SMPTE 2022-6
A Practical IP Software Encode/Decode Implementation
Jean Lapierre
Senior Director of Engineering
and
Bruno Latorre
Software Project Manager

Matrox Video
Matrox = Hardware

• We develop PCIe cards and a powerful SDK
• Our customers build broadcast equipment
• We have more software engineers than hardware engineers!
Why we Decided to do a Study

• Customer asked why he still needed hardware
• Using inexpensive off-the-shelf hardware
• Racks of computers running virtual machines
• Vidtrans presenter said it would help with latency
The Project

- Selected a 1080i signal using SMPTE 2022-6
- Used a Windows machine and 10 Gig network card
- Lots of data to handle, 135000 packets/sec
- Quick response, new packet every 7.4 uSec
- Don’t use too many system resources
Complexity of SMPTE 2022-6

• Transition from a 10 bits aligned (SDI) to a 8 bits aligned (computer) space
• Building a proper SDI frame is complex
• Building UDP packets from scratch
Scope of Software Implementation

- Windows only, process video only on receive/transmit
- Receiver

- Transmitter
Setup

- CPU: E5620 @ 2.40GHz (2 processors, 16 logical cores), RAM: 12GB
- Network: Intel X520-2, Switch: 10GE
- Matrox XMIO3 I/O hardware
Sockets

• What is a socket?
• Windows uses sockets (Winsock)
  – Easy to implement
  – Close to the OSI protocol stack model
  – Filtering is packet based and at user space
  – Our experience with SMPTE 2022-6
Finding a Solution

• Bypassing some of the layers
• Winsock and raw sockets
• A solution without Winsock
  • Implement a protocol driver to bypass the protocol stack
  • Implement a more efficient packet filtering (kernel space)
  • Minimize context switches between kernel and user space
Our Solution

Transmit

Data

Application Builds Raw Ethernet Frames

Data Link Layer

Physical Layer

Receive

Data

Receive Raw Ethernet Frames

Efficient Packet Filtering

Application Layer

User

RTP, HBRMP, UDP, IPv4

802.3, …
Receiving and CPU Load

– What we got ...

– What we wanted ...
What We Did

• Distribute CPU load when receiving data
  – Receive side scaling (RSS)
Results Receiver

- 1 stream, multicast
- 3 streams, multicast
Issues with Transmission

• Bursting the network is prohibited
• Notification granularity of operating system is in milliseconds whereas what we need is microseconds!
• Need to implement a busy loop function
Results Transmitter

• 1 stream transmitter
Conclusions

• Will my customer still need hardware?
• 4 Input and 4 Output CPU usage
Conclusions

• Can everything run on generic hardware?
Conclusions

• Can everything run on generic hardware?
• Observations about virtual machines.
Conclusions

• Can everything run on generic hardware?
• Observations about virtual machines.
• Can IP video really deliver lower latency?
Conclusions

• Can everything run on generic hardware?
• Observations about virtual machines.
• Can IP video really deliver lower latency?
• The future needs compression
Conclusions

• Can everything run on generic hardware?
• Observations about virtual machines.
• Can IP video really deliver lower latency?
• The future needs compression
• What is in it for me?
Contact Information

• Jean Lapierre
• Senior Director of Engineering
• Matrox Video Product Group
• jean.lapierre@matrox.com