



Institut Supérieur D'Informatique de
Modélisation et de leurs Applications

COMPLEXE DES CEZEAUX

BP 125 – 63173 AUBIERE CEDEX



Centre Européen de Recherche
Nucléaire

1211 Genève Suisse

Rapport de Stage 2^{ème} année

Surveillance du réseau de l'expérience LHCB

Présenté par: **Oualid LEBTAINI**

Responsable ISIMA : **Emanuel Mesnard**

Responsable CERN : **Artur Barczyk**

Date : du **03/04** à **01/09 2006**

Durée: **5 mois**

REMERCEMENTS

Je tiens à remercier particulièrement mon responsable de stage Monsieur Artur Barczyk pour sa disponibilité, son aide technique et administrative, et ses explications et remarques pertinentes.

Je remercie madame Clara Gaspar pour ses aides techniques en PVSS, ses remarques pertinentes et de m'avoir aidé et fournir les informations nécessaires à la bonne compréhension du PVSS.

Je tiens aussi à remercier tout les membres du groupe LHCBONLINE pour leurs accueil chaleureux au sein du groupe et leurs aide, Spécialement M. Niko Neufeld, Loïc bardera, Gary Moine, Badri, Lanta Abadie et les membres des secrétariats PH et LHCb.

Glossaire

- ❖ **DAQ** : Data acquisition, c'est un sous système de LHCb ONLINE qui concerne tout le trafic entre les électroniques et les détecteurs du LHCb.
- ❖ **SCADA** : Ce sont les systèmes logiciels commerciaux utilisés intensivement dans l'industrie pour la surveillance et le control des processus industriels.
- ❖ **PVSS** : Système de contrôle et de visualisation de processus, est un produit industriel, de type SCADA
- ❖ **DIM** : DIM est un système de communication pour environnements distribués/mixtes, il fournit une couche transparente de communication d'interprocessus de réseau
- ❖ **LHCb** : Détecteur de beauté du grand collisionneur d'hadrons, c'est la nouvelle expérience du CERN
- ❖ **DP** : *DATAPOINT, Point De Données* : Des variables de PVSS utilisé comme les variables globaux qui permettront de conserver les données nécessaires.
- ❖ **DPE** : Un élément de Points de Données. Il peut être sous forme d'un nœud (structure) ou d'un type classique (entier, décimale, caractère etc.)

TABLE DES FIGURES ET ILLUSTRATIONS

Figure 1 CERN site de Meyrin.....	9
Figure 2 Circonférence des détecteurs CERN	10
Figure 3 Structure de LHCb	11
Figure 4 Tunnel (accélérateur à gauche, couloir de circulation).....	12
Figure 5 Chambres du réseau du DAQ dans le puits	13
Figure 6 Exemple d'un Patch Panel	13
Figure 7 Topologie du réseau de control de test	14
Figure 8 Topologie du réseau de donnés de test	15
Figure 9 Structure de réseau de l'expérience LHCb	17
Figure 10 Structure du système de contrôle de l'application	17
Figure 11 Structure de fonctionnement de DIM.....	18
Figure 12 1 ^{ère} version de la structure des services exportant les Statues des ports	20
Figure 13 La version finale de la structure de services de Statues des ports	21
Figure 14 exemple d'inscription des services à afficher par DIM	22
Figure 15 inscription des services simples	22
Figure 16 Commandes de DIM : Structure et exécution	23
Figure 17 RPC Structure et exécution	24
Figure 18 Structure des Points de données du projet.....	25
Figure 19 Calcule et mise à jour des statues des équipements avec les contrôles manageurs	27
Figure 20 Convention des Statues exportées et calculé des ports	28
Figure 21 Statues des Switch : calcule et conventions	29
Figure 22 Structure hiérarchique des panneaux de surveillance	30
Figure 23 Panneau de configurations	31
Figure 24 Panneau d'ajout de référence PVSS.....	31
Figure 25 Panneau de surveillance globale.....	32
Figure 26 Liste des Cartes d'un Switch de type TerasCale.....	32
Figure 27 Liste des Port du grand Switch TerasCale	34
Figure 28 Liste des ports Type HP Gigabit	35
Figure 29 Statistique d'un port.....	36
Figure 30 Panneau de configuration de base de données	37
Figure 31 Panneau de configuration multiple des ports	38

RESUME

Dans le cadre de L'expérience LHCB au CERN, le groupe LHCBONLINE aura à surveiller le bon déroulement des acheminements des données via le réseau. On distinguera deux types : un réseau de donnée qui transmettent les données à grande vitesse et un réseau de contrôle qui contrôle les équipements réseau et les PC. Pour l'application de surveillance on utilise l'outil de contrôle des systèmes SCADA PVSS et le Framework qui nous permet de recueillir des informations sur le réseau envoyer par un serveur via DIM et parfois selon le besoin, l'application peut prendre des actions et les transmettre via DIM pour les appliquer sur les équipements.

Dans le réseau on aura un grand nombre de Switch Ethernet à contrôler et à configurer, HP 3400 (48 port Gigabit) et un grand router Switch TerasCale (1256 ports) plus plusieurs serveurs et PC. La supervision des Switch se fait en utilisant le protocole SNMP. Il faut intégrer cette supervision dans un système SCADA (PVSS) et automatiser la configuration des équipements qui est disponible que par la ligne de commande à l'aide de l'outil EXPECT et les intégrer dans le système de contrôle distribué.

A la fin de notre application on a pu faire une première vue globale du test qui marche en utilisant un Processeur réseau. On a utilisé la base de données contenant les informations nécessaires sur les équipements.

Mots clés : **SCADA, PVSS, DIM, BDD, HP Procurve, TerasCale, Expect.**

ABSTRACT

In the new experiment for the LHCB CERN, we have a lot of devices to monitor. Especially the system has about 400 hundred of HP Ethernet switches and a TerasCale router switch. These devices are monitored via SNMP. In our application we are using a SCADA system (PVSS) to integrate the monitoring of network in our sub system. With Framework component we can send or receipt information from a server via DIM

We have two type of the network. Control network and data's one. Both of them need to be monitored by PVSS that we use with the framework component: especially DIM. This component permits to use information got by a server and may get some action to be applied in this server via command.

In LHCB network we have a big number of Ethernet switch's to be monitored. Basically we have a HP 3400 switch's and TerasCale switch this one is the bigger switch to be used in a data acquisition network (1260 ports) and also a lot of PCs.

The switch monitoring use SNMP witch must be configured and integrated to lhcb system control using PVSS (SCADA system). The configuration for this device should be programmed using library tools like expect. The all should be integrated to the distributed control system.

At the end of my placement we have tested this project using a network processor. We use data base to read all necessary information for all device in network monitoring system.

Key words: **SCADA, PVSS, DIM, BDD, HP Procurve, TerasCale, Expect.**

TABLE DES MATIERES

REMERCEMENTS

GLOSSAIRE

TABLES DES FIGURES ET ILLUSTRATIONS

RESUME, ABSTRACT

TABLE DES MATIERES

INTRODUCTION.....	7
I- CERN.....	8
1. Présentation	8
2. LHC.....	10
4. LHCBOONLINE	15
5. Les system SCADA et PVSS :.....	15
a. SCADA.....	15
b. PVSS.....	16
II- Surveillance du réseau LHC (Network Monitoring)	16
1- Le but du stage.....	16
2. DIM	18
a. Introduction.....	18
b. Les Services et commandes DIM du projet :	19
3. Points de données DP :	24
a. Introduction.....	24
b. Les types de Point de Données :.....	25
c. Configuration de l'archivage :	25
d. Control de PVSS :	26
e. Statues des Ports et Switch :	27
4. Panneaux de surveillance :	29
a. Introduction	29
b. Panneaux de configurations.....	30
c. Panneau de surveillance global	32
d. Liste de Carte pour TerasCale	32
e. Liste de ports :	33
f. Statistiques	35
5. Configuration de base de données	36
a. Introduction.....	36
b. Panneau de configuration de la base de données	37
Conclusion.....	39

INTRODUCTION

Dans le cadre du Stage de ma deuxième année d'études à l'ISIMA j'ai effectué un stage au Centre Européen de Recherche Nucléaire CERN durant une période de 5 mois commençant en début d'avril 2006.

Le sujet de stage est la configuration et la surveillance des équipements réseau, surtout des Switch, qui seront utilisés dans la nouvelle expérience LHCb au CERN.

Dans le LHCb on aura un grand nombre de Switch Ethernet à contrôler et à configurer. La supervision des Switch se fait en utilisant le protocole SNMP. Après on intègre cette supervision dans un système SCADA (PVSS).

Certains aspects de la Configuration des Switch sont disponibles uniquement en utilisant la ligne de commande (CLI). Mais vu le nombre des Switch dans l'expérience (des centaines), ces aspects doivent être automatisés à l'aide d'outils comme "Expect" et aussi intégrés dans le système de contrôle distribué du LHCb.

Dans ce stage le travail s'est fait d'une manière parallèle des deux côtés : client et serveur, avec Monsieur Artur Barczyk. J'ai intégré le groupe LHCb ONLINE dont on fait des conférences qui permettront de visualiser les démarches de travail et les nouveautés de chacun des membres et aussi pour pouvoir accueillir les critiques des autres membres.

I- CERN

1. Présentation

Le CERN est l'Organisation Européenne pour la Recherche Nucléaire, le plus grand centre de physique des particules du monde. Il est situé de part et d'autre de la frontière franco-suisse, près du petit village de Meyrin (banne lieue de Genève). L'idée de construire de tel centre venait du grand physicien français Louis de Broglie lors de la conférence européenne de la Culture en 1949. Fondé en 1954, le laboratoire a été l'une des premières entreprises communes à l'échelle européenne et il comprend maintenant 20 états membres.

Bien que le CERN soit principalement fondé pour les recherches physiques, il est similaire à un groupement de plusieurs grandes entreprises. Les services de ressources humaines, les services informatiques, service postal et les services de livraison et de sécurités contient en elles même des grandes départements similaires a des entreprises de part entière.

✓ Le but du CERN

Les particules sont extrêmement petites, et pour les voir et les étudier, les scientifiques ont besoin d'outils très spéciaux. Les accélérateurs accélèrent les particules jusque atteindre une vitesse de la lumière et les amènent à de très hauts niveaux d'énergie avant de les précipiter sur d'autres particules. Autour des points où les "collisions" se produisent, les scientifiques montent des expériences qui leur permettent d'observer et étudier ces collisions. Ces expériences utilisent des instruments, parfois gigantesques (comme le LHCb), faits de plusieurs sortes de détecteurs de particules (RITCHES, VELO ...). En accélérant les particules et en les faisant entrer en collision, les physiciens peuvent identifier leurs composants ou créer de nouvelles particules, révélant la nature des interactions agissant entre elles.

Bien que le CERN soit fait pour les recherches physiques plusieurs technologies sont développées dans ses laboratoires comme le fameux découvert d'Internet par Tim Berners-Lee et Robert Cailliaud En 1989-1990 qui ont conçu et développé un système d'information hypertexte, c'est le début du World Wid Web. La Thérapie du cancer, imagerie médicale et industrielle, techniques de gestion des radiations, électronique rapide, instrumentation de mesure, nouvelles techniques de fabrication, nouveaux matériaux, ... sont quelques autres technologies développées au CERN pour la recherche en physique des particules.



Figure 1 CERN site de Meyrin

✓ Domaine de la physique au CERN

Les scientifiques ont découvert que toute la matière de l'Univers est faite d'un petit nombre de constituants de base appelés particules élémentaires, régis par quelques forces fondamentales. Certaines de ces particules sont stables et forment la matière normale, les autres n'existent qu'une fraction de seconde et se désintègrent en particules stables. Toutes ont coexistées pour un bref instant après le Big Bang. Depuis lors, seule l'énorme concentration d'énergie qui peut être atteinte dans un accélérateur au CERN peut les faire renaître. Dans le but de comprendre la formation des étoiles, de la terre, de tout ce qui nous entoure.

✓ réussites du CERN

Le CERN a fait de nombreuses découvertes importantes pour lesquelles des scientifiques du CERN ont reçu des distinctions prestigieuses, dont notamment des Prix Nobel. La découverte la plus utile pour le public est le World Wide Web. Il fut conçu à l'origine pour améliorer et accélérer le partage des informations entre physiciens travaillant dans différentes universités et instituts aux quatre coins du monde. Actuellement le CERN procède les nouvelles technologies mondiales dans le domaine réseau en vu l'utilité de ce domaine dans le transport des données reçu dans les détecteurs de diverses expériences du CERN.

Le CERN emploie un peu moins de 3000 personnes, de diverses nationalités (plus de 80 nationalités) couvrant un large éventail de compétences et de métiers - physiciens, ingénieurs, techniciens, ouvriers qualifiés, administrateurs, secrétaires... cela sans compte la collaboration avec plus de 500 universités au monde.

2. LHC

Le grand collisionneur d'hadrons, Large Hadron Collider (LHC), est un accélérateur de particules. Elle accélérera deux faisceaux de protons ou d'ions lourds qui entreront en collisions dans de gigantesques détecteurs. Prévu pour démarrer en 2007, il permettra à terme des collisions de faisceaux de protons à une énergie de 14 TeV. Des faisceaux de noyaux de plomb seront également accélérés, entrant en collision avec une énergie de 1150 TeV. LHC est conçu pour tenter de répondre à la question : est-ce que tout sorte de matières à la même origine ?



Figure 2 Circonférence des détecteurs CERN

3. LHCB (Large Hadron Collider beauty)

LHCB est une expérience dédiée. Elle étudiera très précisément la différence de comportement entre la matière et l'antimatière dans le secteur de la beauté. L'exploration de ce domaine commence à peine, à Hambourg, à Stanford et à Tsukuba au Japon.

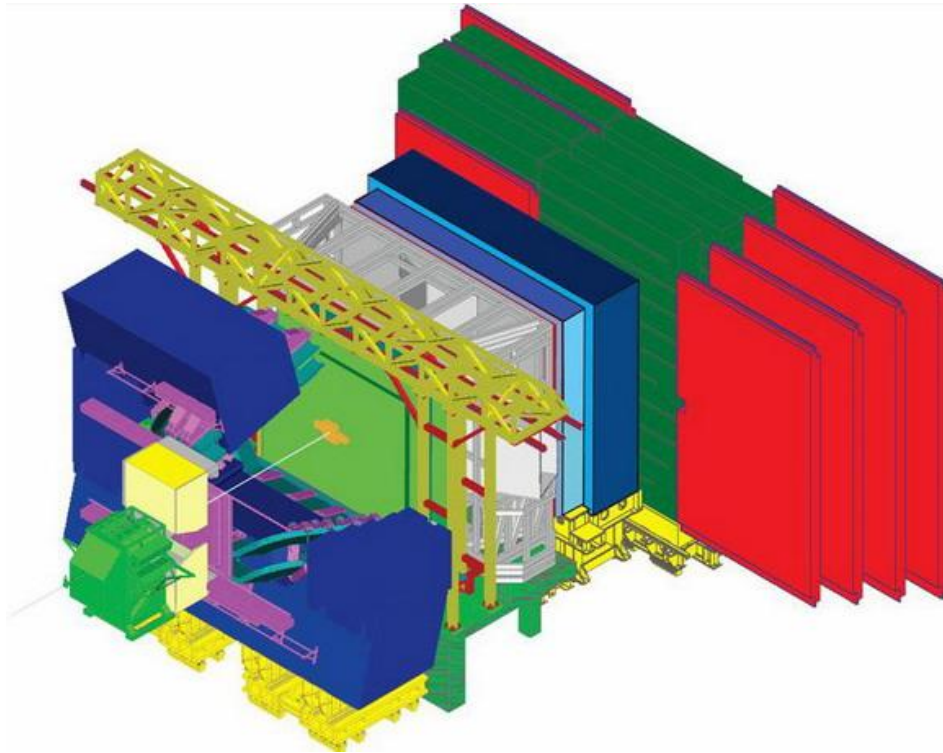


Figure 3 Structure de LHCb

✓ L'expérience

La différence de comportement entre la matière et l'antimatière joue un rôle clef dans l'évolution de l'Univers. A l'instant de sa création, il était composé de matière et d'antimatière en égale quantité. Aujourd'hui l'antimatière a disparu. Pour comprendre cette évolution, un processus traitant différemment la matière et l'antimatière a dû apparaître dans la première microseconde.

Une différence de comportement entre la matière et l'antimatière a été observée dès 1964. Il s'agissait de la découverte de la violation de CP dans le système des mésons K neutres. Mais sa grandeur est trop faible pour expliquer la disparition de l'antimatière au cours de l'évolution de l'Univers. Par ailleurs, la théorie qui décrit la physique à l'échelle subatomique et qui permet de rendre compte de cette différence, est très peu contrainte expérimentalement.

Le but de l'expérience LHCb est l'étude précise de la violation de CP dans tous les couples "particule antiparticule" contenant un quark beau. Ces particules seront produites dans les collisions proton-proton, qui auront lieu dans le futur collisionneur hadronique LHC du CERN. La grande précision de l'expérience permettra de contraindre fortement le modèle théorique et de tester sa validité. Elle pourrait aussi conduire à la découverte de nouvelles sources d'asymétrie matière-antimatière. [CPPM : Introduction à la physique de LHCb]

Le LHCb sera installé dans le point 8 de la circonférence qui représente un circuit de tunnel sous terrains (-100m) où les particules seront accélérées à une vitesse de la lumière.



Figure 4 Tunnel (accélérateur à gauche, couloir de circulation)

✓ Le réseau

Le CERN (Groupe IT/CS) a réservé huit sous réseaux pour la nouvelle expérience de LHCb : de 10.128.0.0/16 à 10.135.0.0/16 avec huit autres sous réseaux de réserves.

Deux de ces huit sous réseaux seront utilisés pour l'interface d'installation et deux autres pour les trois chambres dans les puits (D1, D2 et D3).

Le réseau LHCb se divise en deux réseaux de fonctionnement différents. Le réseau de données (DAQ) qui a pour but de transporter les données acquises par les détecteurs au disque de stockage.

L'autre réseau est pour le contrôle pour la fonctionnalité du contrôle de tous les équipements dans le réseau.

A noter bien que le routage dans l'expérience sera statique aucun protocole de routage ne sera utilisé.

Des listes de contrôle d'accès seront configurées sur les diverses interfaces avec l'adressage IP choisi et ces listes d'accès seront configurées par PVSS avec l'interface de la base de données.

Dans le puits de LHCb les divers équipements seront installés sur 3 chambres isolées dans une zone protégée des radiations. Les interconnexions seront faites par des « Patch Panel » qui sont des panneaux qui comportent des entrées femelle RJ-45 reliées par des câbles dont leur autre bout se situe sur un autre « Patch Panel » de la même structure.

Ci-dessous une image montrant la structure des trois chambres décrit :

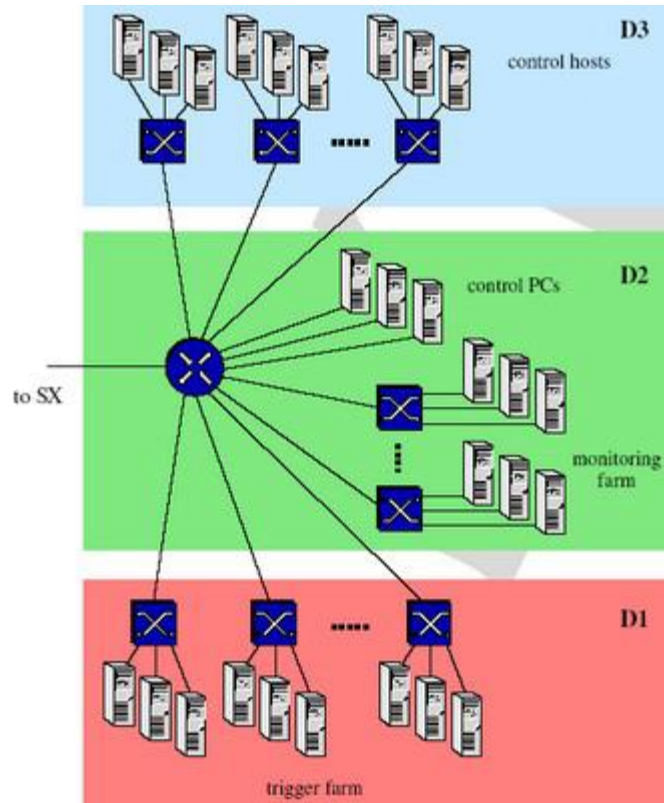


Figure 5 Chambres du réseau du DAQ dans le puits

Les liaisons entre ces chambres se font à l'aide des « Patch Panel » qui sont installés sur les racks à côté des Switch et des PC :

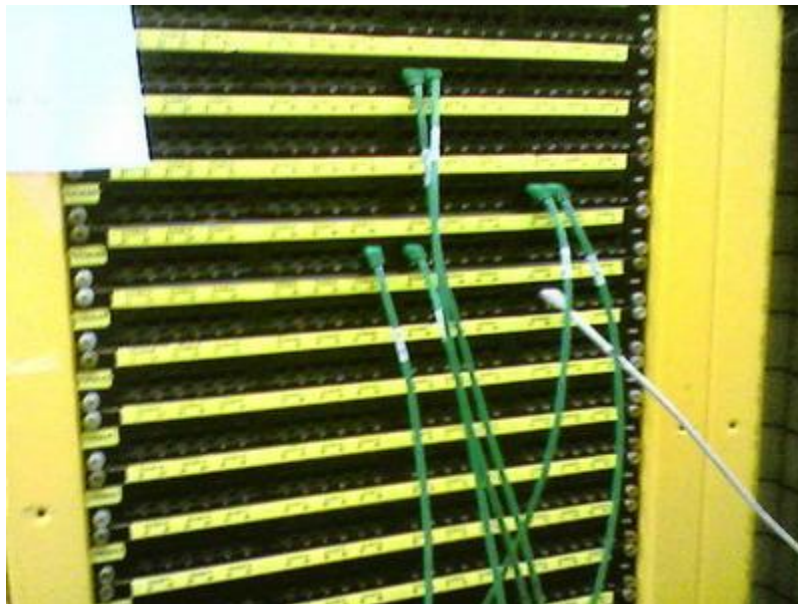


Figure 6 Exemple d'un Patch Panel

✓ Test Bâtiment 157

Pour tester toutes les applications de control et de configuration des divers systèmes un sous bâtiment a été réservé pour la simulation. Ce bâtiment contient un minimum d'équipements permettant de faire des tests réels sur les applications développer pour le LHCb.

Les images des panneaux de surveillance dans se rapport sont le résultat de test de mon projet de stage sur la topologie créée dans ce bâtiment dont voici la structure des deux réseaux :

➤ Réseau de control

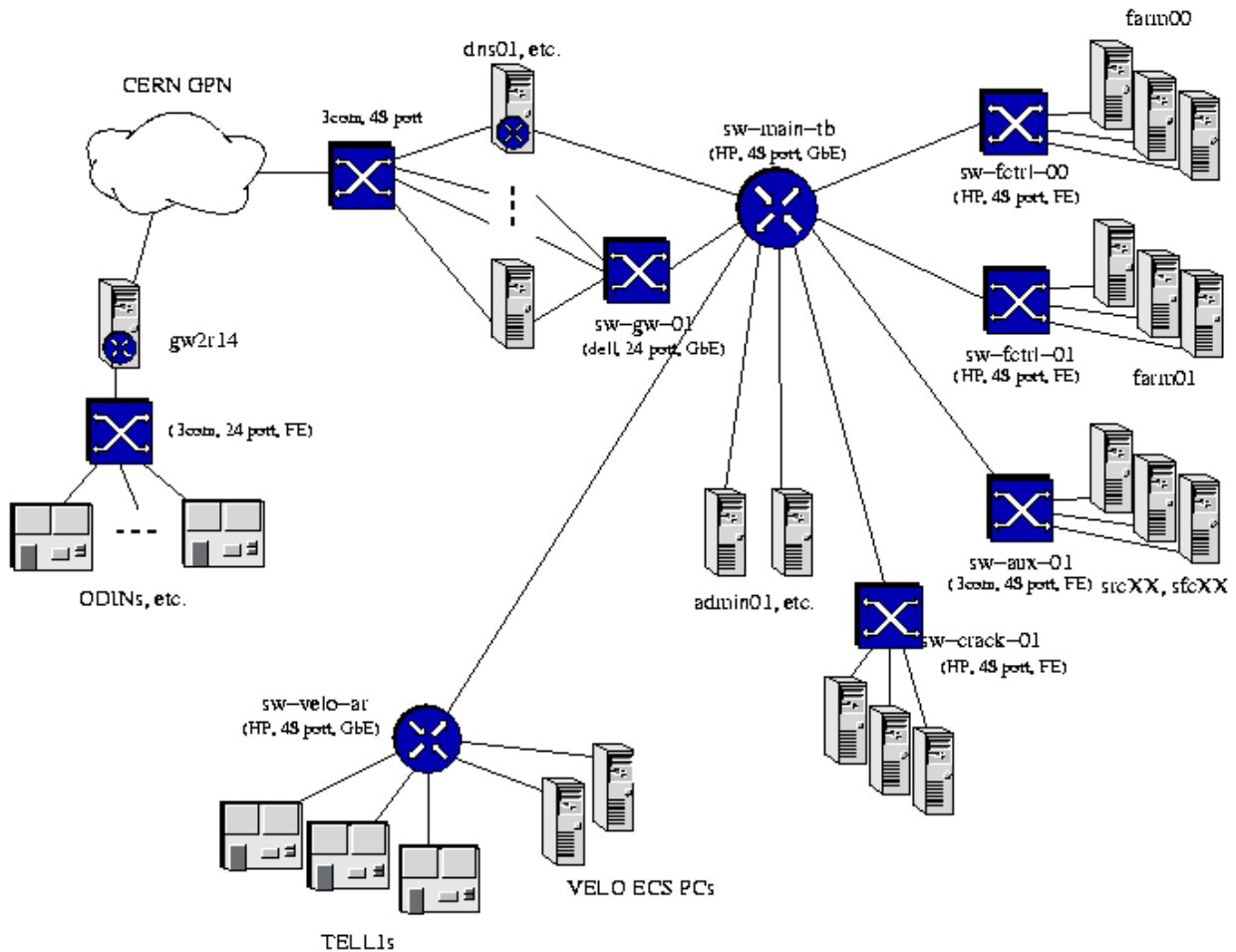


Figure 7 Topologie du réseau de control de test

➤ Réseaux de données

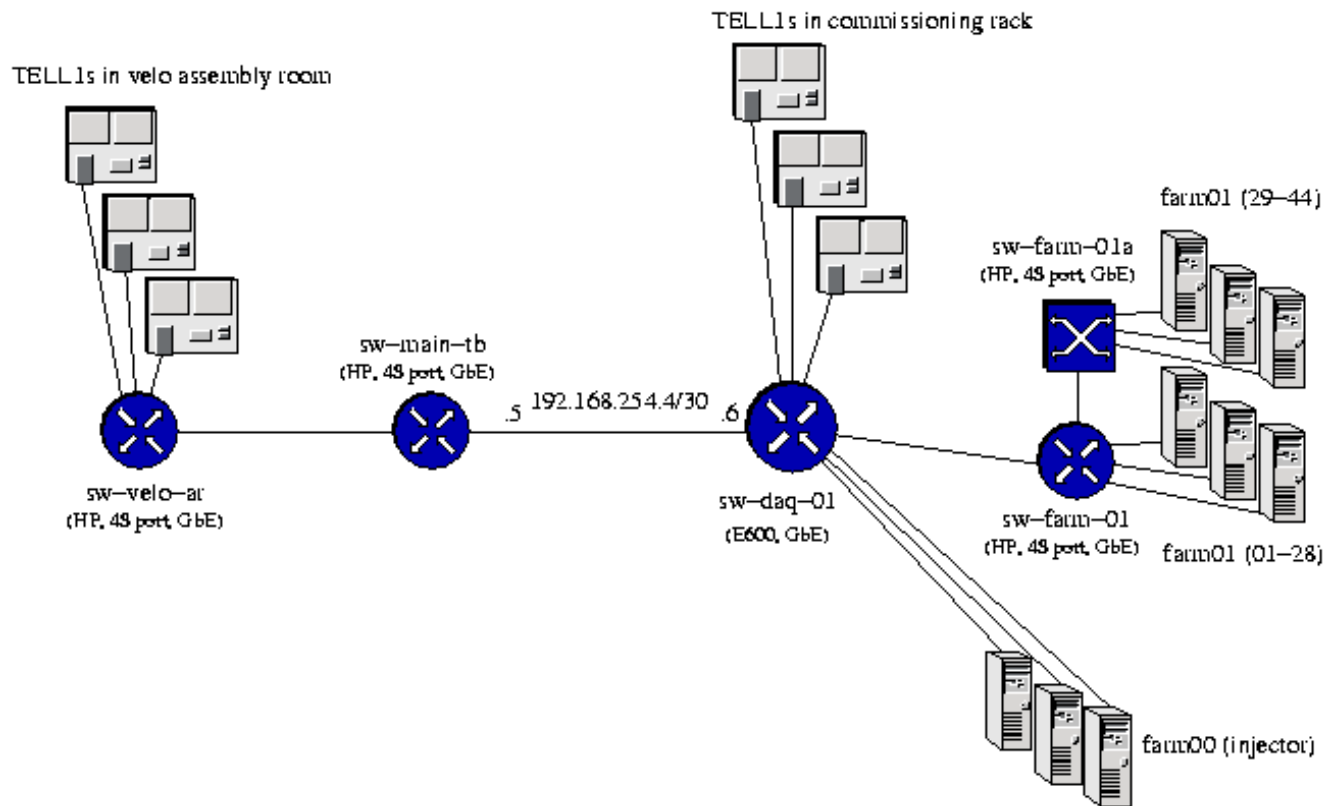


Figure 8 Topologie du réseau de données de test

4. LHCBOONLINE

Le système de LHCb en ligne (LHCBOonline) se charge de tous les aspects d'informatiques, contrôles de l'expérience, la synchronisation et le contrôle rapide.

Il prévoit (provider) l'infrastructure pour les données reçues à partir du grand niveau des triggers ainsi que le contrôle, la configuration et la surveillance de l'expérience entière.

5. Les system SCADA et PVSS :

a. SCADA

SCADA pour control et acquisition de données de surveillance (Supervisory Control And Data Acquisition).

Ce sont les systèmes logiciels commerciaux utilisés intensivement dans l'industrie pour la surveillance et le control des processus industriels.

Les systèmes SCADA sont employés dans une grande variété de domaines industriels, ils fournissent une architecture flexible, distribué et ouvert pour permettre de personnaliser un domaine d'application particulier.

En plus d'un ensemble de fonctionnalité de base de SCADA, ces systèmes fournissent également un ensemble d'interfaces standard au matériel et au logiciel aussi bien qu'une interface de programmation d'application (api) pour permettre l'intégration avec d'autres applications ou d'autres systèmes logiciels.

Les principales fonctionnalités des system SCADA peuvent être récapitulées comme suit :

- ✓ Acquisition de données
- ✓ Enregistrement de données et archivage
- ✓ Manipulation d'alarme.
- ✓ Mécanisme de contrôle d'accès
- ✓ Interface homme-machine comprenant plusieurs dispositifs standard par exemple, les histogrammes, l'affichage d'alarme

b. PVSS

PVSS (Steuerungs-Systeme d'und de Prozessvisualisierungs) pour Système de contrôle et de visualisation de processus, est un produit industriel, de type SCADA, de la compagnie autrichienne ETM. Parmi les points forts de ce produit :

- ✓ Il peut fonctionner d'une façon distribuée avec tous ces manageurs qui fonctionnent de manière distribuée.
- ✓ Il est possible d'intégrer les systèmes répartis.
- ✓ Il est conçu pour les deux plates formes (Linux et Windows) multi plate forme.
- ✓ C'est un dispositif orienté avec un concept flexible de point de donnée.
- ✓ Il a avancé des possibilités de Scripting.
- ✓ Il a un api flexible permettant l'accès à tous les dispositifs de PVSS à partir d'une application externe.

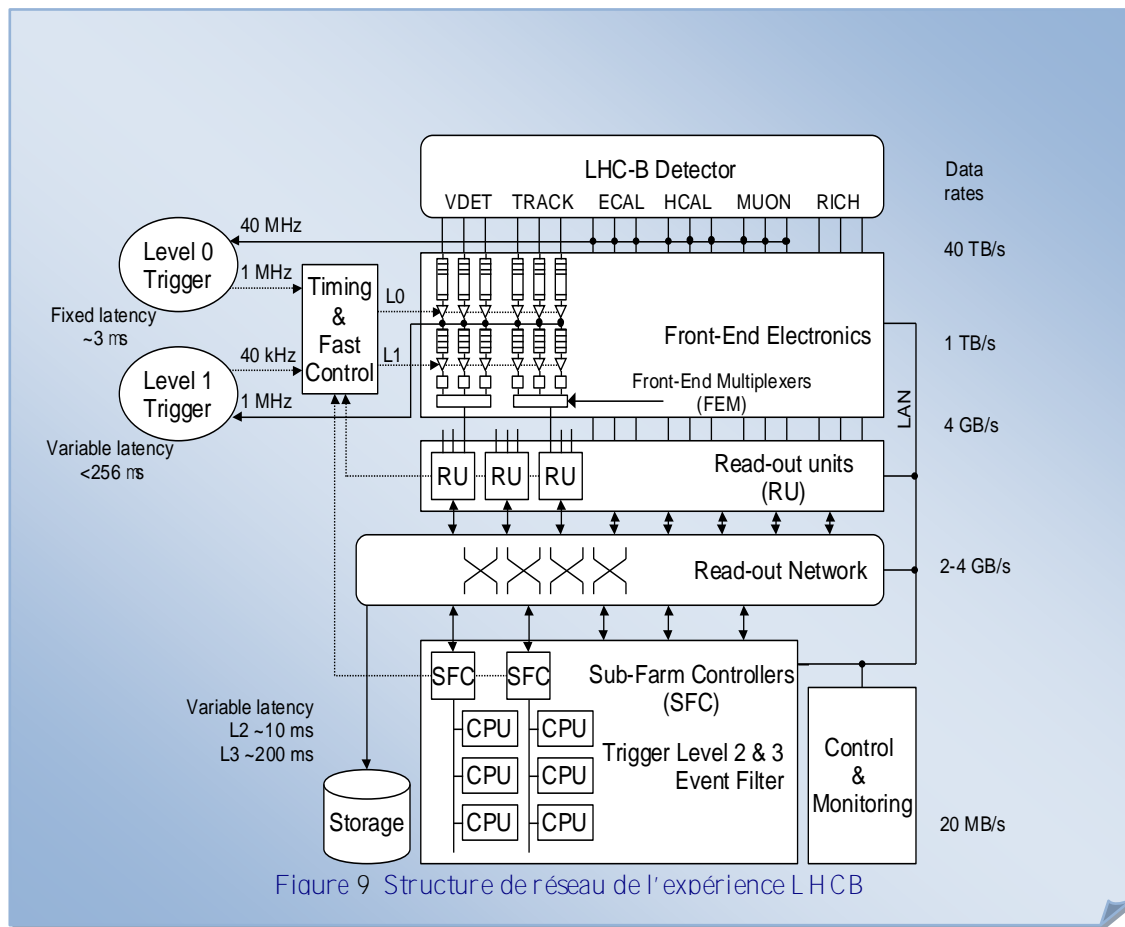
II- Surveillance du réseau LHCb (Network Monitoring)

1- Le but du stage

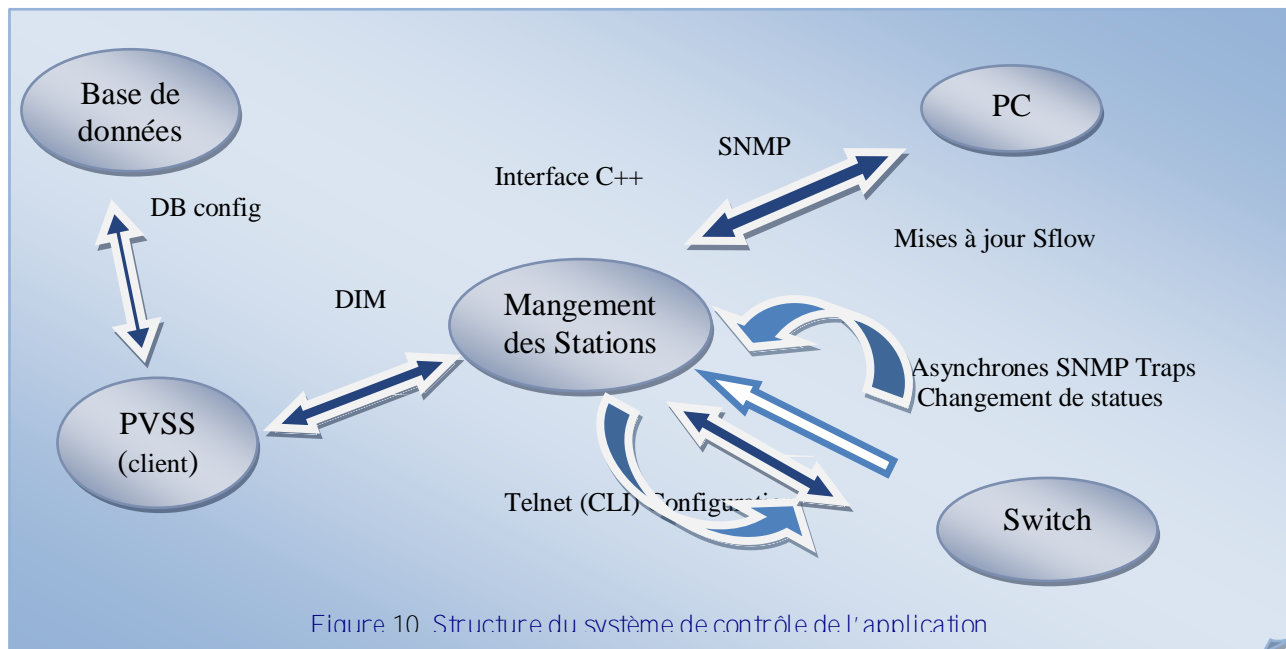
Dans la nouvelle expérience du CERN le LHCb, plus particulièrement les détecteurs du LHCb, envoient des données pour être analysées.

Ces données ne seront pas analysées en temps réel, ils seront juste redirigés, via le réseau de données, vers des disques de stockage à une connexion de haut débit (10 gigabits par seconde).

Et pour contrôler les équipements faisant cet acheminement le LHCb ONLINE a conçu un réseau séparé qui a le rôle de contrôle.



Au CERN le système de contrôle de l'expérience sera fait à l'aide des panneaux conçus en PVSS et de divers composants du Framework. Le but du stage est de créer un sous système de surveillance du réseau intégrable au sein du système de contrôle globale de LHCb. Le contrôle des équipements réseau, à dire les Switch, les routeurs ou les PC sera fait par PVSS à l'aide d'une interface en C++ utilisant SNMP et Sflow pour réagir avec divers équipements.



La liaison entre le client (le système de control PVSS) et le serveur (Interface C++) est fait par DIM. Ce composant du Framework permet d'acheminer des informations dans les deux sens (client serveur)

La communication entre le serveur et les équipements surveillés Switch et PC sera fait par SNMP, pour les mises à jour des statues des Switch, et par SFLOW pour les mises à jours. Un autre aspect de cette communication et l'utilisation de Telnet pour exécutés des commandes sur le Switch d'une façon automatique.

A la fin l'application doit utiliser la configuration de base de données du LHCb pour insérer, lire, et mettre à jour les informations sur la topologie du réseau ainsi que sur les équipements qui les constitue.

2. DIM

a. Introduction

DIM est un système de communication pour environnements distribués/mixtes, il fournit une couche transparente de communication d'interprocessus de réseau. Comme la plupart des systèmes de communication, DIM est basé sur le paradigme de client/serveur.

Le concept de base dans DIM est le concept du « service ». Les serveurs fournissent des services aux clients. Un service est normalement un ensemble de données (de n'importe quel type ou taille) et il est identifié par un nom - « des services appelés ». L'espace nommé pour des services est libre.

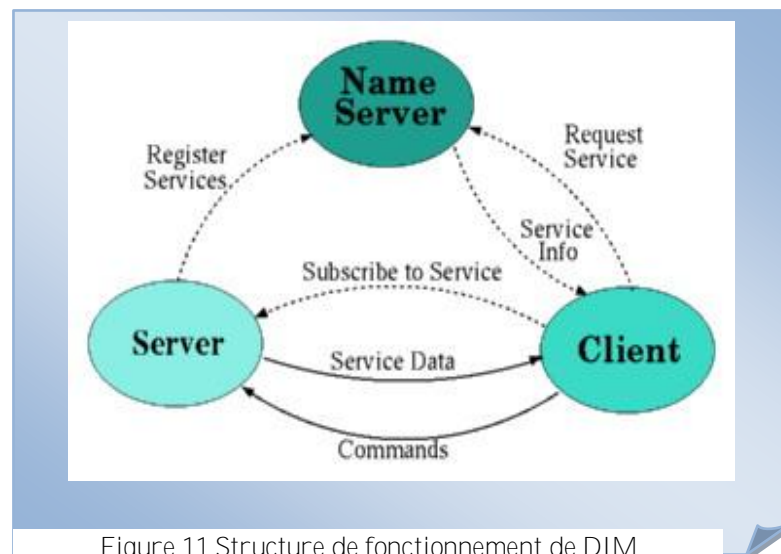


Figure 11 Structure de fonctionnement de DIM

Comment ça marche avec DIM :

Le serveur DIM publie ses services, il les enregistre au DNS NODE du system.

Le client s'informe en consultant le DNS Node sur les services et commandes disponibles et reçoit les informations désirées.

Le client s'inscrit aux services connus.

L'échange de donner (service, commandes) se fait entre le client et le serveur proposant les services et commande de DIM.

Les services DIM envoient les données aux clients déjà inscrits avec une période de mise à jour précisé par le client ou une pris en charge par le serveur.

Par contre les commandes DIM sont des commandes envoyées au serveur via DIM pour l'exécuter sur son environnement ou pour les exécuter sur les Switch désirés.

Et on nome RPC une forme mixte de commande et service. C'est une commande qui renvoi des données c'est-à-dire une commande et un service en un.

Pour recevoir où envoyer des donnes via les services et commandes de DIM, PVSS utilise ce qu'on appelle les points de données (DP : DATA POINTS). Les éléments de ces Points de Données (DPE) doivent êtres de même type que les valeurs envoyé par le service, la commande ou la RPC de DIM.

C'est DP ont une forme spécifique et elles servent comme des variables « globales » au système PVSS.

Pour inscrire un ou plusieurs services ou commandes on peut utiliser l'interface graphique qui est un Panneau du fwDIM ou d'utiliser la librairie de DIM et de créer un nouveau panneau destiné à cette opération.

En vue de la complexité de nos structures, qu'on va mentionner dans le paragraphe de Points de Données, nous avons procédé par la deuxième méthode.

b. Les Services et commandes DIM du projet :

➤ Services de Statues des ports

Les services de base du projet sont les services indiquant les Statues des ports des Switch (PortStatus).

On export un service par port et le serveur force la mise à jour du client quand un changement a eu lieu (Sflow).

Dans la première version de projet, on a conçu deux services par Switch. Un service qui export les statues administrative « activé ou désactivé par l'administrateur » des ports et un autre indiquant les statues opérationnelles des ports du Switch.

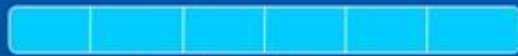
Cette démarche avait un inconvénient, en effet les statues opérationnelles des ports d'un seule Switch peuvent changer très fréquemment et dans un temps très bref, et il pourrait avoir plusieurs changements juste pour un seul port. Lorsque les Statues des autre ports restent inchangés le service envoi tout le tableau pour chaque variations.

Ce qui fait que, par exemple, en une demie seconde on peut envoyer 10 fois des tableaux dont juste une valeur change 10 fois sans que les autres changent. Avec cette méthode la bande passante sera entièrement remplie.

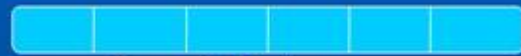
Une solution intermédiaire constitue à conserver la structure d'un service pour les Statues administratives des ports d'un Switch par Switch. Cela n'à pas trop d'influence sur la bande passante car les statues Administrative d'un port ne peut changer que si l'administrateur la fait lui-même. Donc la période de changement et très grande.

Par contre pour les Statues administratives on a choisit de faire un service par port, de cette manière on envoi que les valeurs de ports qui changent de statues puisque chaque service s'occupe des sont propre port.

Admin & Operat Status



Admin Status



Operat Status

- ❖ **Bandwidth**
- ❖ **send values that not changed**
- ❖ **Operat status change frequently**



Get One Operate Status services by port

Figure 12 1^{ère} version de la structure des services exportant les Statues des ports

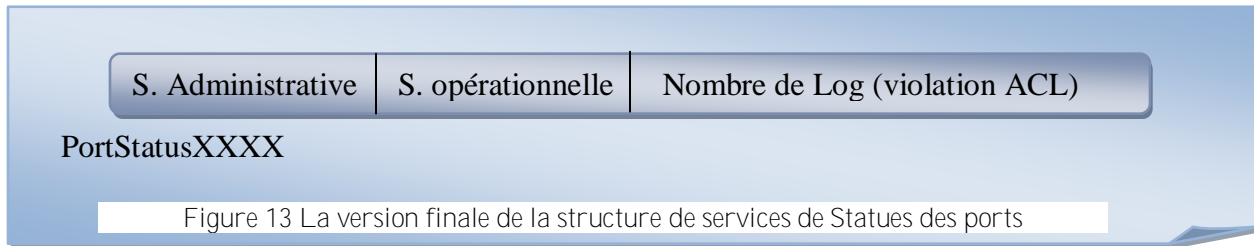
La valeur calculée par le client (PVSS) des Statues des ports utilise deux services par port. Un qui export un tableau qui contient les statues administrative de tous les ports du Switch concerné et un autre service qui export un entier indiquant la Statues opérationnelles.

La difficulté de cela et dans les scripts qu'on a crée pour les mises à jour et le calcul des Statues globales. En fait en PVSS on utilise une fonction du control qui permet de faire des opérations, de façon à appeler des fonctions lorsqu'une valeur dans un élément de point de donnée change.

Les inconvénients :

- ❖ Si un port change de SA (Statue Administrative) le service va envoyer tout les SA des ports du Switch concerné et mettre à jour l'élément de point de donnée qui est aussi un tableau.
- ❖ Lors d'une mise à jour d'un élément du point de donnée contenant les statues d'un port la fonction vérifiant les statues globales des Ports et des Switch vont exécuter les fonctions une par port pour chaque Switch. Lorsque juste une valeur change juste une exécution est nécessaire les autres sont supplémentaires puisqu'il n'y a pas de changement pour les statues des autres ports. Les fonctions de mises à jour et de calcule sont une par port. Et dans les panneaux de control chaque bouton du panneau fait appelle à ces fonctions au cas où les éléments des points de données concerner change.

La solution finale est de faire un service par port, ce service sera de la forme d'un tableau de 3 entiers. Le premier contient la statue administrative du port, le deuxième la statue opérationnelle et la troisième les nombres de violations de liste de control d'accès (ACL) configurer sur ce port via la base de données.



Ainsi lorsqu'il y a un changement de l'une de ces trois valeurs on fait une seule opération par port au lieu de faire trois de plus.

➤ Services des Informations sur les Switch et les ports

Ces services sont mis à jour périodiquement, la période de ces mises à jour est déterminée par le client au moment de l'inscription à ces services. Actuellement on utilise 20 seconds mais on peut la changer par le panneau qu'on a créé « panneau de configuration du projet ».

On distingue deux types d'informations :

- ✓ Informations qui vont être traitées en histogramme

Se sont des statistiques concernant les taux de transfert et les erreurs dans les paquets envoyés etc.

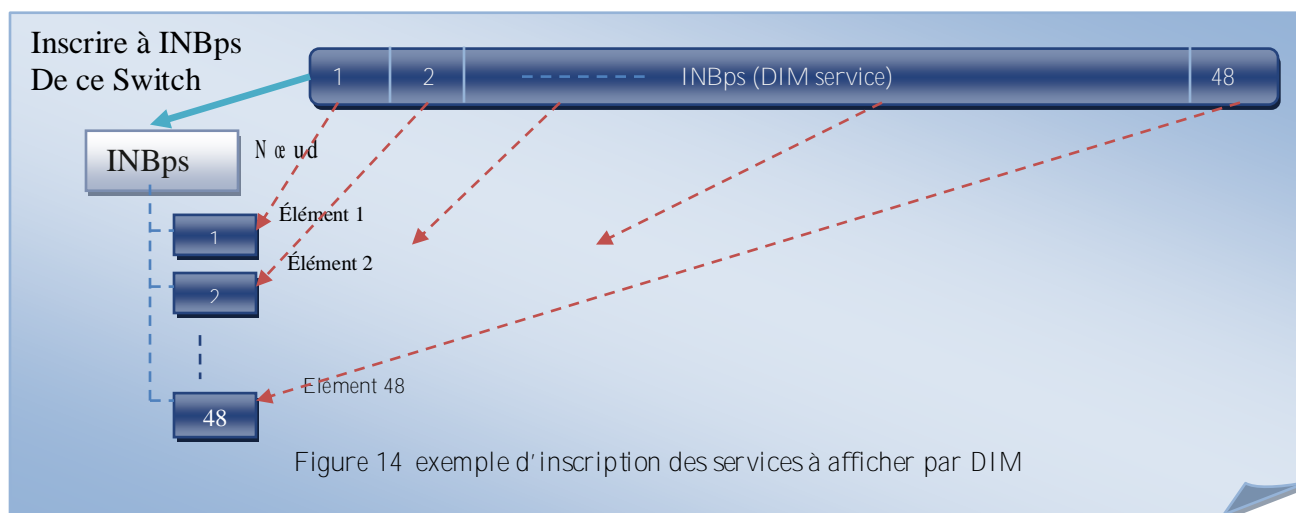
- ➔ IN(Out) Bps : nombre d'octets envoyé ou reçu par un port par seconde.
- ➔ IN(Out) Pps : nombre de paquets envoyé ou reçu par un port par seconde.
- ➔ IN(Out) Errors : nombre d'erreurs envoyé ou reçu par un port par seconde.
- ➔ IN(Out) Discards : nombre d'erreurs envoyé ou reçu par un port par seconde.
- ➔ IN(Out) BCps : nombre de broadcast envoyé ou reçu par un port.
- ➔ FCSErrors :

Ces services sont de type tableau de nombre décimal par Switch. Mais les éléments de points de données où l'on inscrit ces services sont de type structuré. En fait l'utilisation des histogrammes un pour chaque information concernant un port se pose à un problème.

Dans PVSS il fallait changer les programmes des panneaux des histogrammes « Trends » pour les adapter à accepter un index comme paramètre au lieu de prendre le nom d'un point de données qui est de type entier ou décimale pour l'afficher.

Suite à des discussions avec M. Clara GASPAR la responsable du groupe et expert DIM et PVSS on a préféré une solution souple concernant à enregistrer ce type de service dans un point de données qui est de type nœud qui contient des éléments de type entiers.

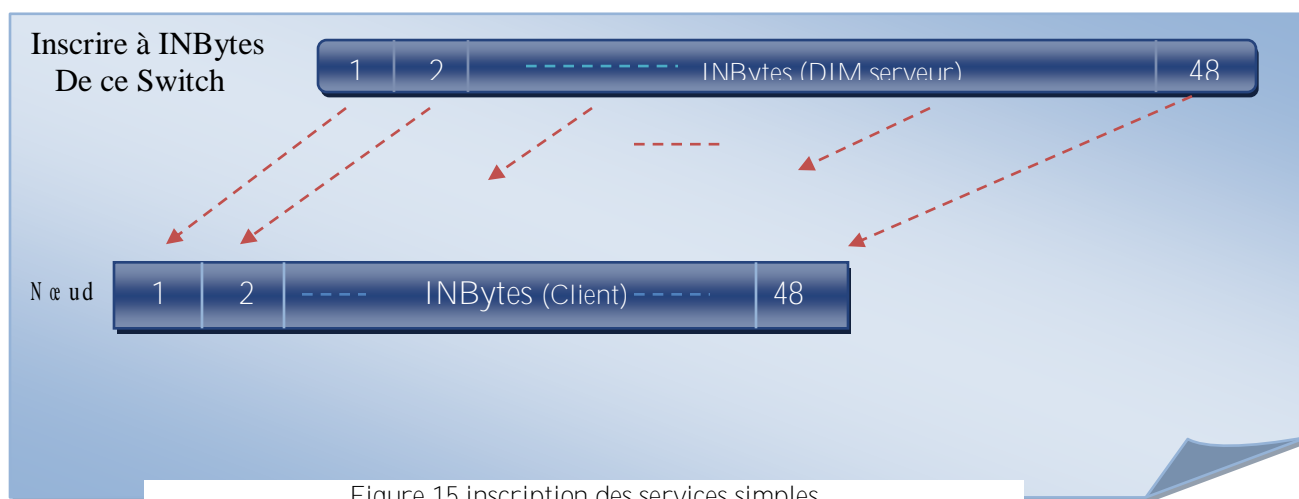
Dans l'inscription des services on cite le nom des nœuds concernés et DIM enregistre chaque case du tableau dans l'élément de même rang dans ce nœud.



✓ Informations qui vont être comme statistique en cas de besoins.

Ce sont des statistiques concernant le nombre de paquets envoyé reçu ou autres durant la période de mise à jours par chacun des ports.

- ➔ IN(Out) Bytes : Nombre d'octets envoyé ou reçu par un port, (tableau de décimale).
- ➔ IN(Out) Paquets nombre de paquets envoyé ou reçu par un port, (tableau de décimale).
- ➔ Nr PORT : Nombre de port par Switch, (entier).

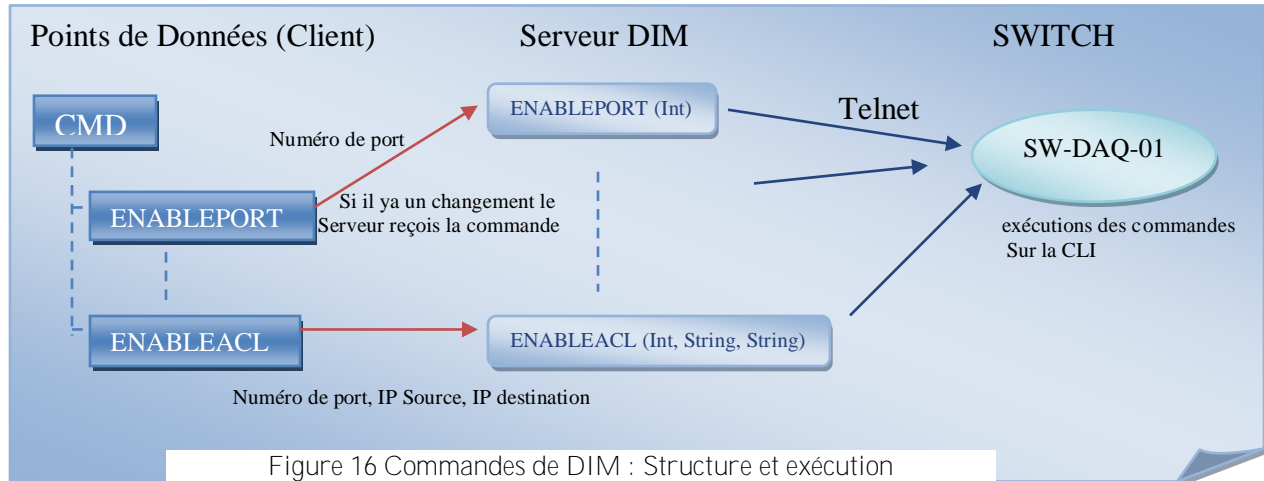


Pour ces services on les enregistre normalement dans des «éléments de points de données de type semblable aux services. C'est-à-dire que les nœuds ici ne sont pas de type structure mais de type semblable aux services.

➤ Commande DIM utilise dans le projet :

On se sert des diverses commandes pour exécuter des actions sur les Switch par l'intermédiaire du serveur qui utilise un Telnet pour passer les commandes à la CLI des Switch. Ces commandes sont :

- ➔ ENABLEPORT : activer un port. Type un entier qui le numéro de port dans le Switch.
- ➔ DESABLEPORT : désactive un port. Type entier qui le numéro de port dans le Switch.
- ➔ SAVECONFIGURATION : commande permettant de sauvegarder la configuration actuelle d'un Switch soit dans le serveur lui-même ou juste de mémoriser la configuration dans la NVRAM. Type un entier (0 : NVRAM, 1 : serveur).
- ➔ CLEARSYSLOG : Permet de mettre un RAZ aux messages de notification et d'erreur du système de surveillance.
- ➔ ENABLEACL : permet de configurer une ACL « Liste de control d'accès » sur un port. Type une structure : un entier pour le numéro de port et deux chaînes de caractères pour les IP de source et de destination avec leur masque (sous forme de \32).
- ➔ DESABLEACL : efface une ACL déjà configurée sur un port. Type un entier envoyant l'identifiant de l'ACL sur ce port.
- ➔ CLEARACLLOG : permet de mettre un raz sur les messages reçu lorsqu'il ya des violations des l'ACL configurées sur le Switch.
- ➔ ENABLEGLOBALACL : permet de configurer des ACL sur tout le Switch en utilisant les informations déjà stockées sur la base de données.



➤ RPC utilisé dans le projet :

Une RPC est une structure qui contient un service ou des services et une ou des commandes à la fois.

Dans ce type le client envoie une commande au serveur et reçoit l'appel de l'exécution de cette commande par un « service ».

On utilise des RPCS pour les Commandes suivantes :

- ➔ READCONFIGURATION : On demande de lire la configuration actuelle (NVRAM) du Switch, pour faire appel il faut envoyer une valeur comme sortie du côté du client et entrée de la côté de serveur. Ainsi le serveur va exécuter la commande et envoyer le résultat comme sortie au niveau du serveur et entrée au niveau du client pour la RPC en question.
- ➔ READACLLOG : on récupère le message de (log) d'une ACL en indiquant le numéro du message que les services des statues des ports l'envoi en 3^{ème} case. La valeur reçue est une chaîne de caractères contenant le message de violation désirée.
- ➔ READSYSLOG : on envoi une valeur pour recevoir le message du système.
- ➔ GETSYSOS : on envoi une commande pour recevoir les noms des images des dernières versions du système d'exploitation du Switch (OS). Ces versions sont stockées sur le serveur.

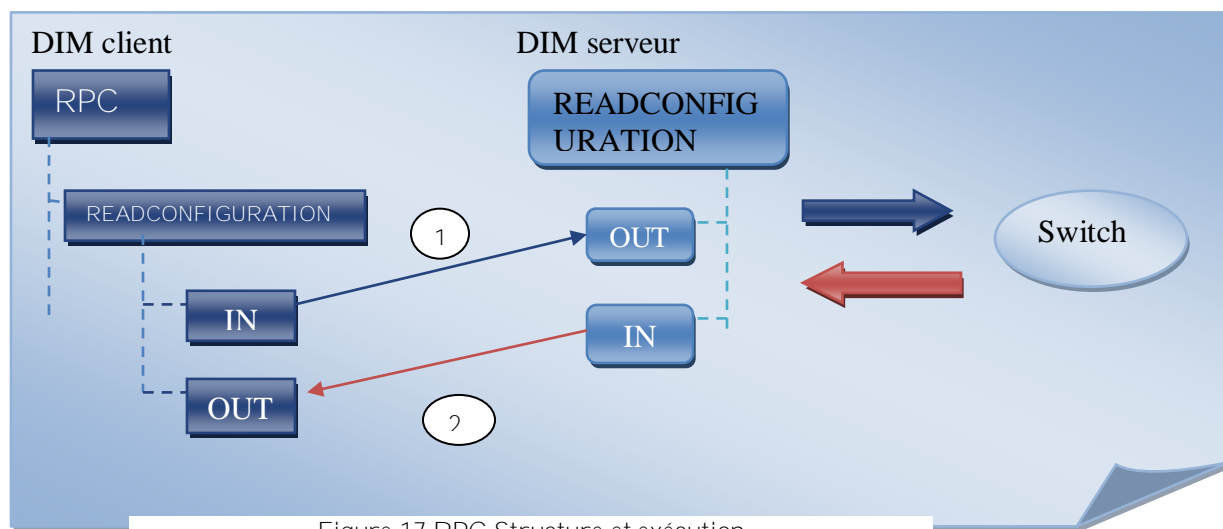


Figure 17 RPC Structure et exécution

3. Points de données DP :

a. Introduction

Les points de données servent à regrouper des données dans ces éléments (DPE) appelés élément de points de données, qui sont de structure, de type et de taille différentes. On peut les simuler à des variables globales qui assurent la mémorisation des données pour un projet entier en PVSS.

Dans le projet de surveillance on utilise deux types de DP en vue de la mise à jour des valeurs de ces éléments.

- ❖ Mise à jour par DIM (Serveur) : Ce sont les Points de Données où on a inscrit les divers services, commandes et RPC de DIM. Ces DP (Points de Données) sont mis à jour périodiquement (20 second : au choix du client) ou forcé en cas de changement de statues des équipements.
- ❖ Mise à jour par les scripts du projet (Client) : Ces Points de données servent comme variables globales commun pour tout le projet PVSS, ce qui permet de mémoriser les diverses statues et informations calculés à partir des Points de Données du premier type.

Dans notre projet on a conçu la hiérarchie suivante :

- ❖ « Racine » DP qui est de type Structure, on a une par équipement « Switch ».

- ❖ Élément de point de données DPE de type simple. Ces des type entier, tableau d'entier dynamique ou tableau de décimaux dynamique.
- ❖ de point de données DPE de type complexe. Ces DPE s'adaptent à la structure des services publiés et au besoin de surveillance. Dans ces types on fait différences entre des DPE dont les données vont être utilisées pour afficher des statistiques sous forme d'histogrammes et d'autres qui serviront pour accueillir les divers services et envoyer les diverses commandes utilisées dans le projet.

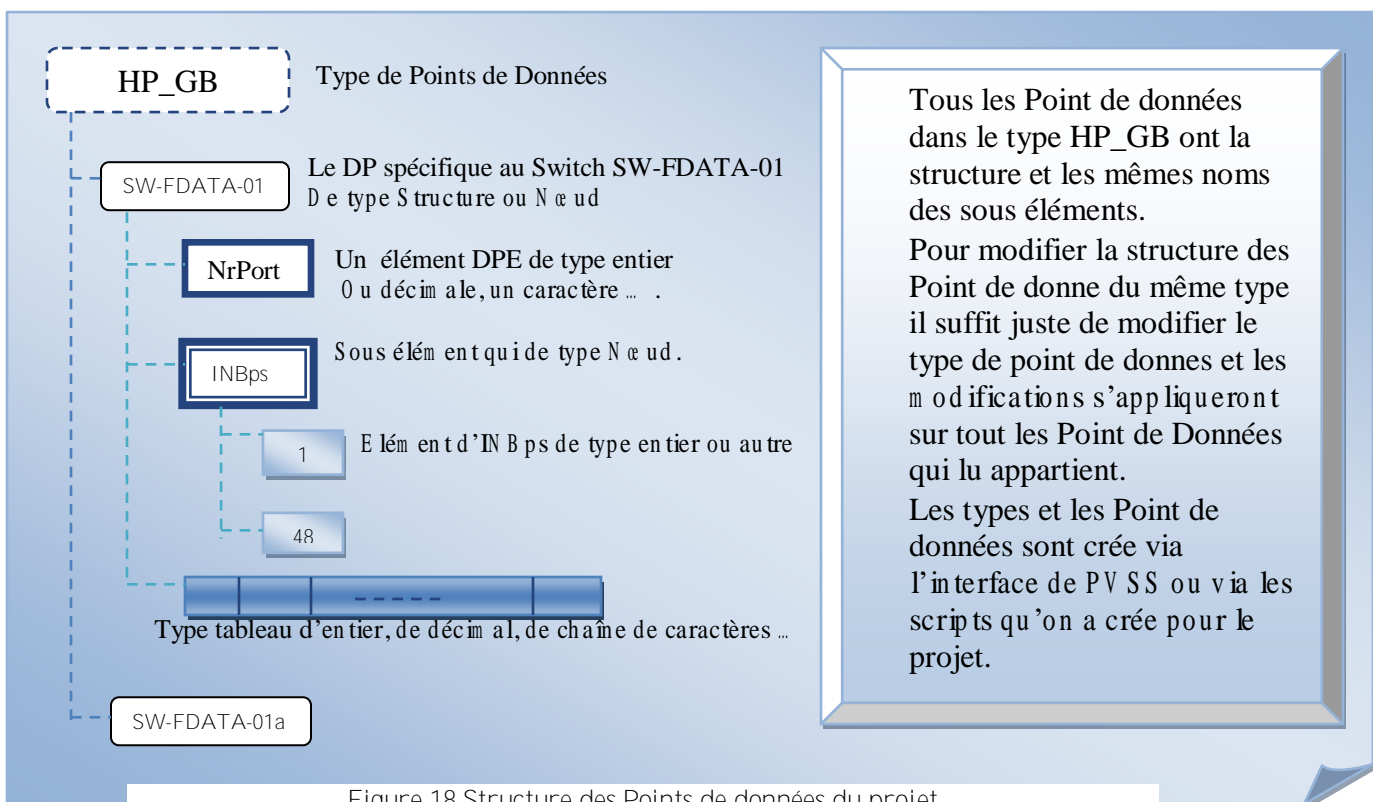
b. Les types de Point de Données :

A fin de crée une ou des points de données il est nécessaire d'avoir un type pour ces Points.

Le type est la définition de la structure que doit prendre une « Point de Donnée » et les noms de ces sous éléments ainsi que leurs types et structures.

On peut crée plusieurs Point de Données de même type par exemple dans notre projet on aura plus de 400 Switch et pour ces Switch on a crée un Type HP_GB qui spécifie la structure des DP de ces Switch.

La création des Type de Points de Données (DPT), ainsi que la création des points de données, se font par le panneau « para » du PVSS. Mais vu la complexité du projet et le nombre des DP que va contenir un type on a procédé par la création d'un nouveau panneau de configuration dans le quel on crée les types et les points de données automatiquement au demande du client.



c. Configuration de l'archivage :

On utilise la configuration de l'archivage de PVSS pour sauvegarder les statistiques et les statues de divers équipements pour une durée déterminée.

La configuration manuel est difficile en vu le nombre de points de données qu'on a dans le projet (dans l'ordre de 20000 pour le teste ou on utilise 10 Switch).

La solution adaptée a ce problème est la création d'un outil dans le panneau de configuration du projet qui permet de faire une configuration automatique pour tous les Points de Données du projet tout en permettant de choisir la classe de l'archivage ainsi que la durée d'archivassions.

L'archive est très utile surtout dans les statistiques des ports (taux de transfert, erreurs de transmission etc....). Lorsqu'il y aura des problèmes, l'administrateur ou le surveillant, peuvent bien avoir l'idée quand il a eu lieu ou ça peut les aidés à trouver les sources des problèmes.

Par exemple s'il y a un problème de broadcaste avec les histogrammes on peu déterminer la source du problème et en désactivant par exemple le port on peut résoudre le problème ou essayer de l'éviter dans le future sur ce port la.

d. Control de PVSS :

La partie Control de PVSS est la partie des fonctions permettant de programmer les panneaux. La syntaxe et la méthodologie sont similaires au C++ et C.

Il y a deux méthodes pour faire :

La première méthode consiste à écrire les fonctions et les scripts directement dans les panneaux et ces différents objets et les exécutions de ces programmes seront en même temps que l'exécution du panneau ou selon une action que l'utilisateur fait. Cette méthode à utiliser que pour exécuter des fonctions agissant sur les objets du panneau où on écrit ces scripts. Pour les scripts agissant sur les Points de Données et qui n'ont pas de relation avec les objets des panneaux c'est préférable de mettre comme manager de control. En effet Par cette méthode le script sera exécuter le nombre de fois que les utilisateurs ouvrent le panneau. Surtout dans le cas ou ces utilisateur ouvriront le même panneau en même temps le script sera exécuter le nombre de fois que le panneau sera exécuté. Mais pour la deuxième les scripts seront exécutés une seule fois même si les utilisateurs ouvriront le même panneau à la fois.

La deuxième méthode consiste à écrire dans un fichier avec une extension *.CTL et le mettre comme un API manager de control dans le consol du projet. Ainsi le script en question sera exécuté une seule fois et on règle le problème de la première méthode concernant les scripts agissant sur les Points de Données. De cette manière le script s'exécutera en arrière plan.

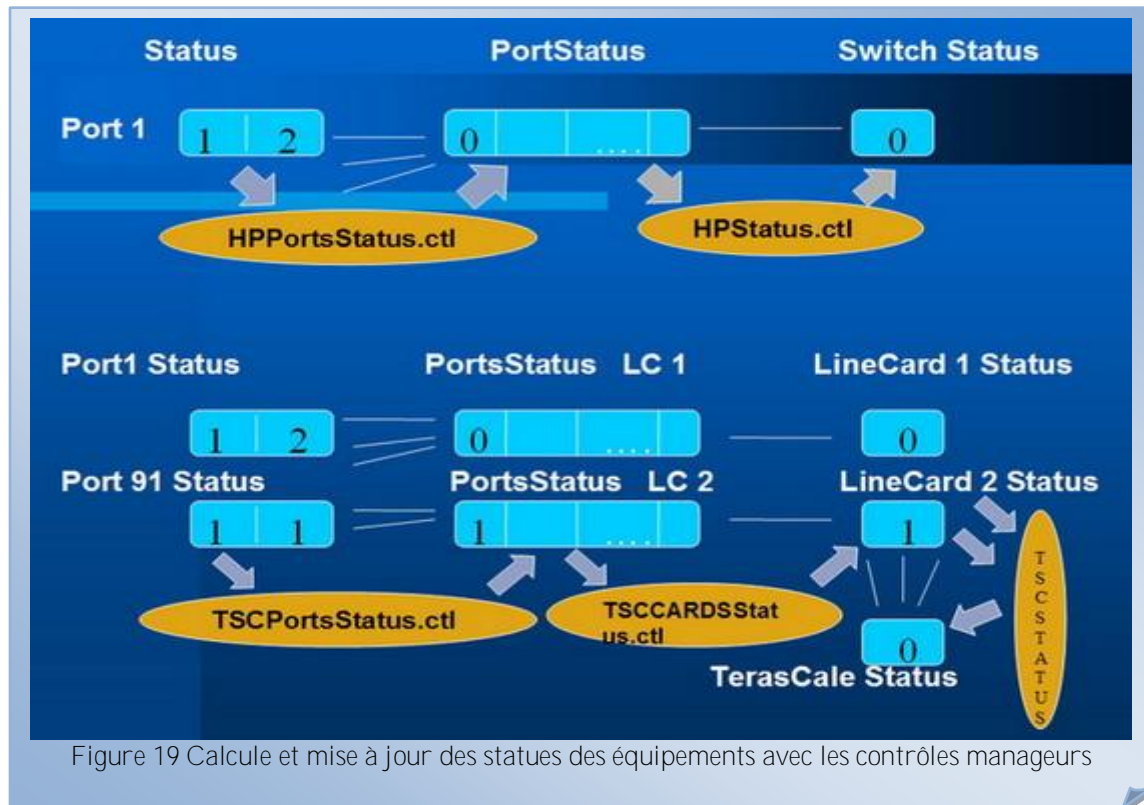
On utilise 5 script différent pour mettre à jour les Points de données qui sont créés pour contenir la valeur instantanée des statues des ports et des Switch et dans le cas du grande Switch les statues des Cartes.

Pour calculer les statues des ports on utilise un script « **HPPortsStatus.clt** » qui permet de calculer les statues courantes des ports en utilisant les valeurs de la statue administrative, de la statue opérationnelle et le nombre de violation et met les statues calculées de chacun des ports du même Switch dans un élément de points de données de forme d'un Tableau d'entiers.

Puis un autre script « **HPSwitchStatus.ctl** » qui à partir du dernier tableau mis à jour par le premier, calcul la statues globale. Les fonctions de ces scripts s'exécutent lorsque les Points de Données concernées changent de valeur.

Cette structure est utilisée pour tous les Switch de type HP. Pour le grand Switch TerasCale qui contient 1260 ports on considère une carte « LINE CARD » comme un Switch de type HP.

Par cette procédure on a 3 scripts. Un « **TSCPortStatus.ctl** » pour calculer les statues des ports de chaque carte et les mettre dans une DPE sous forme d'un tableau de 90 éléments (le nombre de ports par Carte) et un autre « **TSCCardStatus.ctl** » pour calculer les statues des Carte à partir de ces tableaux. Et on ajoute un troisième « **TSCSwitchStatus.ctl** » qui, à partir des statues des Cartes, calcule et mis à jour l'élément de DP contenant les statues globales du Switch.



e. Statues des Ports et Switch :

Les services « PortStatus » exportent des tableaux de trois valeurs. La 1^{ère} valeur est la statue administrative, la 2^{ème} est la statue opérationnelle et en 3^{ème} le nombre de violations des ACL et 0 si il n'y a pas de violations.

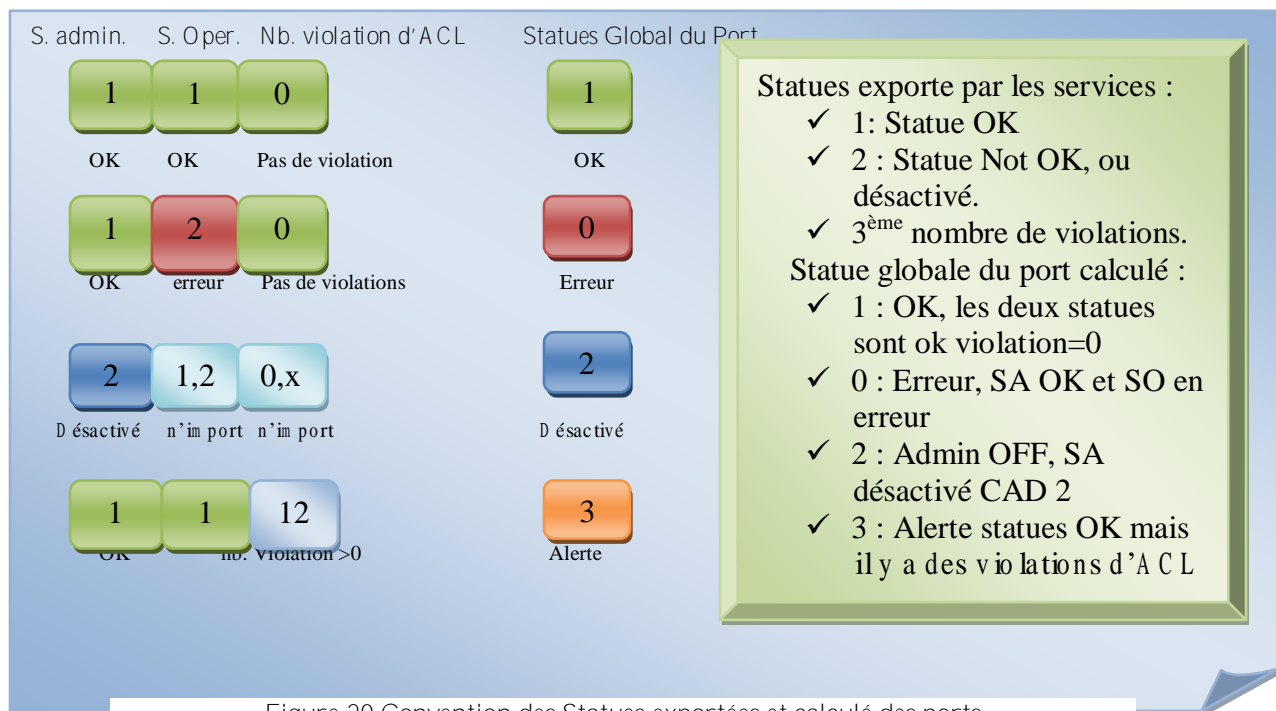


Figure 20 Convention des Statues exportées et calculé des ports

Pour calculer les statues des Switch le script manager met à jour un tableau qui contient tous les statues des ports d'un Switch ou d'une carte du cas du TerasCale.

- **Statue OK** : Si tous les ports sont en statues OK ou désactivés.
- **Statues Erreur** : Si au moins un port est en statue erreur.
- **Statues d'alerte** : si tous les ports sont en statues OK et au moins un port est en statue d'alerte.

Et pour calculer la statue du grand Switch on le considère comme un Switch HP qui a 14 ports et ces ports sont les cartes qu'on a déjà calculé leurs statues.

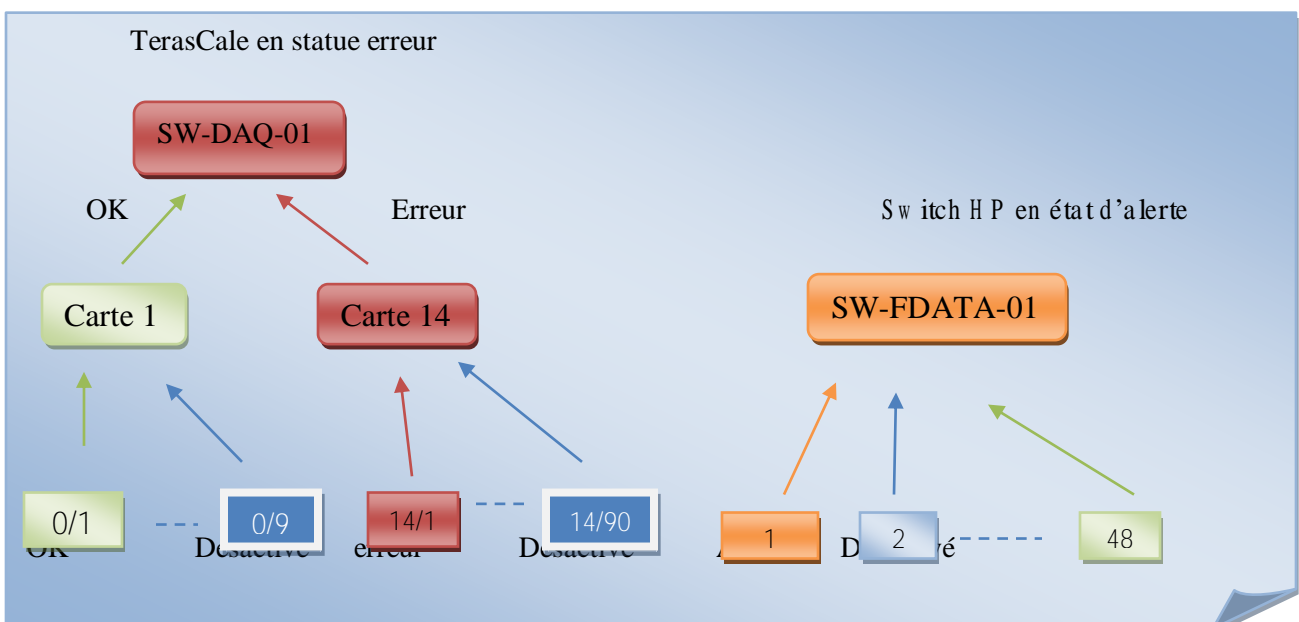


Figure 21 Statues des Switch : calculé et conventions

NB : les conventions de couleur sont les mêmes que ceux suggérées par le Framework de PVSS :

- ➔ Vert : Statues OK
- ➔ Rouge : Erreur
- ➔ Bleu : OK physique.
- ➔ Orange : Alerte.

4. Panneaux de surveillance :

a. Introduction

Les panneaux de surveillance sont conçus d'une manière hiérarchique permettant la facilité de gestion et de surveillance du réseau.

Le panneau qui sera vu dans le système de control est le panneau de surveillance globale. Ce panneau contiendra les divers Switch sous forme de boutons colorés avec des couleurs (avec les conventions de couleurs) signalant les statues de ces Switch. Pour visualiser les listes des ports de chacun de ces Switch, pour les HP ont click sur le Switch concerné et un nouveau panneau fils s'ouvriras avec une liste de boutons colorés accompagner d'autres boutons d'activation/d'désactivation de chacun de ses ports. En plus de ces boutons d'autres qui sont pour la configuration (lecture et stockage) et les Listes de control d'accès (configuration / suppression). Pour voir les statistiques et les détails d'un port il suffit de cliquer sur le bouton du port concerner et un autre nouveau panneau (fils) s'ouvrira où on trouve les statues des ports et aussi les histogrammes des valeurs nécessaires (INBps, INPps etc.).

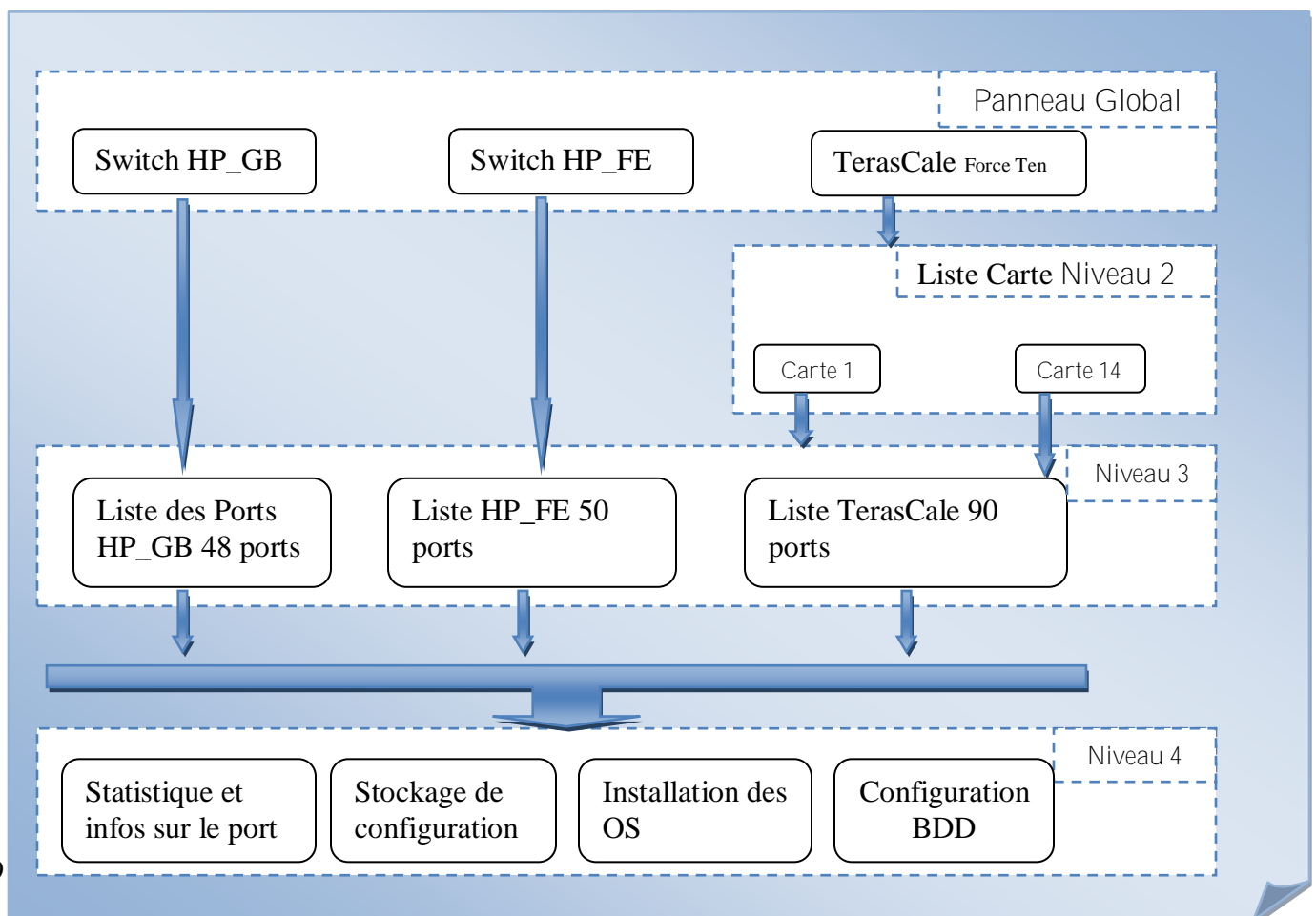


Figure 22 Structure hiérarchique des panneaux de surveillance

Une petite différence entre les Switch HP de type Gigabit Ethernet et Fast Ethernet est qu'on utilise deux panneaux différents pour la liste des ports de chacun d'eux, car le 1^{er} a 48 ports et le 2^{ème} 50 ports.

Une exception pour le grand Switch c'est qu'on va utiliser un panneau intermédiaire entre le panneau global et la liste des ports. En fait, mettre 1250 boutons dans un seul panneau est impossible d'où la nécessité et l'utilité de structurer le Switch logiquement d'une manière à afficher un panneau contenant la liste des Cartes avec leurs statuts et en cliquant sur une carte on peut voir un panneau similaire à celui des Switch de types HP mais avec 90 ports.

b. Panneaux de configurations

La structure complexe du système et des types qu'on utilise ainsi le nombre considérable de Points de Données à créer et à configurer (types, archivage..) et l'inscription de ces DP aux divers Services, commandes ou RPC de DIM nous a poussé à créer ce panneau qui contient des boutons permettant l'automatisation de ces actions citées qui posent des problèmes.

L'utilité de ce panneau est de pouvoir réinstaller le projet sur n'importe quelle machine sans avoir besoin de recourir au copiage de projet de PVSS. En fait, la copie du répertoire contenant les panneaux et des scripts « manager de control » suffit pour installer ce projet sur une autre machine.

Un autre exploit de ce panneau est de résoudre les problèmes de copiage de projet entre un projet conçu sur une machine sous Windows et une autre sous Linux.

L'utilisation du Framework dans le projet s'impose comme un obstacle pour l'utilisation de la fonction de copiage habituelle des projets de PVSS à cause des incompatibilités des versions Framework Windows et Linux. On a testé ce panneau pour créer à nouveau nos éléments et le résultat était satisfaisant.

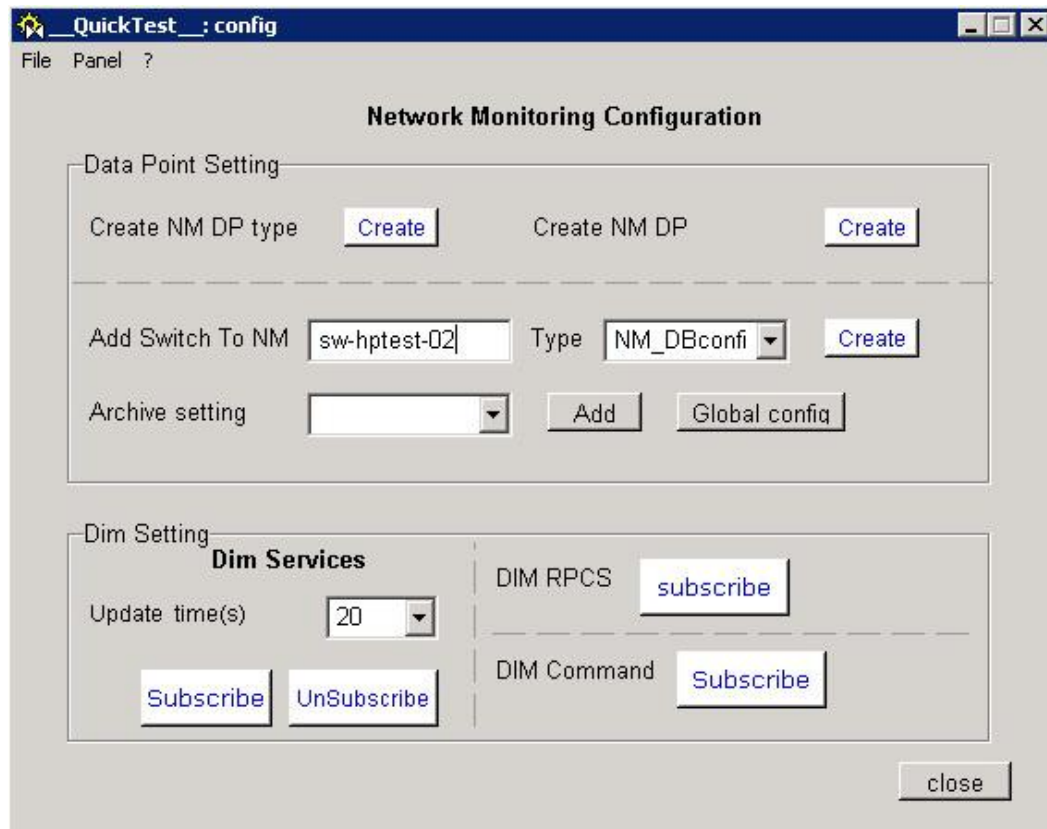


Figure 23 Panneau de configurations

Aussi ce panneau est la porte à utiliser lorsqu'on désire intégrer un nouveau Switch dans la surveillance. La dernière version permet à partir de l'ajout du nom du Switch de créer les points de données nécessaires pour cet élément, inscrire les divers services, commandes, RPC, configurer l'archivage et insérer les informations de ce Switch dans la base de données.

Quand à l'ajout du bouton représentant le Switch dans le panneau global on a crée un petit panneau (TemplateSwitch) qui contient un bouton paramétré dont l'insertion se fait par PVSS (ajouter une référence) et le bouton s'insère comme un objet PVSS existant.

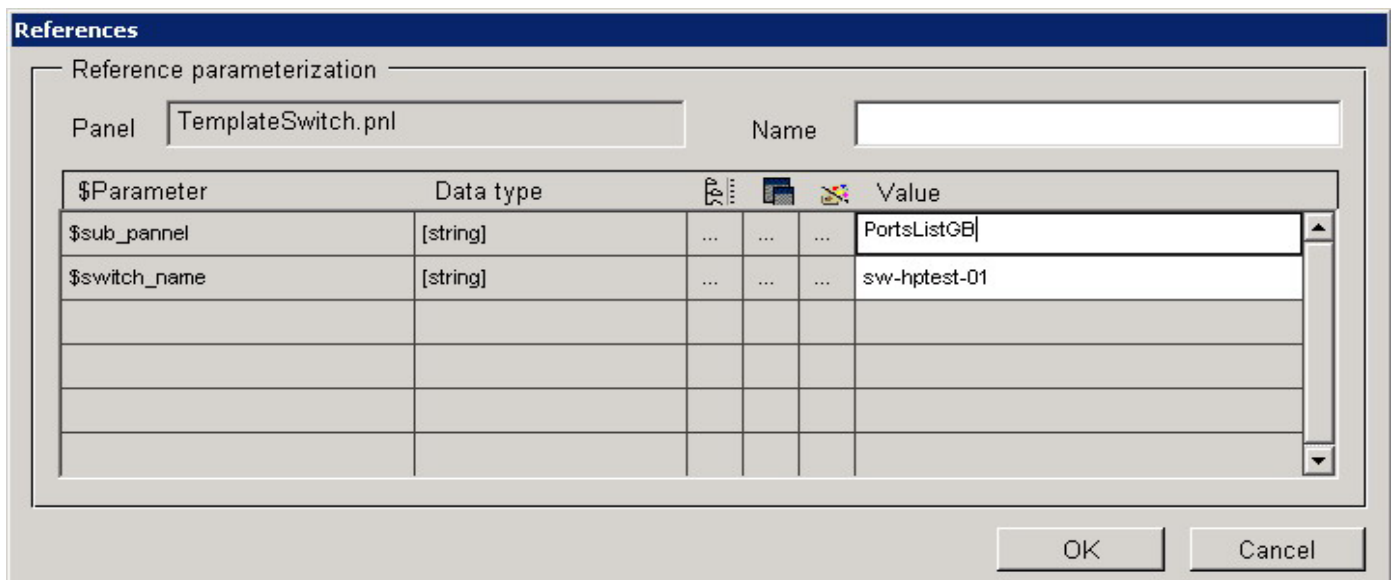


Figure 24 Panneau d'ajout de référence PVSS

L'utilisateur indique uniquement le nom de Switch et son type et ça sera installé dans l'application.

c. Panneau de surveillance globale

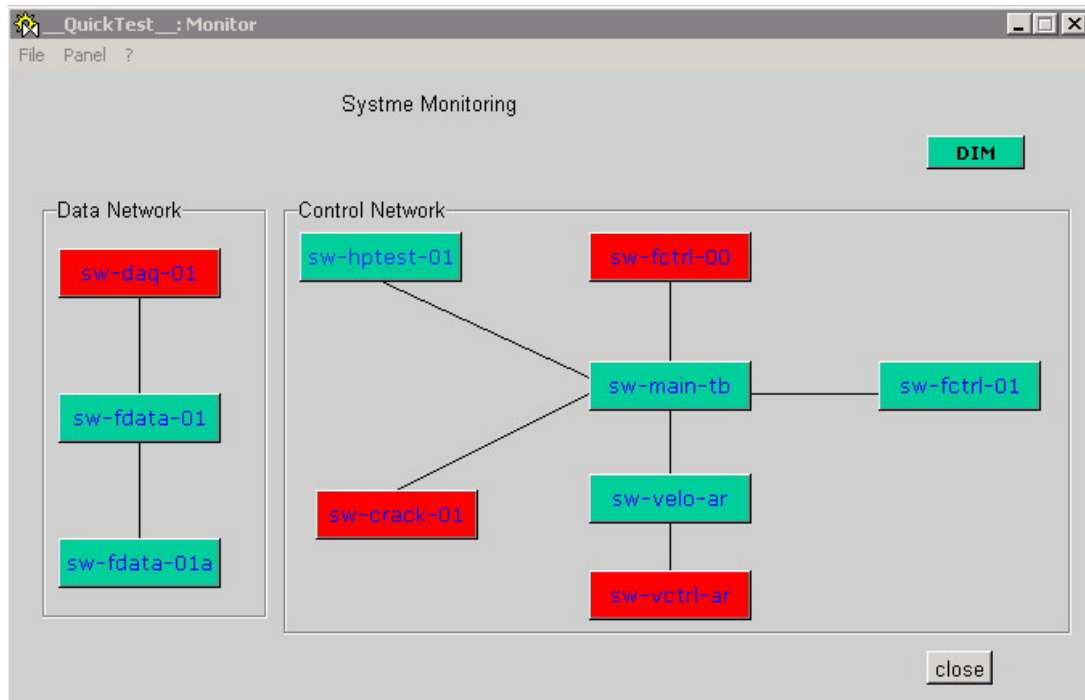


Figure 25 Panneau de surveillance globale

Le corps du projet de surveillance. Ce panneau contient la liste des Switch sous forme de boutons permettant d'accéder aux autres niveaux de panneaux avec des couleurs montrant les diverses statuts de ces équipements. La position des Switch est sous forme de dessin représentant la topologie du réseau (control ou de données).

Dans le teste on a environ 10 Switch.

d. Liste de Carte pour TerasCale

Ce panneau expose la liste des Carte du Switch type TerasCale avec leurs Statues.



Figure 26 Liste des Cartes d'un Switch de type TerasCale

e. Liste de ports :

La seule différence entre les panneaux des 3 types de Switch est le nombre de ports (boutons) par panneau.

✓ TerasCale

Pour les Switch TerasCale on a un panneau un peu grand par rapport aux autres. En fait ce panneau contient 270 boutons de façon à avoir trois boutons par port dont l'un est pour la statue du port et permet de rediriger l'utilisateur vers le panneau fils (panneau de statistiques), un deuxième pour activer ou désactiver le port concerné en cas de besoin et enfin un troisième pour configurer ou effacer la Liste de contrôle d'accès sur ce port (ACL) en utilisant les données déjà insérées dans la table Ports de la configuration de base de données de LHCB. On ajoute d'autres boutons dont l'utilité est de sauvegarder la configuration, installer l'OS du Switch, effacer les messages de log du système ou des ACL et aussi d'avoir accès à la configuration de base de données qui est fait par l'ajout d'autres panneaux dont je vais parler à la fin.

La dernière version permet d'utiliser un panneau générique (TemplatePort). Ce panneau contient les 3 boutons avec les scripts nécessaires et il est paramétré et a comme paramètre le nombre de Carte et le nombre de bouton. Sachant que le nombre réel de port sera calculé par la formule :

$$\text{Numéro de port} = (\text{Numéro de carte} - 1) * 90 + \text{numéro de bouton}$$

Les numéros de carte sont entre 1 et 14 ce qui vérifie la formule (port de 1 à 1260).

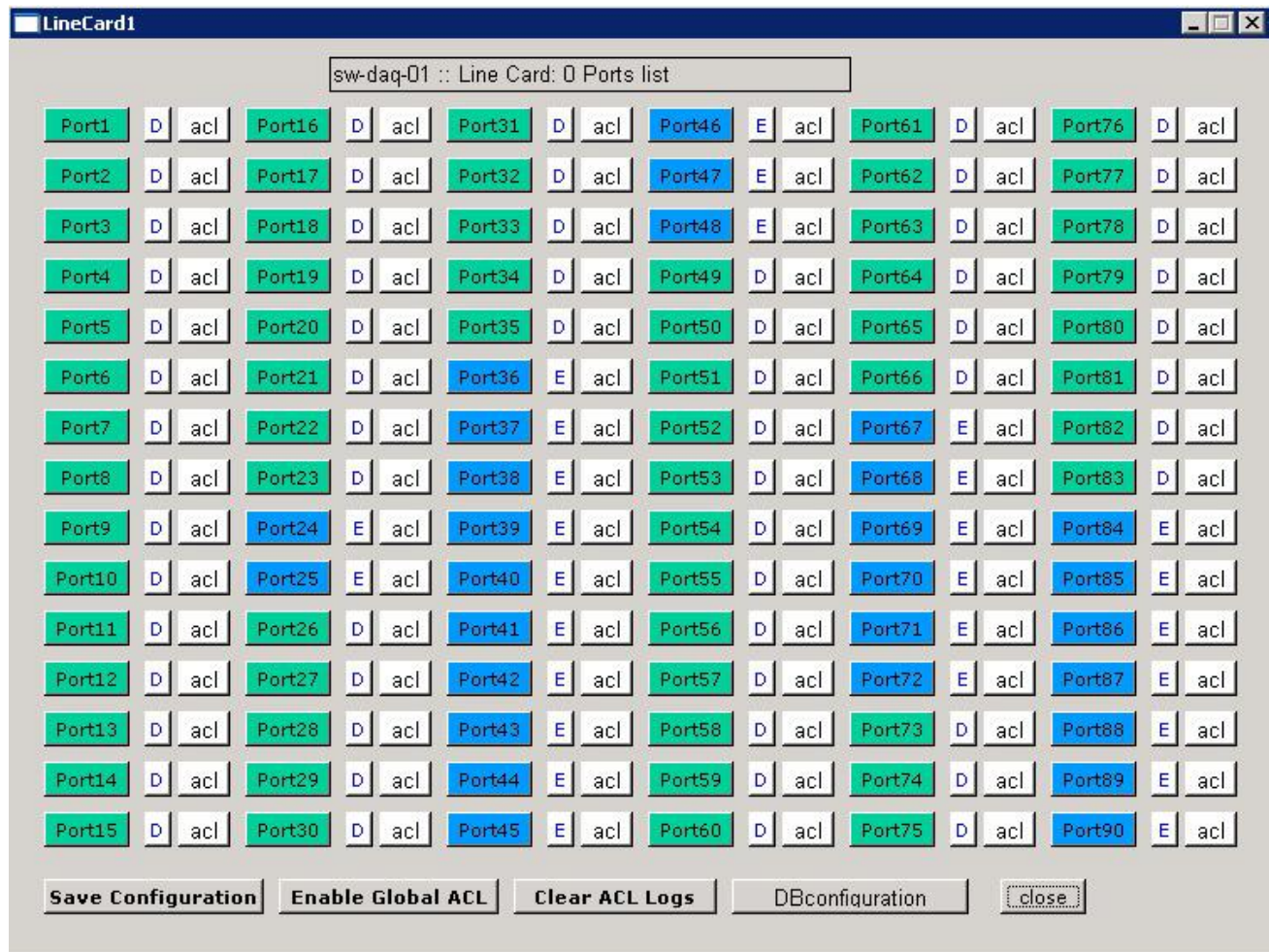


Figure 27 Liste des Port du grand Switch Terascale

✓ HP Gigabit Ethernet :

Ce panneau est similaire à celui du grand Switch. La seule différence est le nombre de bouton qui se réduit à 144 réparties de la même manière expliquée pour Terascale. On utilise dans la dernière version un panneau générique pour tous les types HP qui a un seul paramètre qui est le numéro de port.



Figure 28 Liste des ports Type HP Gigabit

✓ HP Fast Ethernet :

Similaire au panneau du type Gigabit sauf que dans ce type il y a 2 port de plus (50).

f. Statistiques

Ce panneau affiche des informations concernant le port choisit. On affiche avec les couleurs les trois statues reçues par le serveur et numériquement d'autres informations comme le nombre de multicast par seconde, le nombre de Bytes en entrée etc.

La grande partie de ce panneau sera occuper par les points de données importantes à la surveillance du réseau à dire les nombre d'octets par seconde les erreurs etc..., ces informations seront affichées sous forme des histogrammes.

Les divers histogrammes sont classe d'une manière logique pour cela j'ai utilisé un tab qui permet de grouper quaterne panneaux dont chacun comporte les histogrammes des Points de Données en considérant la répartition logique : entrée/sortie, taux de transfert/erreurs comme montre la figure suivante :

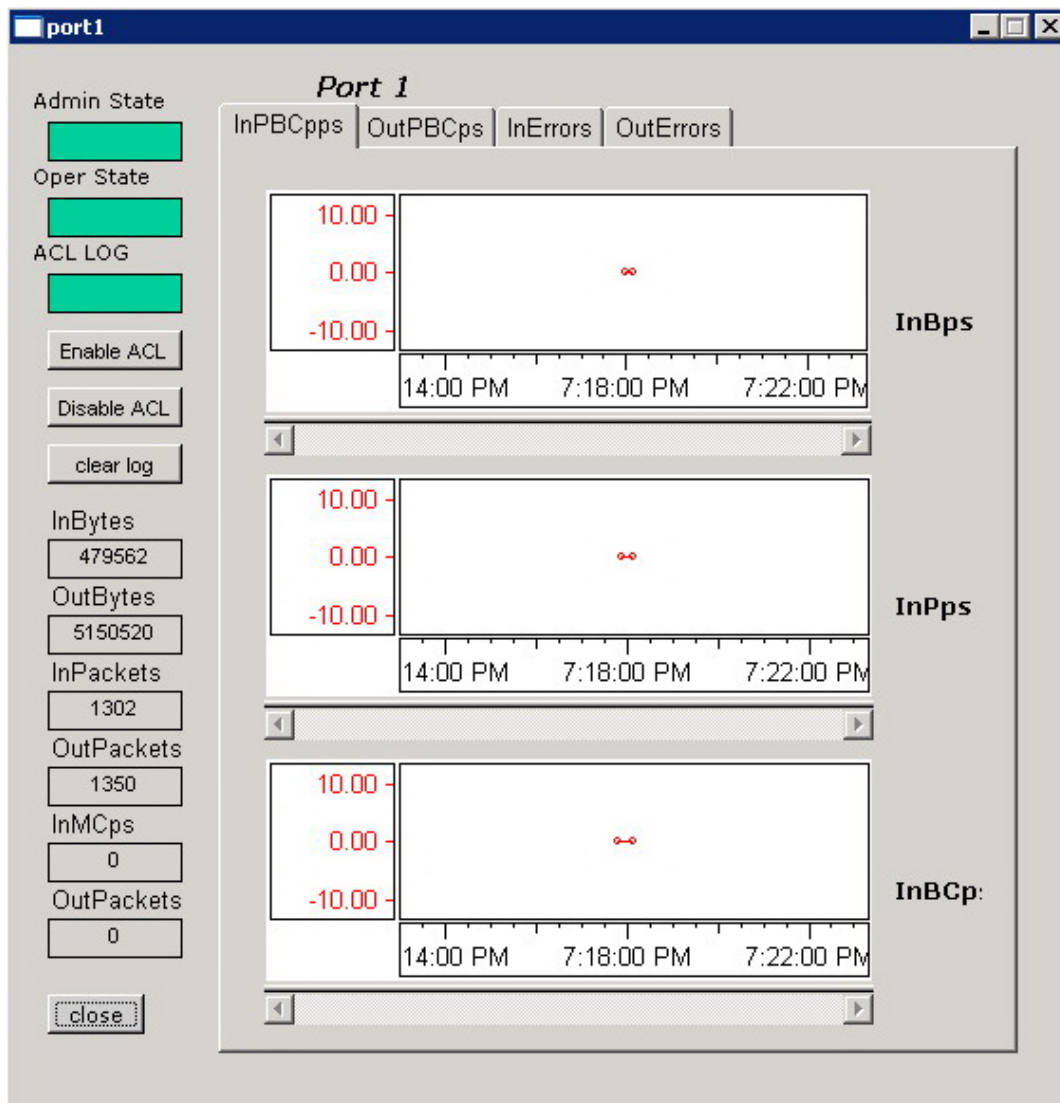


Figure 29 Statistique d'un port

5. Configuration de base de données

a. Introduction

La configuration de la base de données est utilisée dans le LHCb pour contenir les informations nécessaires à tous les équipements utilisés dans le réseau de l'expérience. Le Client (PVSS) a un accès au serveur de la base de données à partir duquel il doit avoir toutes les informations nécessaires sur les Switch utilisés.

Dans le projet on a configuré la base de données de façon à utiliser cette base pour insérer, mettre à jour ou ajouter de nouveaux éléments. Ainsi on a créé 6 panneaux additionnels pour le projet ces panneaux sont destinés à faire ces opérations sur les Types des équipements (par exemple HP_FE : pour HP Fast Ethernet) les équipements eux-mêmes et les ports.

L'ajout qu'a permit la configuration de base de données au projet est de le mettre dynamique au point de vue d'ajout d'équipements à surveiller dans le projet. La structure et les types des équipements que le réseau contient est stable mais pour le moment on utilise juste des dizaines de ces équipements pour le teste.

Pour permettre d'ajouter un nouveau Switch l'opération sera légère :

- Ajouter ce Switch dans la base de données avec le panneau AddDevice.pnl
- Créer un simple bouton à partir de la référence *TemplateSwitchHP* (ou TSC pour TerasCale)
- Donner le nom de Switch comme paramètre.
- Dessiner les liens (trait) vers les autres Switch.

En fait les premières opérations permet à la fois d'ajouter le Switch dans la base de données, crée les point de données réserver pour ce Switch et enregistre les services et commandes DIM. Cela évite le fait d'éditer le code pour l'ajout des Switch. L'application permet aussi d'ajouter plusieurs Switch à la fois et bien aussi tout ses ports avec leurs adresse MAC ou IP.

b. Panneau de configuration de la base de données

La structure de ces panneaux est plus simple. Un panneau pour ajouter ou mettre à jour les types, un deuxième pour ajouter les Switch par leur nom ou les mettre à jour est un 3^{ème} pour la configuration des ports. Les trois restant son de même nature sauf qu'ils permettront d'ajouter plusieurs types, Switch ou port à la fois.

The screenshot shows a software window titled "QuickTest _ : (NoName)" with a menu bar (File, Panel, ?). The main area is titled "Network Monitoring Data Base configuration". Below this, there's a "DB Status" section with a red "Error" button. A tabbed interface has "Device Type", "Device", and "Port" tabs, with "Device Type" currently selected. The "Add Device in Data base configuration" section contains a "Device Setting" box with the following fields:

- Device Type**: Text box containing "DAQ"
- Device Name**: Text box (empty)
- System Name**: Dropdown menu (empty)
- Node**: Dropdown menu showing "0"
- HW Type**: Text box (empty)
- Location**: Text box (empty)
- Comments**: Text box (empty)
- promiscuousmode**: Text box containing "1"
- SerialNb_F**: Text box containing "00000000"
- responsible**: Text box (empty)
- Fonctions List**: Dropdown menu showing "none"

At the bottom of the "Device Setting" box are three buttons: "add multiple types", "register", and "Cancel".

Figure 30 Panneau de configuration de base de données

Ce panneau contient les trois panneaux de configuration simple sous forme de tab. Chacun de ces panneaux contient un bouton qui redirige l'utilisateur vers le panneau de configuration complexe approprié.

On prend comme exemple l'insertion multiple des ports d'un Switch dans la base de données la configuration est faite suivant ce panneau de configuration complexe réservé aux ports.

The screenshot shows a software window titled "QuickTest : PortConfig". At the top, there is a menu bar with "File", "Panel", and "?". Below the menu bar, on the left, is the text "Add Device Type in DBconfig". To its right is a "DB Status" label followed by a green button labeled "Connected".

The main area is divided into two sections. The first section is titled "Port Setting *". It contains a "Device Type" dropdown menu set to "HP_FE", a "Device Name" text field containing "sw-hptest-01", and a "Port Number" section with "From" and "To" input fields set to "1" and "48" respectively.

The second section is titled "Adresse setting". It contains several input fields: "IP Name Text befor number" (containing "hptest-01_"), "From" and "To" (set to "1" and "48"), "Text after" (empty), "IP Adresse network" (containing "169.168.100"), "From" and "To" (set to "1" and "48"), "Subnet" (containing "255.255.255.0"), and "Location" (containing "building 157").

At the bottom of the window, there are three more settings: "Phy" (dropdown set to "SX"), "Port Way" (dropdown set to "input"), and "Speed" (empty text field). At the bottom right, there are two buttons: "register" and "Annuler".

Figure 31 Panneau de configuration multiple des ports

Conclusion

Les objectifs fixés pour mon stage sont atteints, bien que quelques améliorations puissent être envisageables.

Le produit finale répond au besoin cite dans le cahier de charge ainsi qu'il permet une souplesse pour permettre l'ajout des nouveaux équipements à surveiller par l'application dans le futur ou le teste sera sur le LHCB lui-même. Les divers versions du produit m'ont permis, de version à l'autre, de optimiser l'application de surveillance des deux niveaux espace requise et rapidité.

Le suivi de mon superviseur CERN M. Artur BARCZYK ainsi que tout le groupe du LHCB ONLINE mon aidé à bien concevoir cette application pour répondre aux besoins de la supervision des équipements réseau (Switch) de LHCB.

Ce stage m'a permit de découvrir un nouveau domaine où l'informatique collabore avec la physique, l'électronique et bien d'autres domaines industriels pour faire un tel exploit : le LHCB.

Parmi les avantages de mon stage et l'intégration en groupe du LHCB ONLINE, les conférences demi mensuelle et aussi pouvoir travailler en groupe dans une entreprise bien que découvrir en réel les diverses installations réseaux et les nouvelles technologies utilisées pour y manipuler et aussi pouvoir profiter de l'expérience de mon superviseur et de mes collègues du group pour approfondir mes compétence personnel.

Ce stage m'a aidé à améliorer mon niveau de langue parle (anglais) et m'a permis de faire des connaissances et de travailler avec des personnels de plusieurs pays.

REFERENCES BIBLIOGRAPHIQUES

- [BARCZYK A.] BARCZYK ARTUR. Network Architecture of the LHCb DAQ Test Bed. Juin 2006.
- [BARCZYK A. ET coll.] BARCZYK A., JOST B., Neufield N., LHCb ONLINE SYSTEM Network Architecture, LHCb ONLINE, Décembre 2005.
- [CPPM] CPPM, Introduction à la physique de LHCb, site web : <http://marwww.in2p3.fr/lhcb/>
- [ETM] ETM Professional control GmbH. PVSS_II_GettingStarted_Basics_en, 2004,
- [GASPAR C.] CLARA GASPAR, *Site web* <http://www.cern.ch/clara>
- [IT/PVSS] ITCObE, *Site web* <http://cern.ch/itcobe/Services/Pvss>
- [JCOP] CERN/JCOP Framework, *Site web* <http://cern.ch/itcobe/Projects/Framework>
- [LHCb] LHCb, site web <http://lhcb.web.cern.ch/lhcb/>
- [LHCb ONLINE] LHCb Online System, site web: <http://lhcb-online.web.cern.ch/lhcb-online/>



Institut Supérieur D'Informatique de
Modélisation et de leurs Applications

COMPLEXE DES CEZEAUX

BP 125 – 63173 AUBIERE CEDEX



Centre Européen de Recherche
Nucléaire

1211 Genève Suisse

Rapport de Stage 2^{ème} année

Surveillance du réseau de l'expérience L H C B ANNEXE

Présenté par: **Oualid LEBTAINI**

Responsable ISIMA : **Emanuel Mesnard**

Responsable CERN : **Artur Barczyk**

Date : **du 03/04 à 01/09 2006**

Durée: **5 mois**

I. Background Scripts

1. HP Switch

a. HP_PortsStatus.ctl :

Ce script calcul et mis a jour les statues des ports appartenant aux Switch de type HP. Ce scripte tourne en arrière plan ce qui permet un fonctionnement instantanée avec tout changement signalé par DIM.

```
//=====//
//      Author      :    Lebaini oualid. juillet 2006                      //
//      name         :    script PortStatus                               //
//      description  :    this script give a update for status from all ports in the HP Switches's. The data point //
//                      "ALLStatus" that contain a list of ports           //
//                      status will be updated if every change accurse.    //
//=====//
int main()
{
int NR_LINECARDS;
dyn_string nom_dp_switch=NM_Switch_DPs("NM_HP");// NM_HP its include all HP switch type
dyn_int nrPorts;

dyn_int Status,DynStatus;

string name;
for(int j=1;j <=dynlen(nom_dp_switch);j++)
{
    dpGet(nom_dp_switch[j]+".nrPorts",nrPorts[j]);
    ///////////////////////////////////initialisation ///////////////////////////////////
    for(int i=1;i<=nrPorts[j];i++)
        Status[i]=-1;

    dpSet(nom_dp_switch[j]+".ALLStatus",Status);

    ///////////////////////////////////

    for(int i=1;i<=nrPorts[j] ;i++)
    {
        name=Get_Name(i-1);
        ///////////////////////////////////
        dpGet(nom_dp_switch[j]+".Status."+name,DynStatus);
        Status[i]=check_status(DynStatus);
        dpSet(nom_dp_switch[j]+".ALLStatus",Status);
        ///////////////////////////////////
        dpConnect("Update_HP",true,nom_dp_switch[j]+".Status."+name);

    }
}
}

Update_HP (string nom_dp,dyn_int PortStatus)
{
```

```

    dyn_int inStatus;
    string nom_dp_switch=dpSubStr(nom_dp,DPSUB_DP); // its return name of switch data point
    dpGet(nom_dp_switch+".ALLStatus",inStatus);
    int index = substr(nom_dp,strpos(nom_dp,"Status.")+7,4); // port's number
    int port=index+1;

    int value= check_status(PortStatus);

    inStatus[port]=value;

    dpSet(nom_dp_switch+".ALLStatus",inStatus);

}

//*****
// function return port status calculated from admin and operate status
//*****
int check_status(dyn_int PortStatus)
{
    int ret=0; //by default its in error
    switch(PortStatus[1])
    {
        case(1):if(PortStatus[2]==1)
        {
            if(PortStatus[3]=0) ret=1;
            else ret=3;
        }
        else if(PortStatus[2]==2) ret=0;
        break;

        case(2): ret=2;
        break;

    }

    return ret;
}

//*****
// function give switch name formatted with 4 characters format
//*****
string Get_Name(int num)
{
    string tmp=""+num;
    if(num<10 ) tmp="000";
    else
    { if(num<100) tmp="00";
      else if(num<1000) tmp="0";
    }
    tmp=tmp+num;
    return tmp;
}

```

```

//*****
// function give liste of data point's name used by project
//*****
dyn_string NM_Switch_DPs(string Prefix)
{
    dyn_string DP_Names,split,names;
    dyn_string DPT=dpTypes (Prefix+"*");
    int index=1;
    for(int j=1;j<=dynlen(DPT);j++)
    {

        names=dpNames(" ",DPT[j ]);

        for(int i=1;i<=dynlen(names);i++)
        {
            split=strsplit(names[i],":");
            DP_Names[index]=split[2];
            index++;
        }
    }
    return DP_Names;
}

```

b. HP Status.ctl :

Ce script calcul et mis a jour les statues des Switch de type HP. Ce scripte tourne en arrière plan ce qui permet un fonctionnement instantané avec tout changement signalé sur les point de données contenant la liste des statues des ports de chaque Switch.

```

//=====//
//  Lebtaini oualid 07/2006                                //
//  Name   : Update SwitchStatus                            //
//  Description: script that check all of port status and calculate //
//           switch status                                    //
//=====//

main()
{
    dyn_string nom_dp_switch=NM_Switch_DPs("NM_HP");// NM_HP its include all HP switch type
    dyn_int   nrPorts;
    dyn_int Status;
    for(int j=1;j <=dynlen(nom_dp_switch);j++)
    {
        dpGet(nom_dp_switch[j]+" .nrPorts",nrPorts[j]);

        dpConnect("Update_Switch_State",true,nom_dp_switch[j]+" .ALLStatus");
    }
}

```

```

}
Update_Switch_State(string nom_dp,dyn_int ALLStatus)
{
    int err=0,ok=0;
    int Status;
    string nom_dp_switch=dpSubStr(nom_dp,DPSUB_DP); // its return name of switch data point
    int NrPorts;
    dpGet(nom_dp_switch+".nrPorts",NrPorts);

    for(int i=1;i<=NrPorts;i++)
    {
        if(ALLStatus[i]==0)
        {
            Status=0;
            err++;
        }
        else ok++;
    }

    if(err==0) Status=1;

    dpSet(nom_dp_switch+".SwitchStatus",Status);

}

dyn_string NM_Switch_DPs(string Prefix)
{
    dyn_string DP_Names,split,names;
    dyn_string DPT=dpTypes (Prefix+"*");
    int index=1;
    for(int j=1;j<=dynlen(DPT);j++)
    {

        names=dpNames(""*,DPT[j ]);

        for(int i=1;i<=dynlen(names);i++)
        {
            split=strsplit(names[i],":");
            DP_Names[index]=split[2];
            index++;
        }
    }
    return DP_Names;
}

```

2. TERASCALE Switch

Dans la conception de ce projet on considère qu'un Switch/Router de type Terascale logiquement est simulée à un groupement de 14 Switch de types HP (de 90 ports chacun).

Pour les scripts TSCPortsStatus (statues des ports) et TSCStatus (statues des LineCards) sont semblable au script HPPortsStatus et HPStatus.

a. TSCPortsStatus

```
//=====//
```

```

// Lebtaini oualid 07/2006
// Name      : Update Line Card Status
// Description : script that check all of port status and group
//              in the ALL_P_Cards data point element 91 by linecards //
//=====//

int main()
{
    dyn_string nom_dp_switch=NM_Switch_DPs("NM_TerasCal");// NM_ terascale its include all HP switch type
    dyn_int  nrPorts;
    dyn_int PortStatus;
    dyn_dyn_int Status;
    string name;
    int NR_LINECARDS;

    for(int j=1;j <=dynlen(nom_dp_switch);j++)
    {
        dpGet(nom_dp_switch[j]+"nrPorts",nrPorts[j]);
        NR_LINECARDS = (nrPorts[j]-2)/90;
        ///// initialize value
        for(int j=1;j<=NR_LINECARDS;j++)
        {
            for(int i=1;i<=90;i++)
            {
                Status[j][i]=-1;
            }
            int index;
            for(int l=1;l<=NR_LINECARDS;l++)
            {
                // All_P_Cards contains an array of list of ports status value
                dpSet(nom_dp_switch[j]+"ALL_P_Cards."+l,Status[l]);
                for(int i=((l-1)*90+1);i<=l*90 ;i++)
                {
                    index=i-((l-1)*90+1)+1;
                    name=Get_Name(i-1);
                    // initilizing programm first time running
                    dpGet(nom_dp_switch[j]+"Status."+name,PortStatus);
                    Status[l][index]=Check_Status(PortStatus);

                    dpSet(nom_dp_switch[j]+"ALL_P_Cards."+l,Status[l]);

                }
            }
        }
    }

    Update_LineCards (string nom_dp,dyn_int inStatus)
    {
        // its return name of switch data point
        string nom_dp_switch=dpSubStr(nom_dp,DPSUB_DP);
        // port's number

```

```

int index = substr(nom_dp, strpos(nom_dp, "Status.")+7, 4);
int port = index+1;
dyn_int Card_Status;
    // de determinate card number NB: port 90 is in card n:1 so devid by 91 ;)
int card= port/91 + 1;
int port_index=port - ((card-1)*90 +1)+1;

dpGet(nom_dp_switch+".ALL_P_Cards."+card, Card_Status);
Card_Status[port_index]=Check_Status(inStatus);
dpSet(nom_dp_switch+".ALL_P_Cards."+card, Card_Status);
}

```

int Check_Status(dyn_int PortStatus)

```

{
    int ret=0; //by default its in error
    switch(PortStatus[1])
    {
        case(1): if(PortStatus[2]==1)
            {
                if(PortStatus[3]==0) ret=1;
                else ret=3;
            }
            else if(PortStatus[2]==2) ret=0;
            break;
        case(2): ret=2;
            break;
    }

    return ret;
}

```

string Get_Name(int num)

```

{
    string tmp="" + num;
    if(num < 10 ) tmp="000";
    else
    { if(num < 100) tmp="00";
      else if(num < 1000) tmp="0";
    }
    tmp=tmp+num;
    return tmp;
}

```

dyn_string NM_Switch_DPs(string Prefix)

```

{
    dyn_string DP_Names, split, names;
    dyn_string DPT=dpTypes (Prefix+"*");
    int index=1;

```



```

for(int j=1;j<=dynlen(DPT);j++)
{

    names=dpNames("*",DPT[j]);

    for(int i=1;i<=dynlen(names);i++)
    {
        split=strsplit(names[i],":");
        DP_Names[index]=split[2];
        index++;
    }
}
return DP_Names;
}

```

b. TSCStatus

```

main()
{
int NR_LINECARDS;
dyn_string nom_dp_switch=NM_Switch_DPs("NM_TerasCal");// NM_ terascale its include all HP switch type
dyn_int nrPorts;

dpGet(nom_dp_switch[1]+".nrPorts",nrPorts[1]);

NR_LINECARDS=(nrPorts[1]-2)/90;

for(int j=1;j<=dynlen(nom_dp_switch);j++)
{
    for(int l=1;l <= NR_LINECARDS;l++) //l its a line cards numbers
    {
        dpConnect("Update_Card_State",true,nom_dp_switch[j]+".ALL_P_Cards."+l);
    }
}
}

```

Update_Card_State(string nom_dp,dyn_int ALL_P_Status)

```

{
    int err=0,warn=0;
    int Card_Status;
    string nom_dp_switch=dpSubStr(nom_dp,DPSUB_DP); // its return name of switch data point
    int card =substr(nom_dp,strpos(nom_dp,"ALL_P_Cards.")+12,1); // card's number

    for(int i=1;i<=90;i++)
    {
        switch(ALL_P_Status[i])
        {
            case(0):Card_Status=0; err++;break;

```

```

        case(3):warn++;break;
    }
}

if(err !=0)
{
    // if there are one error== switch in error
    Card_Status=0;
    dpSet(nom_dp_switch+".SwitchStatus",Card_Status);
}
else
{
    if(warn!=0) Card_Status=3;
    else Card_Status=1;
}
dpSet(nom_dp_switch+".CardsStatus."+card,Card_Status);
}

```

dyn_string NM_Switch_DPs(string Prefix)

```

{
    dyn_string DP_Names,split,names;
    dyn_string DPT=dpTypes (Prefix+"*");
    int index=1;
    for(int j=1;j<=dynlen(DPT);j++)
    {

        names=dpNames(" ",DPT[j ]);

        for(int i=1;i<=dynlen(names);i++)
        {
            split=strsplit(names[i],":");
            DP_Names[index]=split[2];
            index++;
        }
    }
    return DP_Names;
}

```

c. TSCSwitchStatus.ctl

*Ce script nouveau par rapport au Switch HP calcul le statue global du Switch partir des statues global des LINECARDS.
(Calculer et mise à jour par TSCStatus.ctl).*

```

main()
{
    int NR_LINECARDS;
    dyn_string nom_dp_switch=NM_Switch_DPs("NM_TerasCal");// NM_ terascale its include all HP switch type
    dyn_int  nrPorts;

```

```

dpGet(nom_dp_switch[1]+".nrPorts",nrPorts[1]);

NR_LINECARDS=(nrPorts[1]-2)/90;

for(int j=1;j<=dynlen(nom_dp_switch);j++)
{
    for(int l=1;l <= NR_LINECARDS;l++) //l its a line cards numbers
    {
        dpConnect("Switch_Status_TSC",true,nom_dp_switch[j]+".CardsStatus."+l);
    }
}

}

Switch_Status_TSC(string nom_dp,int CardStatus )
{
int NrPorts,err=0,SwitchStatus;
int Line_Card_Status;
    string nom_dp_switch=dpSubStr(nom_dp,DPSUB_DP); // its return name of switch data point
    int card =substr(nom_dp,strpos(nom_dp,"CardsStatus.")+12,2);

    dpGet(nom_dp_switch+".nrPorts",NrPorts);

    int NBLINECARDS=(NrPorts-2)/90;

    if(CardStatus==0)
    {
        SwitchStatus=0;
    }
    else
    {
        if(CardStatus==3)            SwitchStatus=3;
        else
        {
            // status of this line card is ok verify the others
            for(int i=1;i< card ;i++)
            {
                dpGet(nom_dp_switch+".CardsStatus."+i,Line_Card_Status);
                if(Line_Card_Status==0)
                {
                    err++;
                    SwitchStatus=0;
                }
                else SwitchStatus=1;
            }
            for(int i=card+1;i <= NBLINECARDS;i++)
            {
                dpGet(nom_dp_switch+".CardsStatus."+i,Line_Card_Status);
                if(Line_Card_Status==0)
                {
                    err++;

```

```

        SwitchStatus=0;
    }
    else SwitchStatus=1;
}
}
dpSet(nom_dp_switch+".SwitchStatus",SwitchStatus);
}
}
dyn_string NM_Switch_DPs(string Prefix)
{
    dyn_string DP_Names,split,names;
    dyn_string DPT=dpTypes (Prefix+"*");
    int index=1;
    for(int j=1;j<=dynlen(DPT);j++)
    {

        names=dpNames("*",DPT[j ]);

        for(int i=1;i<=dynlen(names);i++)
        {
            split=strsplit(names[i],":");
            DP_Names[index]=split[2];
            index++;

        }
    }
    return DP_Names;
}

```

II. Panel Script

Cette partie est en rédaction Une liste complète sera mise en ligne pour téléchargement des codes sources et explications des fonctions nécessaires.