Enhanced Ethernet Switching Technology for Adaptive Hard Real-Time Applications

Rui Santos, rsantos@ua.pt, UA
SUMMARY
MOTIVATION

- Switched Ethernet became common in real-time communications
  - Some interesting properties
    - Large bandwidth
    - Cheap network controllers
    - Micro-segmentation
      - Collisions are eliminated
    - Multiple parallel forwarding paths
    - High availability
  - But there are still limitations
    - FIFO queues
    - Limited number of priorities
    - Memory overflows

January 28, 2010, Aveiro, Portugal
SOLUTIONS

- **Commercial Of-The-Shelf Ethernet Switches**
  - Limiting the generated traffic by the application design
  - Traffic shaping
  - Master-Slave protocols (FTT-SE, …)

- **Customized Ethernet Switches**
  - TTEthernet
  - Profinet-IRT
    - Static pre-defined configuration
    - Online admission control is not generally available
    - Miss on-line adaptation
  - FTT-Enabled Switch (our solution)

January 28, 2010, Aveiro, Portugal
FTT-Enabled Switch (Mechanism)

- Based on **Flexible Time-Triggered Paradigm**
- **Master-slave** transmission control technique
- Communication occurs in fixed slots
  - (Elementary Cycles – Ecs)
    - ECs are organized in **synchronous** and **asynchronous** windows
    - Supports **synchronous**, **asynchronous** and **non real-time** traffic, with strict temporal isolation
- The ECs start with a **Trigger Message (TM)** sent by the Master (switch), that contains the schedule for each EC

January 28, 2010, Aveiro, Portugal
FTT-Enabled Switch (Properties)

- Traffic scheduling and management
  - Global traffic coordination in a common timeline
  - Supports online admission control and dynamic QoS management
  - Allows arbitrary traffic scheduling policies

- Traffic classification, confinement and policing
  - Seamless integration of standard non-FTT-compliant nodes without jeopardizing the real-time services
  - Asynchronous traffic is autonomously triggered by the nodes
  - Unauthorized transmissions can be readily blocked at the switch input ports, thus not interfering with the rest of the system
HOW CAN WE IMPLEMENT THAT?

FTT-ENABLED SWITCH (ARCHITECTURE)

Complex Algorithms
Difficult HW Implementation

Code Reuse

FTT-SE Master

Predictability

Determinism

Speed of execution

January 28, 2010, Aveiro, Portugal
FTT-Enabled Switch (Implementation)

- **Switching Module**
  - Implemented in hardware

- **Master Module**
  - Implemented in FPGA embedded processor (Synthesizable or Hardwire)
  - Utilization of a CPU – communication with the FPGA is carried out by the conventional interface (Ethernet, USB, PCI, …)

January 28, 2010, Aveiro, Portugal
FTT-ENABLED SWITCH (IMPLEMENTATION)
Submitted traffic 
1kB packets, $T_{avg} = 250 \mu s$

**Offset at the switch egress** (relative to the TM)
Measure:

- $T_{TM_{avg}} = 1,000ms$
- $T_{TM_{max}} = 1,0003ms$
- $T_{TM_{min}} = 0,99998ms$
- $STD_{TM} = 138\text{ns}$
**SERVER-BASED TRAFFIC SCHEDULING**

**Motivation**
- Address the growing NES requirements to:
  - support streams with arbitrary arrival patterns
  - provide QoS guarantees.

**Solution**
- We propose to integrate CPU-based server policy in the FTT-Enabled Switch
  - Polling Server, Deferrable Server, Sporadic Server
- Providing **hierarchical composition, reconfigurability** and **adaptability**
  - Online creation, deletion and adaptation of servers

January 28, 2010, Aveiro, Portugal
SERVER-BASED TRAFFIC SCHEDULING (INTEGRATION)

- **First Level**
  - SW – Polling Server
  - AW – Polling Server or a Deferrable Server

- **Second Level**
  - Manages the sporadic and the NRT traffic inside the AW

- **Third Level**
  - Implements specific servers, *virtual channels*

January 28, 2010, Aveiro, Portugal
SERVER-BASED TRAFFIC SCHEDULING (IMPLEMENTATION)

- **Servers implemented in Hardware (Switching Module)**
  - High reactivity
  - Less flexibility (the number of the servers is fixed)
  - Complex server scheduling methods can require a significant amount of hardware resources.

- **Servers implemented in Software (Master Module)**
  - High flexibility
  - The server latency is relatively large

January 28, 2010, Aveiro, Portugal
**SERVER-BASED TRAFFIC SCHEDULING (EXPERIMENTAL RESULTS)**

- **Elementary Cycle** = 1 ms; Asynchronous Window = 42%
- SS1, SS2 – *sporadic servers* with C=3200B and T=1 ms
- BS – *background server* uses the remaining bandwidth

- **Video SS1**
  - Peak load = 21.9 Mbps

- **UDP SS2**
  - Average load = 99.9 Mbps

---

January 28, 2010, Aveiro, Portugal
SERVER-BASED TRAFFIC SCHEDULING
(EXPERIMENTAL RESULTS)

January 28, 2010, Aveiro, Portugal
THE SAME EXPERIMENT WITH A NORMAL SWITCH!

Video – SS1

Traffic gen. – SS2

TCP – BS

January 28, 2010, Aveiro, Portugal
**FTT-ENABLED SWITCH (MULTIPLE SWITCHES)**

**Problem**
- How to create a network with multiple switches, where the communication is based on the FTT-Enabled Switch (HaRTES)?

**Solutions**
- Network with one FTT-Enabled Switch and multiple COTS switches
- Network with multiple FTT-Enabled Switches
FTT-Enabled Switch (Multiple Switches)

- Network with one FTT-Enabled Switch and multiple COTS switches
  - **Properties**
    - Trigger Messages are generated by FTT-Enabled Switch and disseminated by the others switches
  - **Advantages and Disadvantages**
    - ✓ Solution compatible with common networks
    - ✖ COTS switches don’t perform traffic policing
    - ✖ The Trigger Message latency can generate problems of synchronization

January 28, 2010, Aveiro, Portugal
**FTT-Enabled Switch (Multiple Switches)**

- **Network with multiple FTT-Enabled Switches**
  - **Properties**
    - Each FTT-Enabled Switch creates its own **synchronization domain**
    - It needs a **gateway** to interconnect different synchronization domains
    - Gateway can be avoided if FTT-Enabled Switches are slaves to each other
  - **Advantages and Disadvantages**
    - ✓ Whole network is covered by the traffic policing
    - ✗ It needs a gateway

January 28, 2010, Aveiro, Portugal
FTT-Enabled Switch (Current Status)

- **HaRTES/B**
  - Basic switching: Executed
  - Capability to separate different traffic classes: Executed
  - On-line scheduling: Executed

- **HaRTES/S**
  - Error detection
  - Traffic policing: Partially executed

- **HaRTES/Q**
  - Dynamic QoS management capabilities: Partially executed

January 28, 2010, Aveiro, Portugal
The growing availability of FPGAs, associated tools and communication IP cores opens the way to build customizable devices with properties that are tuned to specific application domains.

We propose an enhanced Ethernet switch that:
- Provides **seamless integration of** any (kind of) **nodes without causing any interference**
- Provides **filtering of unauthorized transmissions**
- Allows **arbitrary synchronous traffic scheduling policies**
- Allows **arbitrary server scheduling and hierarchical composition**
- Provides **dynamic creation and adaptation of servers**
ON GOING AND FUTURE WORK

- Finish the proposed work in the project
- Study and integrate multiple switch architecture
  - Adapt the enhanced switch to allow integration in architectures with multiple synchronization domains
- Replicate the Master
- Study over the schedulability analysis of the server-based traffic scheduling

January 28, 2010, Aveiro, Portugal
THANK YOU