# Software compatibility and human interface for DV over IP

Tsuyoshi Hisamatsu
(ringo@sfc.wide.ad.jp)
Keio University

# Targets of this research

- Providing high quality video communication tool for MacOSX
- Providing user friendly interface

# Available video communication tools

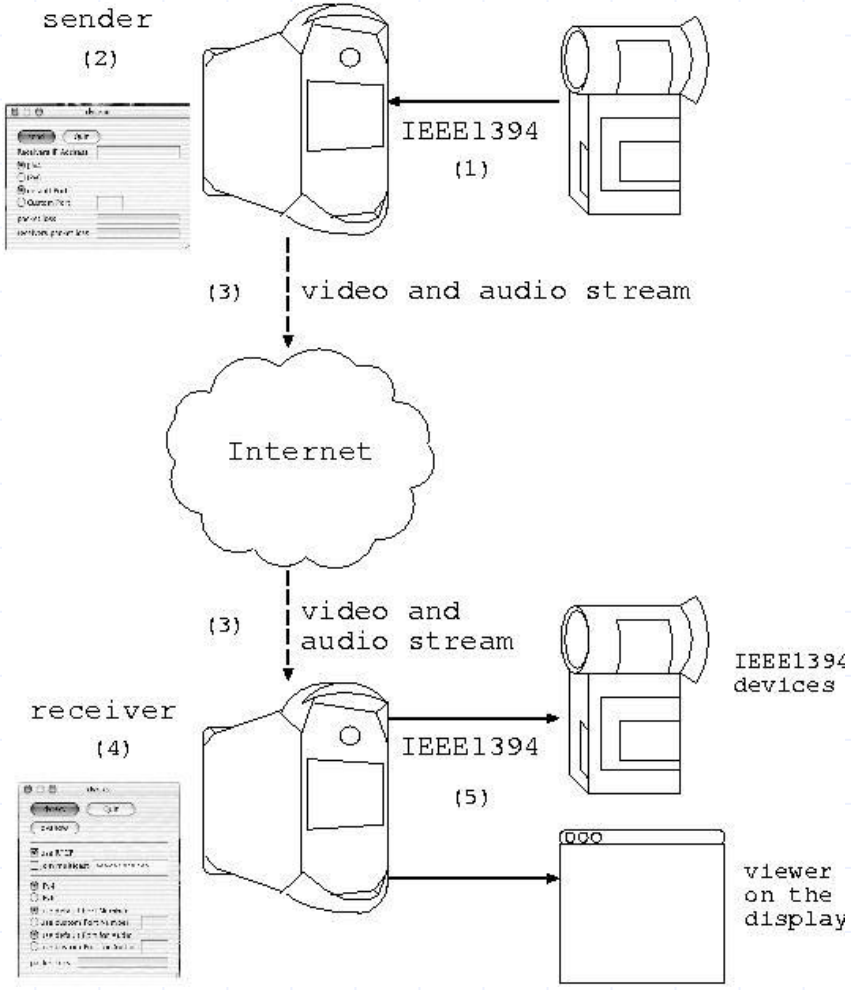
- DVTS
- Netmeeting
- CUSeeMe
- etc.

# Why DVTS?

- Transports TV quality video
- Uses consumer AV devices
- Open source
  - Written in C language

# What is DVTS? 1/2

- Digital Video Transport System
  - Transmitting DV streams from a connected Firewire devices
    - RTP(Real-time Transport Protocol), RTCP(RTP Control Protocol)
  - Supported platforms
    - BSD, Linux, MacOSX, Windows

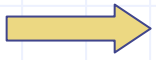


# What is DVTS? 2/2

- Uses consumer AV devices
  - DV cameras
  - Digital-Analog media converters
  - etc.

# The former DVTS 1/2

- Based on CUI(Character User Interface)
  - Using CUI to install
    - Ex.> Running “make”
    - Users must apply a patch to their mother OS and recompile its kernel
  - Using CUI to Configure



Such processes make DVTS hard to approach for traditional MacOS user

# The former DVTS 2/2

- Lacks the functions to playback video streams on the screen

# Machine & OS characteristics

- Each OS has different interface to interact with an application
  - Developpers should implement an Application based on culture of Mother OS
    - UNIX(Linux, *BSD) : CUI with keyboard
    - MacOS:GUI with mouse

# GUI Requirements for DVTS on MacOSX

- Installation
  - With simple “Drag & Drop” install wizard
- Running application
  - With simple mouse and keyboard operation

# Design of DVTS on MacOSX

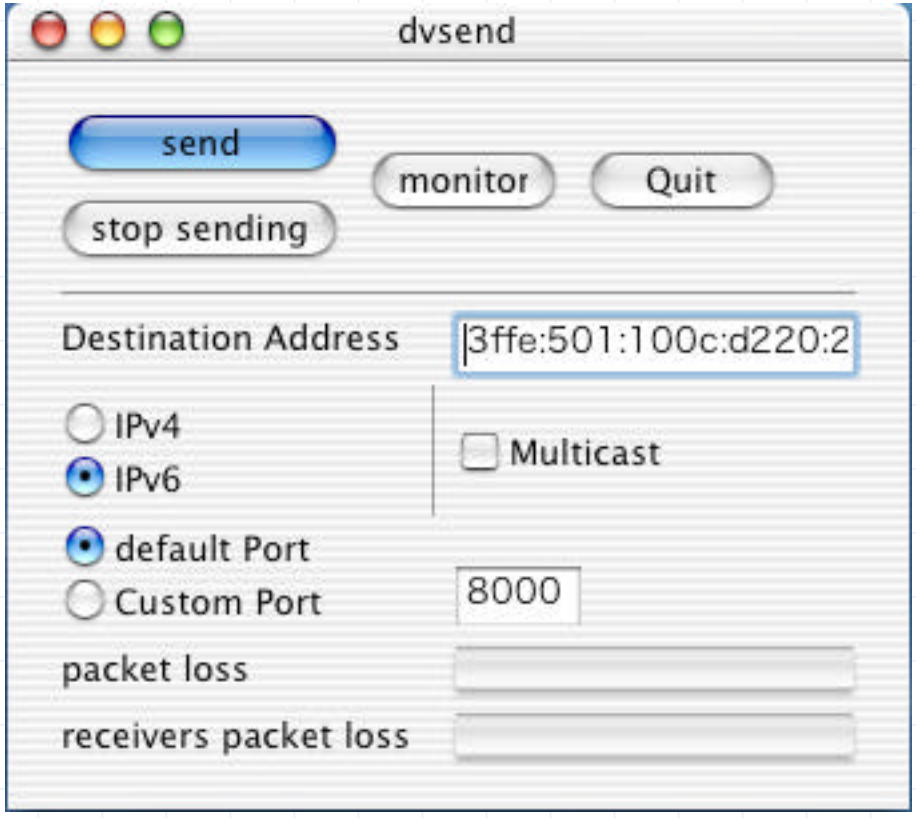
- Using DVTS without difficulty
  - Implementation based on GUI
    - To control DVTS only with the mouse operation except when inserting IP address
- Video playback window
  - To monitor video on the screen

# Design outline of DVTS on MacOSX (sender)1/4

- User connects FireWire devices to a sender machine

# Design outline of DVTS on MacOSX (sender)2/4

- Configure application by adding an option to the sender

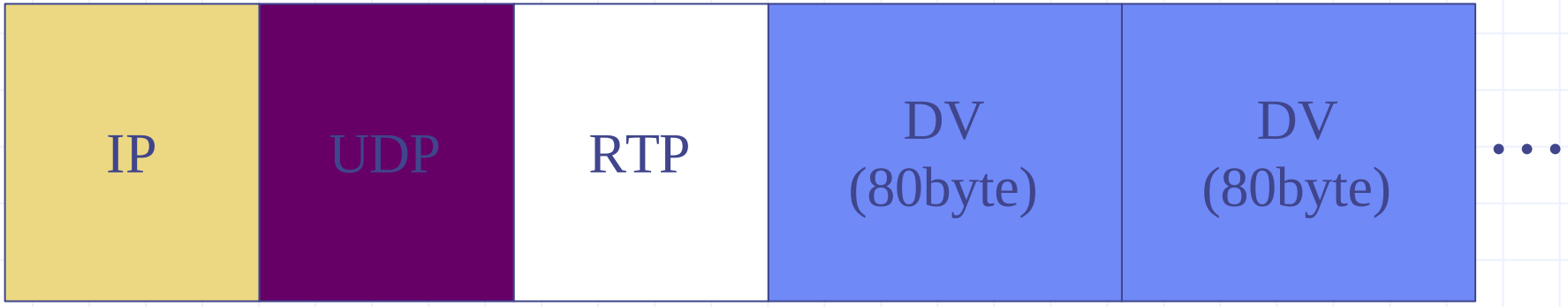


# Design outline of DVTS on MacOSX (sender)3/4

- The sender reads DV from the connected FireWire device.

# Design outline of DVTS on MacOSX (sender)4/4

- The sender adds IP, UDP and RTP header to the data

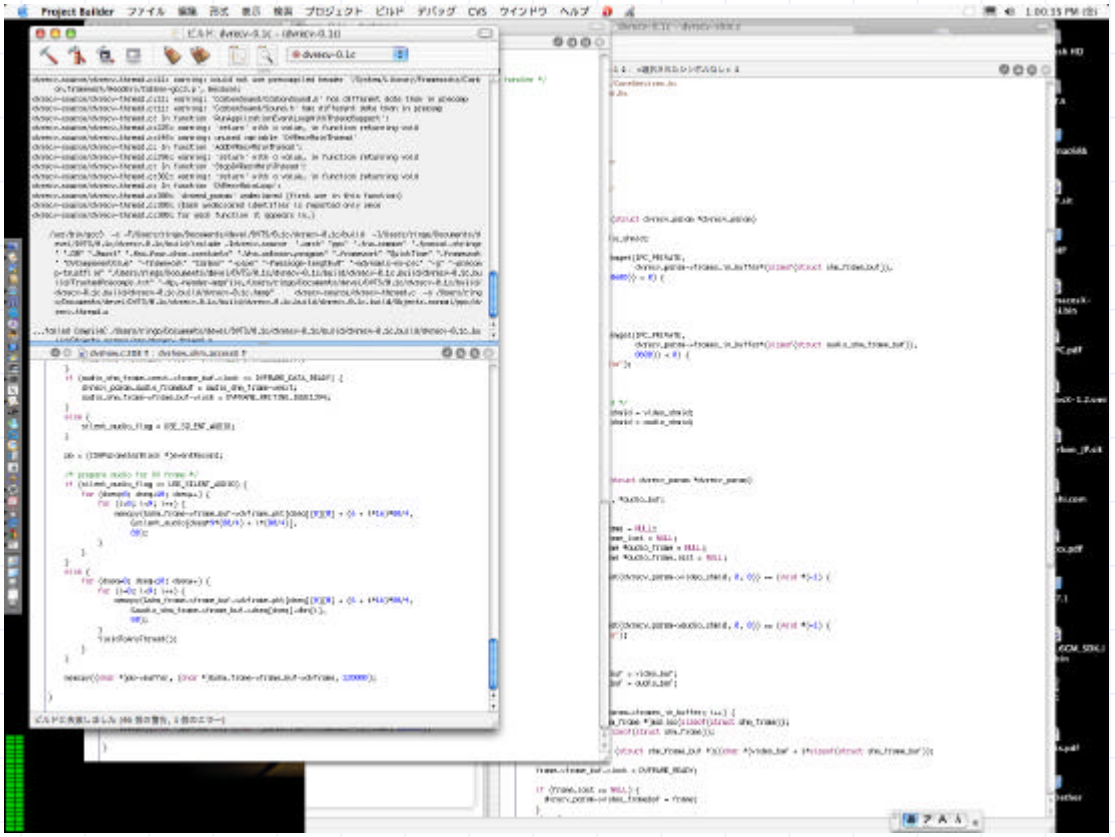


# Design outline(receiver)

- User sets options on the receiver system
- Select output device, the connected FireWire device or the viewer

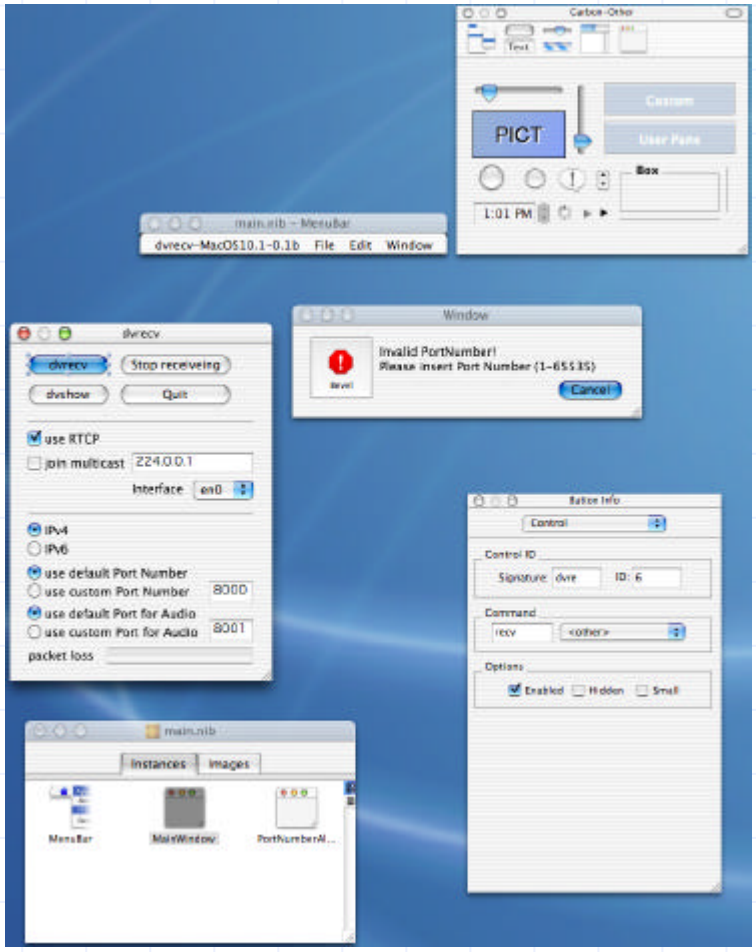
# Tools for implementing new DVTS 1/2

- Project Builder



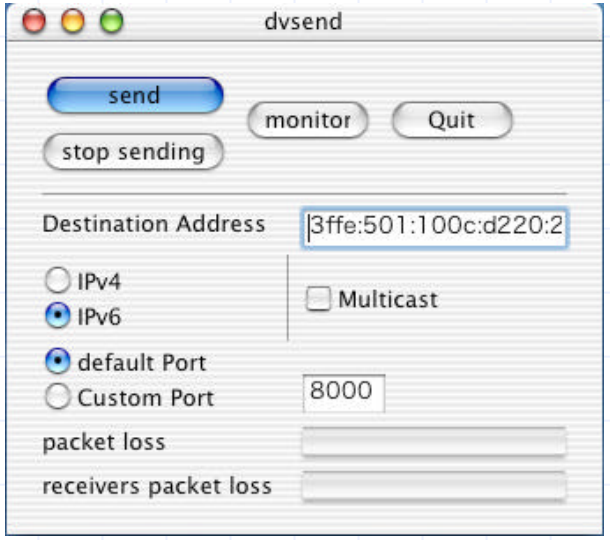
# Tools for implementing new DVTS 2/2

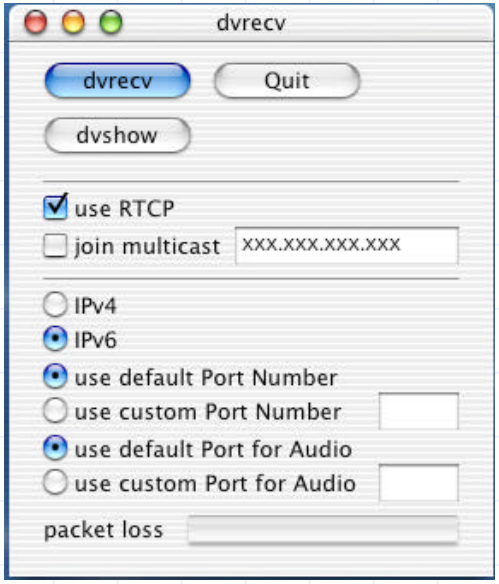
- Interface Builder



# Implementation (interface)1/4

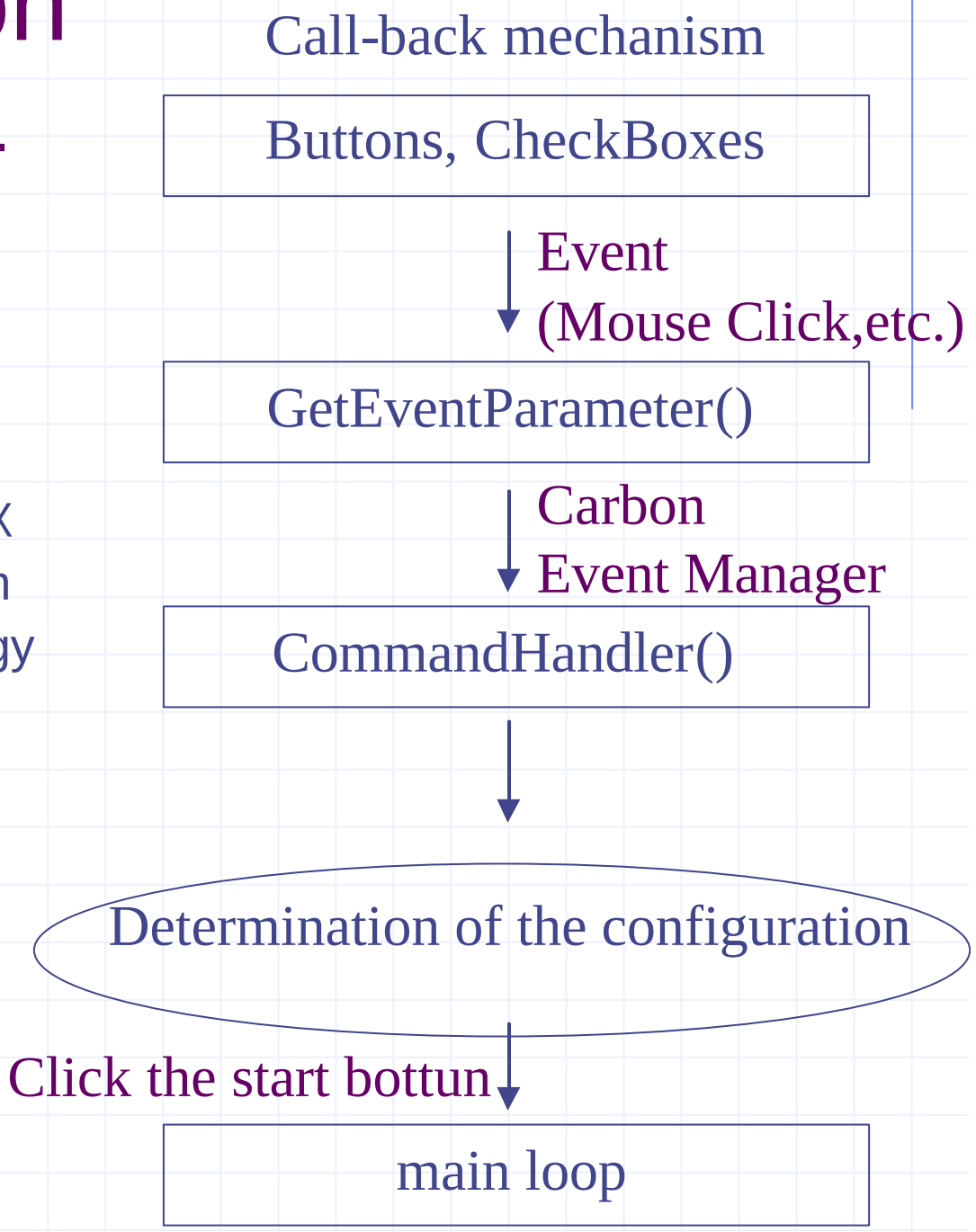
| | |
|---|---|
| Sender | ◆Start sending packet<br>◆Quit Application Loop<br>◆Input IP Address of receiver<br>◆Send using IPv4 or IPv6<br>◆Use default or custom Port Number |
| Receiver | ◆Start receiving packet<br>◆Quit Application Loop<br>◆Start drawing received movie (dvshow)<br>◆Join multicast<br>◆Receive using IPv4 or IPv6<br>◆Use default or custom Port Number |



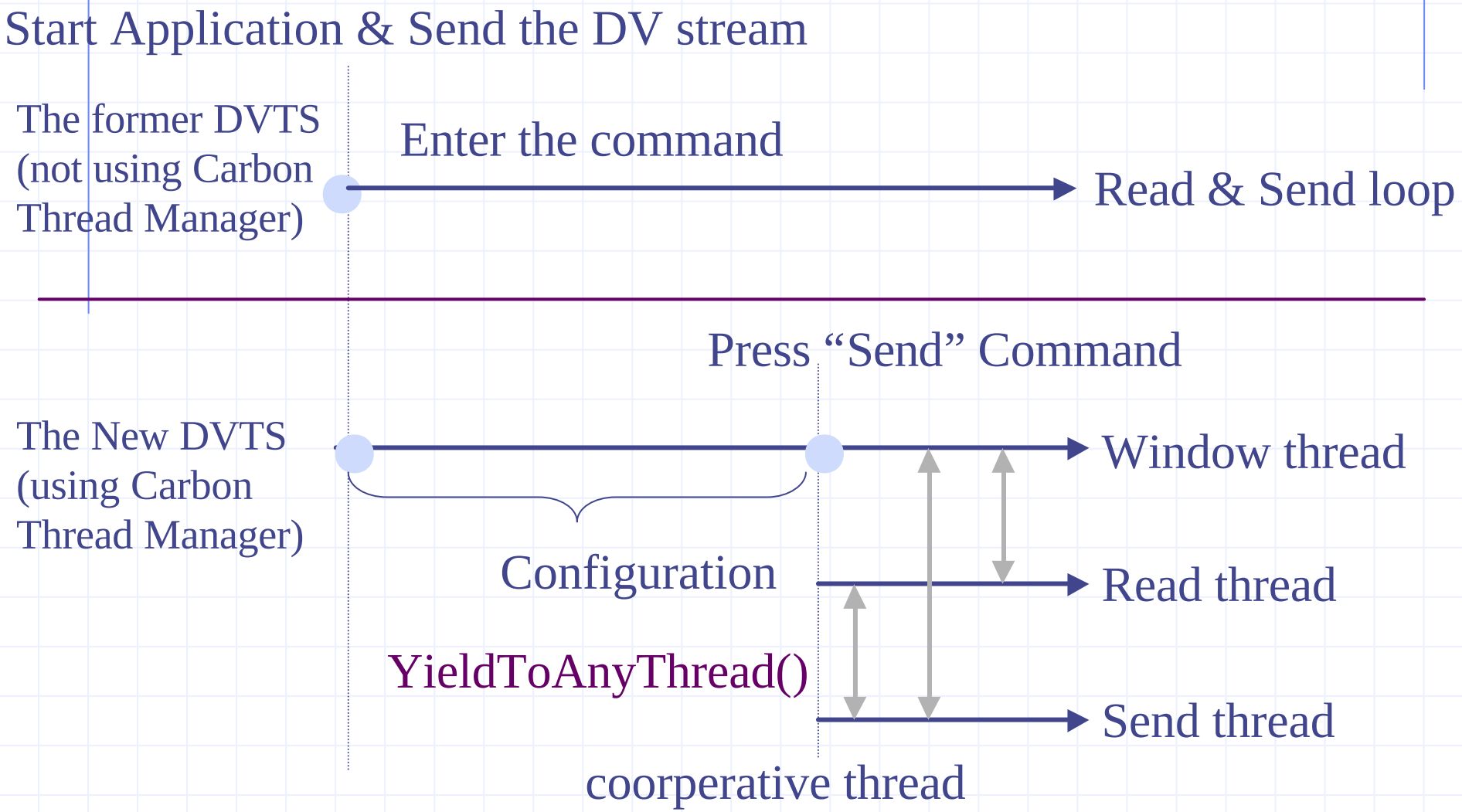



# Implementation (controller)2/4

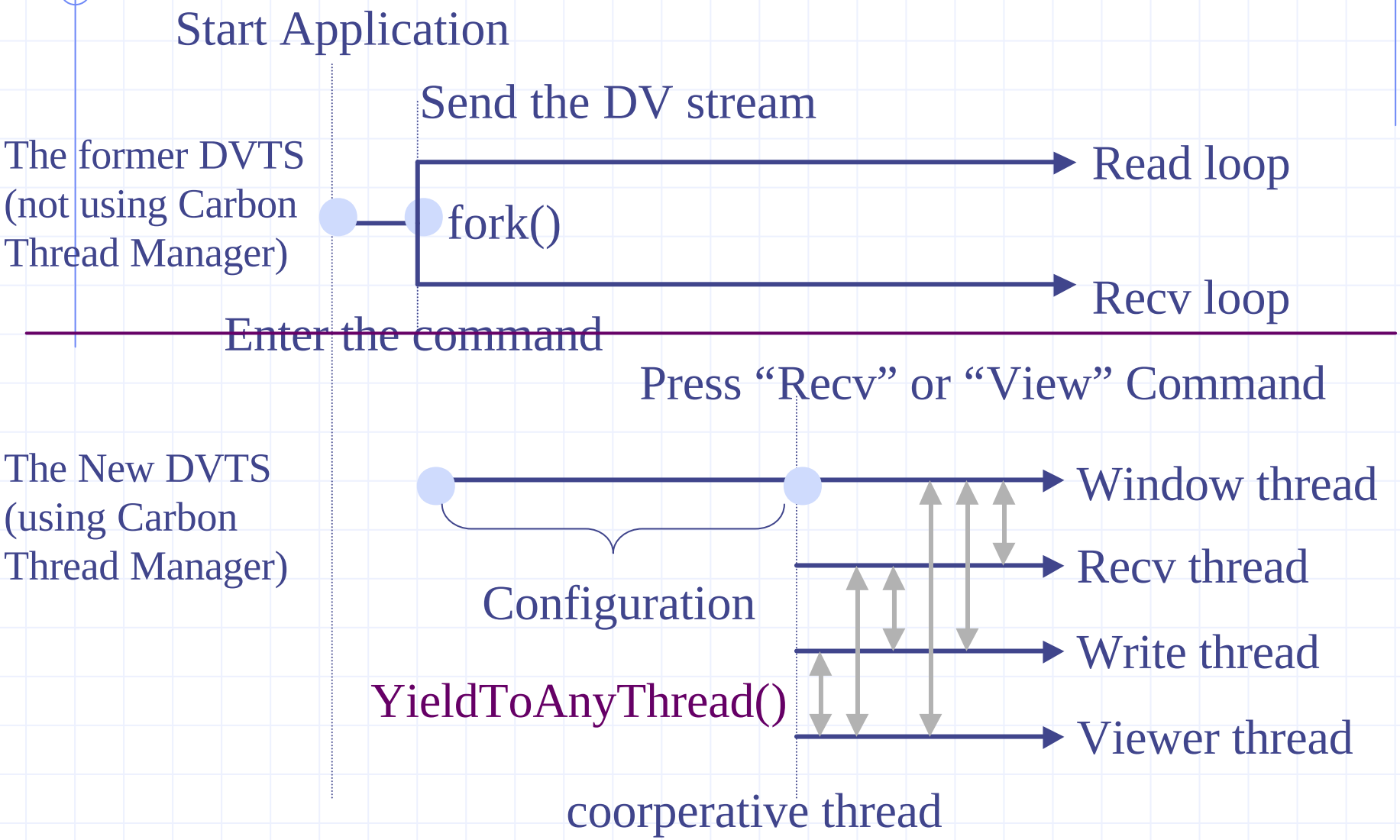
- Carbon
  - C programming interface for MacOSX
  - Call-back mechanism
  - Usable the technology of MacOSX
    - Memory protection
    - Preemptive multitasking
    - Dynamic resource assignment
    - etc.



# Implementation (controller on sender)3/4

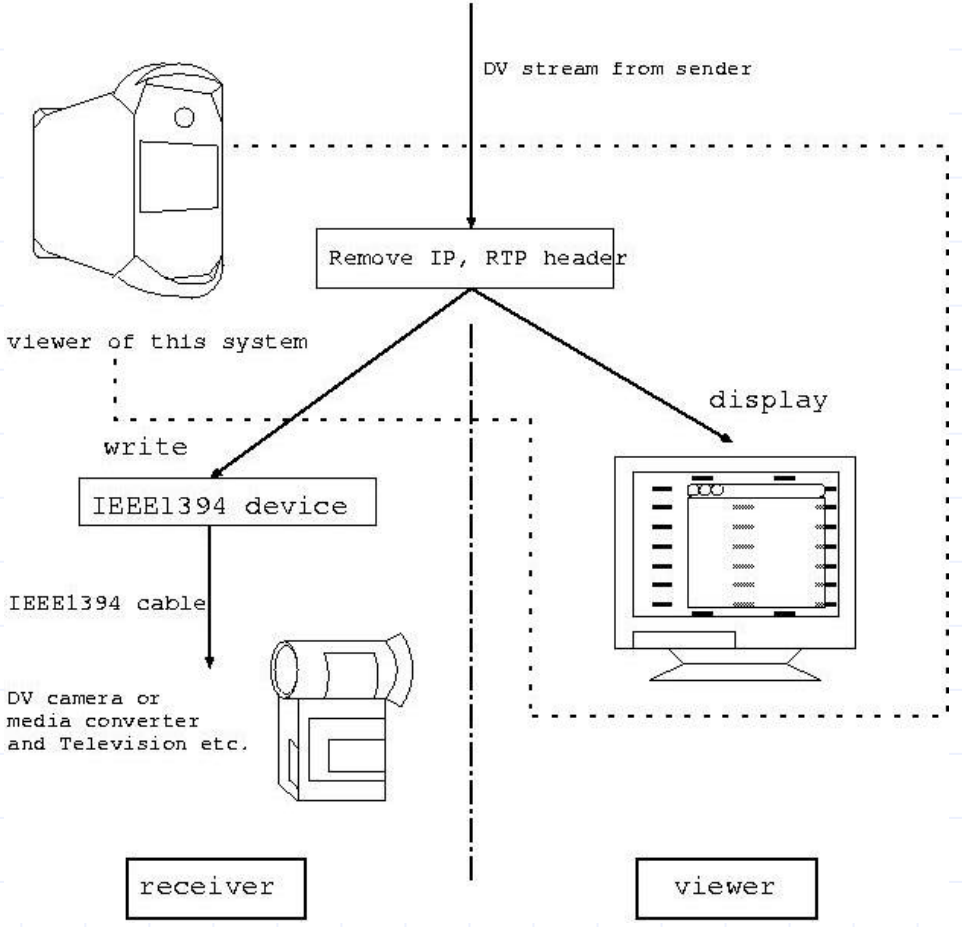


# Implementation (controller on receiver)4/4



# Implementation (video playback window)

- QuickTime framework
  - Movie Toolbox
- Integration receiver and viewer
  - The output of the DV stream will be redirected from FireWire devices to the viewer window



# Comparing the former DVTS to this system

| Steps | The former DVTS | This system |
|---|---|---|
| 1 | Downloads the source code from the site | Downloads the diskimage from the site |
| 2 | Uncompresses the archive and extract package manually | Click diskimage file stored in the directory |
| 3 | Enter“./configure”in the extracted directory | Drag the system from “disk image” and drop it to the hard disc |
| 4 | Enter“make”to construct a binary | Click the application to run the sender |
| 5 | “make install” to install the application to the application to the binary directory | Click the application to start receiving |
| 6 | “dvrecv -h” to see the command line options | |
| 7 | The user manually adds the option to the command, for example “dvrecv -4LR” | |

# Conclusion

- Provides standard MacOSX interface
  - Concealing Unix command line input method
  - Easy-to-use high quality video transport system
- Monitoring mechanism of the video image

# Impression of porting DVTS

- Porting from UNIX ware(written in C language) to MacOSX is relatively simple
  - However
    - Unsatisfied documents on Carbon
    - Distinctive threads when using GUI

# Release

- Still implementing the viewer
- Still debugging others

- Scheduled for release on February

  http://www.sfc.wide.ad.jp/DVTS