Configuration DB
Status report

Lana Abadie

Contents

- Objectives & requirements (reminder)
- TFC project status
- DAQ project (in progress)
- Routing tables : use case
- Modeling the network with tables (1/2)
- Routing table algorithm
- Some results
- Design and implementation rules
- Conclusion

Objectives & requirements (reminder)

- **Store all controllable devices with**
  - their properties
  - the links between them
  - their hierarchy
- **Load necessary information for the ECS**
  - to configure detectors
  - to start up an experiment
  - to monitor devices
- **Database design key aspects**
  - generic schema
  - completeness
  - performance
  - maintenance

TFC project status

- TFC table design exists
- Prototype in production
- Use of JCOP tool to save devices and recipes in the conf DB from PVSS
- Usable even if there is no DB connection via "a cache".

DAQ project (in progress)

- DAQ table design exists (very similar to TFC design)
- Main use case : generate routing/destination tables automatically given the network topology. (details in the next slides)

Remaining things:

- Implement a PVSS interface to configure switches
- Design codes & job options for software installed in the farm nodes.

Routing tables : use case

Requirements:

- Generated from the data contained in the DB
- independent from the network topology
- provide IP/Ethernet routing table

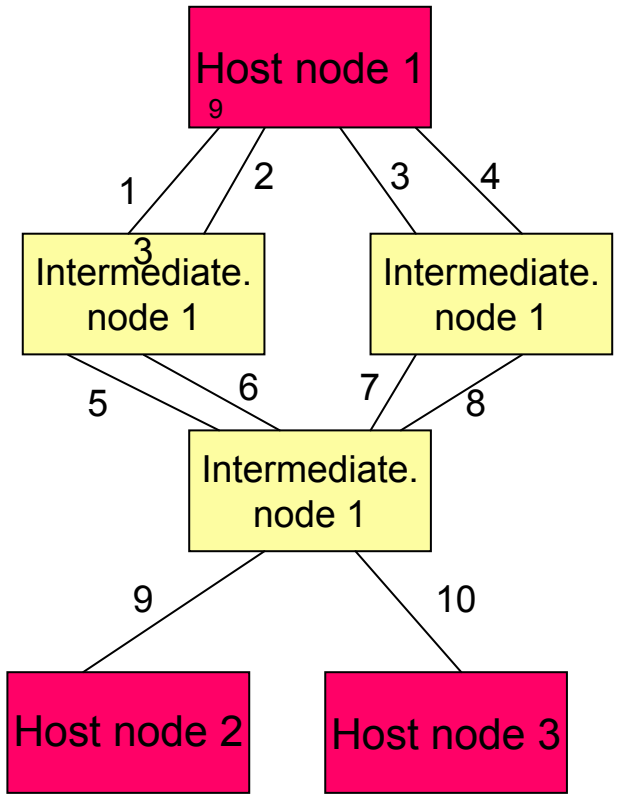
Definitions:

- **Host node:** a node which processes data
- **Intermediate node**: a node which transfers data
- **Path**: sequence of links whose sum of the link weights>0

Prerequisites:

- Connectivity table (links between devices)
- IP/Ethernet table (list of IP/Ethernet @ and devices)

Modeling the network with tables (1/2)

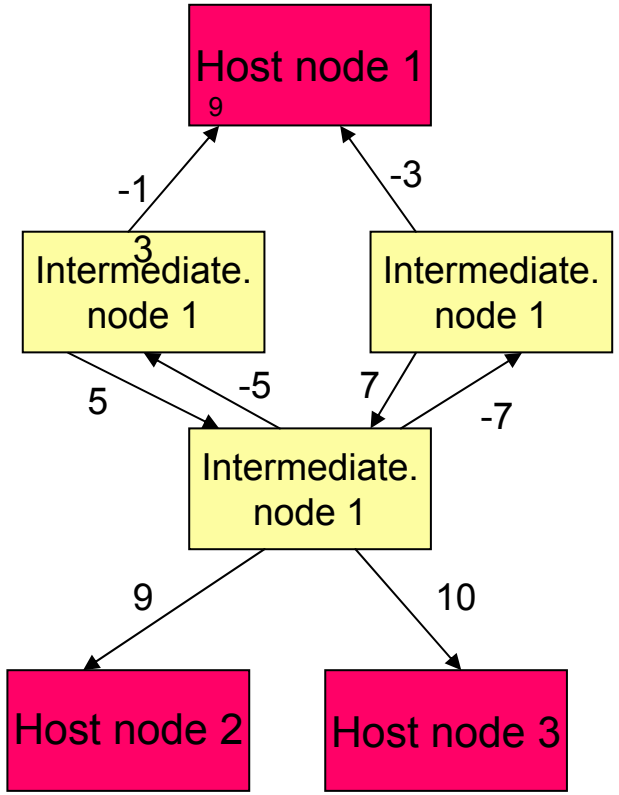


daq_connectivity	
PK	**linkid**
FK2	switch_from
	port_nbrfrom
FK1	switch_to
	port_nbrto
FK4	link_type
	bidirectional_used
FK3	link_trunkNb
	link_termin

Oriented link or not

Link weight

Modeling the network with tables (2/2)



Daq_Nodelink	
PK,FK1	**Linkid**
	link_termin pfrom pto lkaggreg sfrom sto

- Use of “trunk” links between devices

Routing table algorithm

Principles

- Find all the paths between the given switch and the reachable host nodes (destination)
- For each reachable host, select the shortest path
- Generate the routing table with the following entries:
  - IP and Ethernet @ of the next hop
  - Subnet mask, IP and Ethernet @ of the destination

Implementation

- Use of PL/SQL (portable, oracle language…)

Extension

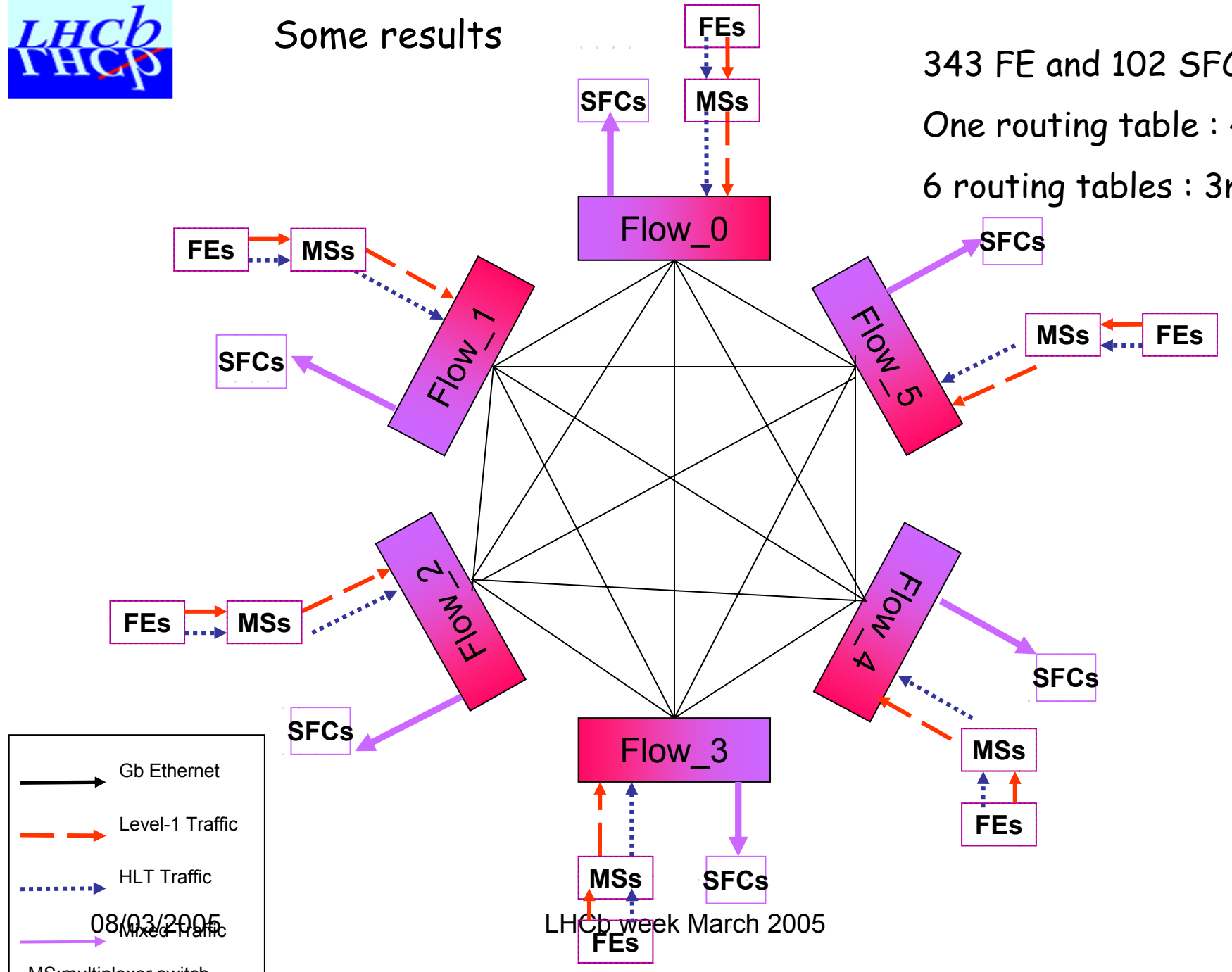
- Destination table if no IP/Ethernet @ (cf TFC system)

Some results

343 FE and 102 SFCs

One routing table : 40 sec

6 routing tables : 3mn 39



Design and implementation rules

- For any subsystems, the design will include:
  - Device Type table
  - Device table
  - Connectivity table
  - Destination/routing tables (automatically generated)
  - JCOP tables (device structures, recipes, hierarchy)
- Use of JCOP tool to save devices and recipes from PVSS to conf. DB
- Implement a "cache" to run a PVSS project even if there is no DB connection (for institutes).

Conclusion

- TFC project in production: waiting for feedback.
- DAQ project in good progress: may be in production for June
- Tools to store connectivity, to check consistency…
- Need to schedule a DB workshop with the subsystems by June. Please start to think of your requirements