CMPT 885-3: SPECIAL TOPICS: HIGH-PERFORMANCE NETWORKS
Final Project Presentation
Spring 2002

Firewire & ns2

An Overview of IEEE 1394 Simulation in ns2

Glendon Holst
gholst@cs.sfu.ca

Outline

- ◆ Introduction to IEEE1394 (Firewire, iLink).
  - ◇ Specifications
  - ◇ Topology and Configuration
  - ◇ Protocols (Arbitration, Asynchronous, Isochronous)
  - ◇ Packets
- ◆ Introduction to ns2 Simulation Concepts
  - ◇ Nodes, Classifiers, Links, MAC, LanNodes, Agents,…
  - ◇ Packets
  - ◇ Timers and Simulating Events
- ◆ Overview of IEEE1394 in ns2
  - ◇ Declaring Topoligies
  - ◇ Arbitration and Mac/ieee1394
  - ◇ Protocols and Agent/ieee1394, LL/ieee1394
- ◇ Summary

Introduction to Firewire

- **Summary**
  - ◇ IEEE1394-1995, IEEE1394a, IEEE1394b (based on IEEE1212)
  - Serial-Bus (Packet based and network-like)
    - Highspeed
      - ◇ Currently: 100 Mbps, 200 Mbps, 400 Mbps
      - ◇ Proposed (IEEE1394b): 800 Mbps, 1.6 Gbps, 3.2 Gbps (OC ~ 100 M).
    - ◇ Powered (Nodes may add or use power to the bus)
    - ◇ Hot-Pluggable (plugging produces a bus-reset)
    - ◇ Peer-to-Peer (Unlike USB which requires a host CPU)
    - ◇ Asynchronous and Isochronous transfers modes.
  - Primarily Connects Peripheral Devices
    - ◇ Well suited for video, hard-drives, cameras, printers, scanners.
    - ◇ Experimental IP over IEEE1394.
- **Topology and Configuration**
  - ◇ Tree Topology (Indentify root node)
  - ◇ Node Identification on Bus-Reset (Sticky IDs)

Introduction to Firewire

- **Protocol Overviews**
  - **125 µsec cycles.** Cycle Master (root) sends Cycle_Start packet.
  - **Arbitration During the Cycle.** Nodes request access to bus and the root grants access.
  - **Subactions.** <[arb] [data-packet] ack-gap [ack] subaction-gap> [arb]
  - **Asynchronous packets require Ack.**
  - **Isochronous packets are NOT acknowledged.** 2x packet size allowed. Up to 80% of the cycle (100 µsec) can be reserved to isochronous transfers.

Introduction to Firewire

- **Arbitration**
  - After time-out (either subaction or bus-idle) send request to parent.
  - Node chooses first request to arrive at its ports (or internal), and passes the request to its parent.
  - The root grants the first request to arrive at its ports (or internal). A Grant packet notifies the winners.
  - Nodes closer to the root have positional priority, but not necessarily overall priority.
  - Nodes can only win arbitration once per cycle (Arb-Won flag).
  - The root node must win arbitration to send Cycle_Start (an issue in 1394a and later optimizations, such as fly-by arbitration).

Introduction to Firewire

- Asynchronous Transfer
  - For Guaranteed Packet Delivery
  - Transaction Based
    - Read, Write, Lock
  - Must arbitrate to initiate a transfer.
  - Ack must return within subaction time.
    - For write, only an Ack is required.
    - For read, there are two possible results:
      - **Concatenated Transfer**: The requested node can return the result fast enough, so the data is appended to the Ack.
      - **Split Transfer**: The requested node cannot return the result in time, so an Ack is sent, indicating a delayed response. The requested node must then arbitrate for access before sending the requested data.
  - Max payload size:
    - 512 bytes for 100 Mbps, 1024 bytes for 200 Mbs, 2048 bytes for 400 Mbs, 4096 for 800 Mbs, etc.

Introduction to Firewire

- **Isochronous Transfer**
  - For Guaranteed Time-Slice (bandwidth).
  - Must obtain channel and bandwidth from Isochronous Resource Manager (root).
    - Channels identify target nodes. 64 channels.
  - Packets transmitted after the Cycle_Start packet.
    - This is the Isochronous abitration phase, which looks like:

      [arb][ch1-packet] isoch-gap [arb][ch2-packet] isoch-gap ...
      [arb][ch1-packet][ch2-packet] isoch-gap ...

    - After these packets are transmitted (in order of channel number), the ansynchronous arbitration phase of the cycle begins.
  - Max payload size:
    - 1024 bytes for 100 Mbps, 2048 bytes for 200 Mbs, 4096 bytes for 400 Mbs, 8192 for 800 Mbs, etc.

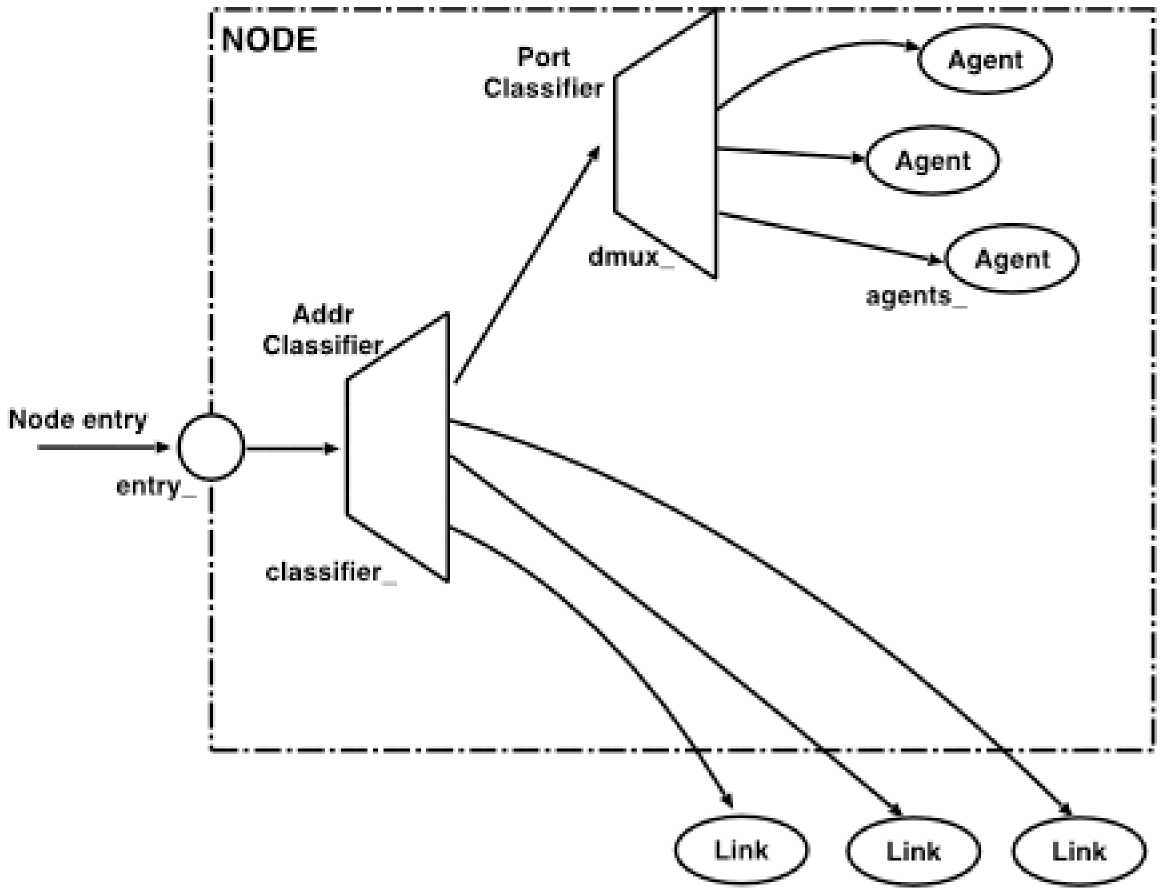
Introduction to Firewire

- **Packet**
  - Header is 20 bytes
  - Packet is padded mod 4 bytes (quad-word).
  - Data Payload length limited by port speed.
  - Last Quad-Word is Data-CRC.
- **Header**
  - **destination_ID** (2 bytes).
  - **tl** (trans label), **rt** (retry), **tcode** (trans code), **pri** (priority) : (2 bytes).
  - **source_ID** (2 bytes).
  - **destination_offset** (6 bytes).
  - **packet_type** (4 bytes).
  - **header_CRC** (4 bytes).
- **Addressing**
  - Node_IDs: 10 bits for Bus, 6 bits for Node on bus.
  - Destination: 48 bits, memory address location on target node.

Introduction to ns2

- Simulation Concepts
  - ◇ Need to understand so we can later map IEEE1394 to ns2.
  - Nodes, Classifiers, Links, and Agents.
    - ◇ A node is like a vertex, a link is like an edge.
    - ◇ When packets arrive at nodes, they are either for that node or for another node. A packet classifier determines this.
    - ◇ If the packet is for this node, it must be sent to the appropriate agent. This is done by another classifier.
    - ◇ If the packet is for another node, the classifier determines which outgoing link to send the packet along.

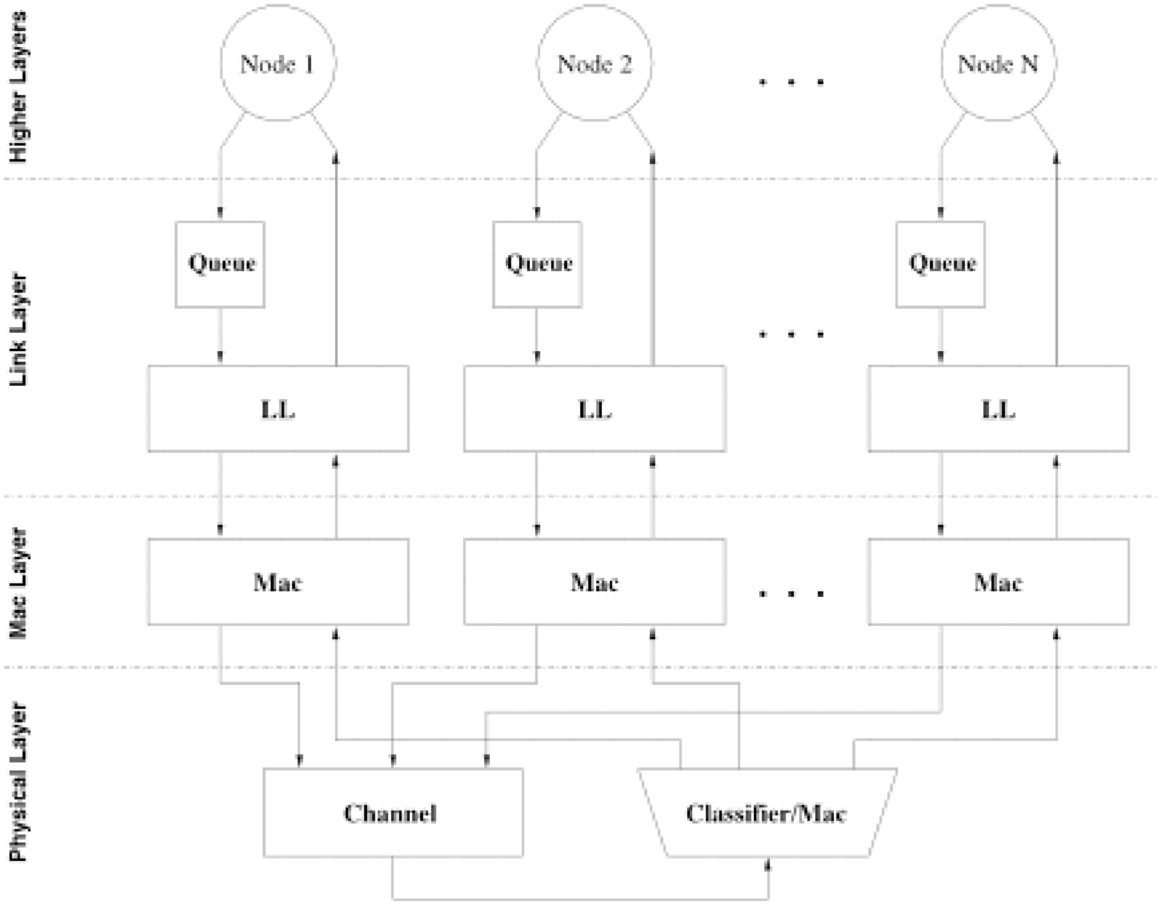
Introduction to ns2



Introduction to ns2

- Lan Nodes
  - Nodes are suited for point to point communication.
  - We need to simulate issues of bus contention and arbitration.
  - LanNodes let us capture Link Layer, MAC layer, and even PHYs layer behaviours.
  - Nodes remain as the higher level network elements.

Introduction to ns2



Introduction to ns2

- **Packets**
  - All packets supported by the system are sent as a Packet.
  - Packets only need to contain sufficient state to simulate the behaviour (they don’t need to exactly match real packets).
    - Packet data is simulated via a size_ variable.
    - We use address information from the IP packets to simulate IEEE1394 addresses.
- **Timers and Events**
  - To simulate events: aquire timer, register event (class and method to invoke).
  - When the timer expires, the event occurs and the method is invoked.
  - Examples: simulating link delays, etc.

Firewire in ns2

- **Topology**
  - Require user to set up network with the following attributes:
    - Tree structure
    - Parents have lower id numbers then their children (i.e., root is lowest numbered node).
  - Example *.tcl Simulation Script:

```
set lan0 [$ns make-lan "$n0 $n1"
                        400Mb 1ms LL/ieee1394 Queue/DropTail Mac/ieee1394]
set lan1 [$ns make-lan "$n1 $n2"
                        400Mb 1ms LL/ieee1394 Queue/DropTail Mac/ieee1394]
set lan0 [$ns make-lan "$n0 $n3"
                        400Mb 1ms LL/ieee1394 Queue/DropTail Mac/ieee1394]
set lan0 [$ns make-lan "$n1 $n4"
                        400Mb 1ms LL/ieee1394 Queue/DropTail Mac/ieee1394]
```

Firewire in ns2

- Arbitration and Mac/ieee1394
    - Wait for end of Subaction timeout.
    - If Arb_Won is NOT set, then request Arb from parent.
        - Arb may be for Aynschronous or Isochronous transfer.
    - If granted Arb, then set Arb_Won flag.
    - Listen for Cycle_Start, and reset Arb_Won flag.
- Protocols and Agent/ieee1394, LL/ieee1394
    - Agent creates packets (determines size, transfer type, destination).
    - Agent handles Ack.
    - LL Ignore Split-Transactions.
    - LL handles Retry.
    - LL sends packets to Mac.

Summary

- Introduction to basic operation of IEEE1394
- Introduction to simulation concepts and classes in ns2.
- Mapping IEEE1394 protocols to ns2.
- Original Goals – See Demo
  - The whole point of the project was to evaluate bandwidth allocation of asynchronous .vs. isochronous transfer modes.
  - The mode to use is primarly determined by the application.
  - Comparing IEEE 802.3 .vs. IEEE 802.5 (Ethernet .vs. Token-Ring) would be more informative (but I wanted to get "under-the-hood" of ns2 and learn more about Firewire).

References

◇ **FireWire System Architecture, 2nd Edition (IEEE 1394a).** Anderson, Don. MindShare Inc. Addison-Wesley Inc. ISBN: 0-201-48535-4.

◇ **P1394b Draft Standard for a High Performance Serial Bus:** p1394b1-33.pdf

◇ **What's New About 1394b:** ppt1.pdf

◇ **Isochronous Resource Management:** br062r00.pdf

◇ **New Technology for 1394 (overview):** 1394ABoverview.pdf

◇ **IEEE 1394-1995 High Performance Serial Bus (overview):** 1394overview.pdf

◇ **NS-2 Documentation** (for implementation purposes): ns_doc.pdf