In addition, we have demonstrated a network intrusion detection system (NIDS) based on sFlow and the open source NIDS Snort. In the experiment, this NIDS can effectively detect network anomalies and network intrusions, such as port scans and denial-of-service attacks.

Finally, a possible DAQ architecture for future LHCb upgrade (super-LHCb) is described. For high speed internetworking, there are two promising candidate technologies: 10GbE and InfiniBand. Based on the requirements of LHCb DAQ upgrade, we have compared these two technologies from the following aspects: performance, market, difficulty of implementing, maturity and QoS. It is not decided yet which technology will be chosen before more R&Ds are done. We assume 10GbE is the solution more likely and we have proposed three schemes based on 10GbE, however they are also applicable to InfiniBand. And we have done some simulations on the solution based on pizza-box switches, the results show that the DAQ network can be implemented with low cost pizza-box switches.

Appendix A

Network Simulation and OMNeT++

In communication and computer network research, network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/routers, data links, packets, etc) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab; various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions.

A network simulator is a software program that imitates the working of a computer network. In simulators, the computer network is typically modeled with devices, traffic etc and the performance is analyzed. There are several popular network simulator today:

- NS-2/NS-3 (Network Simulator): is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. It is a free software.

- OPNET (Optimized Network Engineering Tools): is a commercial software for network modeling and simulation.

- GloMoSim (Global Mobile Information System Simulator): is designed using the parallel discrete event simulation capability provided by Parsec (a parallel programming language). It currently supports protocols for a purely wireless network.

- OMNeT++ (Objective Modular Network Testbed in C++): It is a discrete event simulation tool designed to simulate computer networks, multiprocessors and other distributed systems. Its applications can be extended for modeling other systems as well. OMNeT++ is a free community version, OMNEST is the commercial version.

OMNeT++ is a discrete event simulation environment. It is free for academic and non-profit use. Its primary application area is the simulation of communication networks, but because of its generic and flexible architecture, is successfully used in other areas like the simulation of complex IT systems, queuing networks or hardware architectures as well.

OMNeT++ provides a component architecture for models. An OMNeT++ model consists of modules that communicate with message passing. The active modules are termed simple modules, where the network protocols are implemented. They are written in C++, using the simulation class library. Simple modules can be grouped into compound modules and so forth. The user defines the structure of the model in NED language descriptions (Network Description). Besides the protocol and topology, we still need to generate traffic for the simulation, which is implemented via message in OMNeT++. There is already a basic class cMessage in the simulation library, we will need to subclass cMessage and add various fields to it to simulate the traffic packets. At the end, we need to configure the simulation before running it.

Appendix B

Switching Methods

In general, there are three switching methods: store-and-forward, cut-through and fragment-free. In the store-and-forward switching method, error checking is performed against the frame, and any frame with errors is discarded. With the cut-through switching method, no error checking is performed against the frame, which makes forwarding the frame through the switch faster than store-and-forward switches. Fragment-free switching is a modified form of cut-through switching, the switch with fragment-free switching checks the first 64 bytes.

B.1 Store-and-forward switching

With store-and-forward switching, the switch must temporarily stores the entire frame into the buffer of the inbound port and check the cyclic redundancy check (CRC) field before any additional processing. When checking the CRC filed, the switch will calculate a CRC value, just as the source device did, and compare this value to what was included in the frame. If they are the same, the frame is considered good and will be forwarded out through the correct destination port; otherwise, it is discarded.

Store-and-forward switching is therefore the best forwarding mode to prevent errors being forwarded throughout the network. However, a drawback to the store-and-forward switching method is the high latency for a frame to pass through the switch. Another drawback is that the switch requires more memory and processor cycles to perform the detailed inspection of each frame than that of cut-through or fragment-free switching.

B.2 Cut-through switching

With cut-through switching, the switch reads only the very first part of the frame before making a switching decision. Once the switch device reads the destination MAC address (8-byte preamble and 6-byte MAC address), it begins forwarding the frame (even though the frame may still be coming into the interface). A cut-through switch reduces delay because the switch begins to forward the frame as soon as it reads the destination MAC address and determines the outgoing switch port. Its biggest problem, though, is that the switch may be switching bad frames since the header could be legible, but the rest of the frame corrupted from a late collision.

B.3 Fragment-free switching

Fragment-free switching can be viewed as a compromise between store-and-forward switching and cut-through switching. Fragment-free switching works like cut-through switching with the exception that a switch in fragment-free mode stores the first 64 bytes of the frame from the source to detect a collision before forwarding. Frames are forwarded before any checksums can be calculated. The reason fragment-free switching stores only the first 64 bytes of the frame is that most network errors and collisions occur during the first 64 bytes of a frame.

This is only useful if there is a chance of a collision on the source port - so a fully switched network does not benefit from fragment free. Nowadays, the most popular switching modes are store-and-forward and cut-through.

References

[1] O. S. Brning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, P. Proudlock, *et al.*, “LHC Design Report, v 1; The LHC Main Ring,” 2004. CERN-2004-003-V-1. 1

[2] O. S. Brning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, P. Proudlock, *et al.*, “LHC Design Report, v 1; The LHC Main Ring, Ch 12 Vacuum system,” 2004. CERN-2004-003-V-1. 3

[3] V. Parma and L. Rossi, “Performance of the LHC magnet system,” Sep 2009. 3

[4] ATLAS collaboration, “The ATLAS experiment at the CERN Large Hadron Collider,” *JINST*, vol. 3, p. S08003, 2008. 3

[5] CMS collaboration, “The CMS experiment at the LHC,” *JINST*, vol. 3, p. S08004, 2008. 4

[6] LHCb collaboration, “The LHCb Detector at the LHC,” *JINST*, vol. 3, p. S08005, 2008. 5, 20, 21, 27

[7] ALICE collaboration, “The ALICE experiment at the LHC,” *JINST*, vol. 3, p. S08006, 2008. 5

[8] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay, “Evidence for the 2π Decay of the $K2$ Meson,” *Phys. Rev. Lett.*, vol. 13, pp. 138–140, Jul 1964. 6

[9] N. Cabibbo, "Unitary Symmetry and Leptonic Decays," *Phys. Rev. Lett.*, vol. 10, pp. 531–533, Jun 1963. 6

[10] M. Kobayashi and T. Maskawa, "CP-Violation in the Renormalizable Theory of Weak Interaction," *Progress of Theoretical Physics*, vol. 49, no. 2, pp. 652–657, 1973. 6

[11] A. D. Sakharov, "Violation of CP in variance, C asymmetry, and baryon asymmetry of the universe," *Soviet Physics Journal of Experimental and Theoretical Physics (JETP)*, vol. 5, pp. 24–27, 1967. 6

[12] LHCb collaboration, "LHCb Magnet: Technical Design Report," 2000. CERN-LHCC-2000-007. 9

[13] LHCb collaboration, "LHCb VELO: Technical Design Report," 2001. CERN-LHCC-2001-011. 10

[14] LHCb collaboration, "LHCb Reoptimized Detector Design and Performance: Technical Design Report," 2003. CERN-LHCC-2003-030. 13, 18, 21

[15] LHCb collaboration, "LHCb inner tracker: Technical Design Report," 2002. CERN-LHCC-2002-029. 14

[16] LHCb collaboration, "LHCb outer tracker: Technical Design Report," 2001. CERN-LHCC-2001-024. 16

[17] LHCb collaboration, "LHCb RICH: Technical Design Report," 2000. CERN-LHCC-2000-037. 18

[18] J. Jelley, *Cherenkov Radiation and its Applications*. July 1955. 18

[19] S. Barsuk, "The Shashlik Electro-magnetic Calorimeter for the LHCb Experiment," *11th International Conference On Calorimetry In High Energy Physics*, 2006. 20

[20] LHCb collaboration, "LHCb Technical Proposal," 1998. 20

[21] LHCb collaboration, "LHCb muon system: Technical Design Report," 2001. CERN-LHCC-2001-010. 21

[22] LHCb collaboration, "LHCb trigger system: Technical Design Report," 2003. CERN-LHCC-2003-031. 23

[23] E. Rodriguesa, "The LHCb Trigger System," *Nuclear Physics B (Proc. Suppl.)*, vol. 170, pp. 298–302, 2007. 23, 24

[24] LHCb collaboration, "LHCb online system, data acquisition and experiment control: Technical Design Report," 2001. CERN-LHCC-2001-40. 29, 34, 43

[25] LHCb collaboration, "Addendum to the LHCb Online System Technical Design Report," 2005. CERN-LHCC-2005-039. 29, 43

[26] G. Haefeli, A. Bay, A. Gong, H. Gong, M. Mcke, N. Neufeld, and O. Schneider, "The LHCb DAQ interface board TELL1," *Nucl. Instrum. Methods Phys. Res., A*, vol. 560, pp. 494–502, 2006. 31

[27] B. Jost and N. Neufeld, "Raw-data transport format," 2004. technical document. 32

[28] C. Gaspar, B. Franek, R. Jacobsson, B. Jost, S. Morlini, N. Neufeld, and P. Vannerem, "An integrated experiment control system, architecture, and benefits: the LHCb approach," *Nuclear Science, IEEE Transactions on*, vol. 51, pp. 513–520, June 2004. 34

[29] ETM professional control, "PVSS." `http://www.etm.at/index_e.asp`. 34, 53

[30] l. Holme, M. Gonzlez-Berges, P. Golonka, and S. Schmeling, "The JCOP Framework," Tech. Rep. CERN-OPEN-2005-027, CERN, Geneva, Sep 2005. 34

[31] B. Franek and C. Gaspar, "SMI++: Object oriented framework for designing and implementing distributed control systems," 2004. Presented at

2004 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS / MIC), Rome, Italy, 16-22 Oct 2004. 36, 85

[32] B. Jost and N. Neufeld, “LHCb Online Networking Requirements,” Tech. Rep. LHCb-2003-166, CERN, Geneva, Dec 2003. 36

[33] Force10 networks, *Force10 E-Series Manuals*, 2009. 38, 44, 47, 81

[34] HP Procurve, *Management and Configuration Guide for the ProCurve Switch 2600 Series, Switch 2600-PWR Series, Switch 2800 Series, Switch 4100gl Series, and Switch 6108*, 2009. 38

[35] “Link Aggregation,” no. 802.3ad, 2000. 40, 46

[36] T. Davis, *Linux Ethernet Bonding Driver mini-howto*, 2000. 40

[37] HP Procurve, *Management and Configuration Guide for the ProCurve Series 3500, 3500yl, 5400zl, 6200yl, 6600, and 8200zl Switches*, 2009. 40

[38] LHCb Collaboration, *Data sizes of raw event*, 2006. 42

[39] J. P. Dufey *et al.*, “The LHCb trigger and data acquisition system,” *IEEE Trans. Nucl. Sci.*, vol. 47, pp. 86–90, 2000. 43

[40] A. Barczyk, J. P. Dufey, C. Gaspar, P. Gavillet, R. Jacobsson, B. Jost, N. Neufeld, and P. Vannerem, “The new LHCb trigger and DAQ strategy: a system architecture based on gigabit-ethernet,” *IEEE Trans. Nucl. Sci.*, vol. 51, no. 3, pp. 456–60, 2004. 43

[41] P. Jenni, M. Nessi, M. Nordberg, and K. Smith, *ATLAS high-level trigger, data-acquisition and controls: Technical Design Report.* Technical Design Report ATLAS, Geneva: CERN, 2003. 43

[42] S. Cittolin, A. Rcz, and P. Sphicas, *CMS trigger and data-acquisition project: Technical Design Report.* Technical Design Report CMS, Geneva: CERN, 2002. 43

[43] A. Barczyk, G. Haefeli, R. Jacobsson, B. Jost, and N. Neufeld, "1 MHz Readout," Tech. Rep. LHCb-2005-062. CERN-LHCb-2005-062, CERN, Geneva, Sep 2005. 43

[44] The Tolly Group, *Force10 Networks, Inc. TeraScale E-Series E1200 Resilient Switch/Router 10-GbE and GbE Zero-Loss Throughput*, 2004. 46

[45] "iperf." `http://sourceforge.net/projects/iperf/`. 47

[46] D. Bailey and E. Wright, *Practical SCADA for Industry.* Elsevier, 2003. 52

[47] A. Daneels and W. Salter, "What is SCADA?." `http://ref.web.cern.ch/ref/CERN/CNL/2000/003/scada/`. 52

[48] ETM professional control, "PVSS Manual." 53

[49] C. Gaspar, "PVSS Introduction for Newcomers." `http://lhcb-online.web.cern.ch/lhcb-online/ecs/PVSSIntro.htm`. 53

[50] P. C. Burkimsher, "JCOP Experience with a Commercial SCADA Product, PVSS," 2003. 55

[51] O. Holme, M. Gonzlez-Berges, P. Golonka, and S. Schmeling, "The JCOP Framework," Tech. Rep. CERN-OPEN-2005-027, CERN, Geneva, Sep 2005. 55

[52] ITU-T, "X.700 Management framework for open systems interconnection for CCITT applications," 1992. 56

[53] "DIM: Distributed Information Management System." `http://dim.web.cern.ch/dim/`. 58

[54] C. Gaspar, P. Charpentier, and M. Dnszelmann, "DIM, a portable, light weight package for information publishing, data transfer and inter-process communication," *Comput. Phys. Commun.*, vol. 140, no. 1-2, pp. 102–9, 2001. 58

[55] "Pvss-dim integration." `http://lhcb-online.web.cern.ch/lhcb-online/ecs/fw/FwDim.html`. 59

[56] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 (Third Edition)*. Addison-Wesley Professional, 1999. 59, 83

[57] "Simple Network Management Protocol." `http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol`. 59

[58] "Net-SNMP." `http://www.net-snmp.org/`. 61

[59] "sFlow (on wikipeida)." `http://en.wikipedia.org/wiki/sflow`. 63

[60] "sFlow." `http://www.inmon.com/`. 63, 95

[61] P. Phaal, S. Panchen, and N. McKee, "InMon Corporation's sFlow: a Method for Monitoring Traffic in Switched and Routed Networks." RFC 3176, 2001. 63

[62] "User interface (on wikipeida)." `http://en.wikipedia.org/wiki/User_interface`. 68

[63] K. Ahmat, "Ethernet Topology Discovery: A Survey," *CoRR*, vol. abs/0907.3095, 2009. 70

[64] "Overlay network (on wikipeida)." `http://en.wikipedia.org/wiki/Overlay_network`. 70

[65] "Overlay network (on wikipeida)." `http://en.wikipedia.org/wiki/Link_Layer_Discovery_Protocol`. 71

[66] IEEE, "Station and Media Access Control Connectivity Discovery." IEEE 802.1AB-2005 standard, 2005. 71

[67] A. Bierman and K. Jones, "Physical Topology MIB." RFC 2922, 2000. 71

[68] K. McCloghrie and M. T. Rose, "Management Information Base for Network Management of TCP/IP-based internets:MIB-II," 1991. 77

[69] IEEE, "IEEE Std 802.3 - 2005 Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications," 2005. 77

[70] "PyDim." http://lbdoc.cern.ch/pydim/. 83

[71] C. Lonvick, "The BSD Syslog Protocol," 2001. 83

[72] P.-Y. DUVAL, "Guide for ECS FSM design in LHCb sub-detectors LHCB Technical," tech. rep., 2007. 86

[73] C. Gaspar, "LHCb ECS Guidelines," tech. rep., 2007. 86

[74] "Tcpdump homepage." http://www.tcpdump.org. 89

[75] "Wireshark homepage." http://www.wireshark.org. 89

[76] Cisco Systems, Inc., "White paper: Introduction to Cisco IOS NetFlow - A Technical Overview ." 90

[77] P. Phaal and S. Panchen, "Packet sampling basics." http://www.sflow.org/packetSamplingBasics/index.htm. 90

[78] J. Jedwab, P. Phaal, and B. Pinna, "Traffic estimation for the large source on a network using packet sampling with limited storage." http://www.hpl.hp.com/techreports/92/HPL-92-35.pdf. 90

[79] M. D. Ciobotaru, *Characterizing, managing and monitoring the networks for the ATLAS data acquisition system.* PhD thesis, 2007. 90

[80] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1.* 1991. 90, 91

[81] "MySQL homepage." http://www.mysql.com/. 97

[82] INMON CORP., "Using sFlow and InMon Traffic Server for Intrusion Detection and other Security Applications," 99

[83] J. Reves and S. Panchen, "Traffic Monitoring with Packet-Based Sampling for Defense against Security Threats," 99

[84] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2191–2204, 2003. 99

[85] R. Kawahara, T. Mori, N. Kamiyama, S. Harada, and S. Asano, "A Study on Detecting Network Anomalies Using Sampled Flow Statistics," in *SAINT-W '07: Proceedings of the 2007 International Symposium on Applications and the Internet Workshops*, (Washington, DC, USA), p. 81, IEEE Computer Society, 2007. 99

[86] M. Canini, D. Fay, D. J. Miller, A. W. Moore, and R. Bolla, "Per Flow Packet Sampling for High-Speed Network Monitoring," in *Proceedings of the First International Conference on Communication Systems and Networks (COMSNETS'09)*, Jan 2009. 99

[87] "Snort homepage." `http://www.snort.org/`. 100

[88] The Snort Project, *SNORT Users Manual*. 100

[89] R. E. Jurga and M. losz Marian Hulboj, "Packet Sampling for Network Monitoring," tech. rep., 2007. CERN openlab technical report. 103

[90] J.-P. Koutchouk and F. Zimmermann, "LHC Upgrade Scenarios," Tech. Rep. sLHC-PROJECT-Report-0013. CERN-sLHC-PROJECT-Report-0013, CERN, Geneva, Jul 2009. 107

[91] R. Ostojic, "LHC Interaction region upgrade: Phase-1: goals and conceptual design," p. 6 p, 2009. 107

[92] W. Scandale and F. Zimmermann, "LHC Phase-2 upgrade scenarios," p. 13 p, 2009. 107

[93] "SLHC The High Luminosity Upgrade," 2009. Presentation in 2nd SLHC-PP annual meeting, 26th February 2009. 107

[94] T. Nakada, O. Ullaland, and W. Witzelling, "Expression of Interest for an LHCb Upgrade," Tech. Rep. LHCC-G-139. CERN-LHCC-2008-007, CERN, Geneva, Apr 2008. 108, 109, 110, 111

[95] D. Bortoletto, "The ATLAS and CMS Plans for the LHC Luminosity Upgrade," Tech. Rep. arXiv:0809.0671, Sep 2008. Contribution to HCP2008

the 19th Hadron Collider Physics Symposium 2008, Galena, Illinois, May 27-31 2008. 109

[96] "LHC Upgrade Plan and Ideas," 2007. Presentation in LHCb Upgrade Workshop, January 2007. 109

[97] F. Alessio, Z. Guzik, and R. Jacobsson, "Timing and Fast Control and Readout Electronics Aspects of the LHCb Upgrade," Tech. Rep. LHCb-2008-072. CERN-LHCb-2008-072, CERN, Geneva, Dec 2008. 109

[98] F. Alessio, Z. Guzik, and R. Jacobsson, "A 40MHz Trigger-free Readout Architecture for the LHCb experiment," in *16th IEEE NPSS Real Time Conference 2009*, May 2009. 109

[99] B. Jost, "Online System Upgrade Activities," 2009. Presentation in the LHCb Upgrade Meeting, Feb. 2009, CERN. 110

[100] K. Wyllie, "Status of electronics developments for LHCb upgrade," 2009. Presentation in the LHCb Upgrade Meeting, 15 July 2009, CERN. 111

[101] A. Gong, "Preliminary thoughts for the new TELL10 process card ," 2009. Presentation in the LHCb Electronics Upgrade Meeting, 26 Oct. 2009, CERN. 111

[102] N. Neufeld, "10 Gigabit technologies for a 40 MHz readout," 2007. Presentation in the 1st LHCb Collaboration Upgrade Workshop, January 2007, Edinburgh. 111

[103] IEEE Computer Society, "IEEE Std. 802.3-2008: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," 2008. 112

[104] Cisco Systems, Inc., "Using 10 Gigabit Ethernet Interconnect for Computational Fluid Dynamics in Automotive Design and Engineering." CISCO whitepaper. 113, 117

[105] "Infiniband Trade Association." `http://www.infinibandta.org`. 114

[106] InfiniBand Trade Association, “InfiniBand Architecture Specification, Volume 1, Release 1.2.1.” 115, 120

[107] “Top500 Supercomputer Sites.” http://www.top500.org. 118

[108] F. J. Alfaro, S. Sanchez, and D. Duato, “QoS in InfiniBand Subnetworks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 9, pp. 810–823, 2004. 120

[109] S.-A. Reinemo, T. Skeie, T. Sødring, O. Lysne, and O. Tørudbakken, “An overview of QoS capabilities in InfiniBand, Advanced Switching Interconnect, and Ethernet,” *IEEE Communications Magazine*, vol. 44, no. 7, pp. 32–38, 2006. 120

[110] “802.1Qau - Congestion Notification.” http://www.ieee802.org/1/pages/802.1au.html. 121

[111] “802.1Qbb - Priority-based Flow Control.” http://www.ieee802.org/1/pages/802.1bb.html. 121

[112] C. E. Leiserson, “Fat-trees: universal networks for hardware-efficient supercomputing,” *IEEE Trans. Comput.*, vol. 34, no. 10, pp. 892–901, 1985. 124

[113] Fulcrum microsytems, “Load balancing in telecom servers using focalpoint.” Whitepaper. 126

[114] N. Farrington, E. Rubow, and A. Vahdat, “Data center switch architecture in the age of merchant silicon,” in *HOTI ’09: Proceedings of the 2009 17th IEEE Symposium on High Performance Interconnects*, (Washington, DC, USA), pp. 93–102, IEEE Computer Society, 2009. 126

[115] “OMNET++.” http://www.omnetpp.org. 127

[116] H. J. CHAO and B. LIU, *high performance switches and routers.* 2007. 127