

# QEMU USB Device Emulation Through USB/IP

Scott Tsai

<http://scott.tw>

# QEMU Target Markets

- [Server Virtualization](#) (system emulation, kvm)
- [Remote Desktop Virtualization](#) (Citrix)
- Local Desktop Virtualization (Vmware Workstation)
- Embedded SDKs
  - [Android SDK](#) (system emulation, tcg)
  - [Symbian SVP](#): has python plugins for peripherals
  - [Maemo SDK](#) Scratchbox (user mode emulation)
  - [Andes OSDK](#)
- Other Applications
  - [qemu-systemc](#) -- SystemC devices connected through AMBA to ARM or PCI to x86
  - [Argos](#) -- extends Qemu to enable it to detect remote attempts to compromise the emulated guest operating system. Using dynamic taint analysis it tracks network data throughout execution and detects any attempts to use them in an illegal way. When an attack is detected the memory footprint of the attack is logged.

# QEMU Code Base

- Virtual Hardware, ex: hw/usb-serial.c (FTDI FT232BM)
- Protocols suited to the virtual hardware, ex: qemu-char.c connect emulated serial to telnet or stdio
- TCG: Tiny Code Generator dynamic instruction translator
- KVM: hardware assisted CPU virtualization
- Proven to be both interesting and easy to modify, proof: [all the forks](#)

# QEMU as USB Host

- GregKH's driver tutorial uses the Vernier GoTemp! USB thermometer as an example
- (qemu) info usb
- (qemu) info usbhost
- (qemu) usb\_add host:08f7:0002
- I wrote emulation for this peripheral in qemu: usb-gotemp
- *Show usb-gotemp Notes (~/.work/qemu/THERMOMETER)*

# QEMU as USB Device

- [\[Qemu-devel\] USB proxy the other way \(usb-gadgetfs.patch\)](#) -- [Andrzej Zaborowski](#), 24 Apr 2007
  - qemu USB slave support: `usb_slave_info_s`
  - host <-> guest USB packet injection interface: `gadgetfs`
  - guest hardware interface: `s3c24xx_udc`
  - `openmoko`
- `dummy_hcd`: host controller glued to gadget interface, one device limitation fixable
- `Gadgetfs`: a rather special purpose kernel <-> user space API
- I decided to try using `USB/IP` as the kernel <-> user packet injection interface

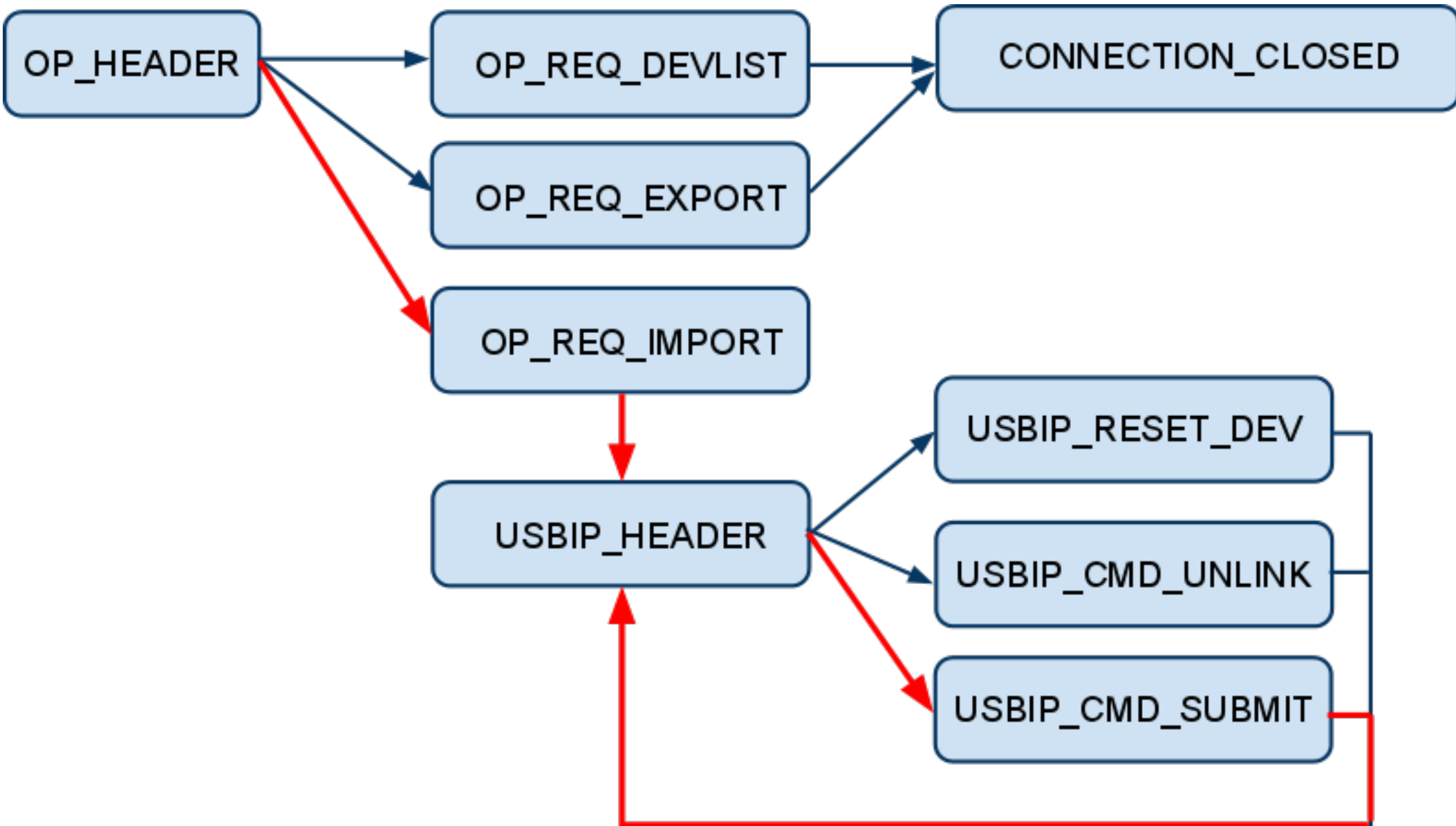
# USB/IP

- [USB/IP - a Peripheral Bus Extension for Device Sharing over IP Network. Takahiro Hirofuchi, Eiji Kawai, Kazutoshi Fujikawa, and Hideki Sunahara.](#) **(USENIX 2005 FREENIX Track Best Paper!)**
- <http://usbip.sourceforge.net>
- Kernel Space: linux-2.6/driver/**staging**/usbip
  - vhcd-hcd.ko: has virtual hub
- User Space: [lepton-wu: Initial opensource windows client ready now](#)
- As we shall see, the protocol is somewhat "implementation defined"

# USB/IP Demo

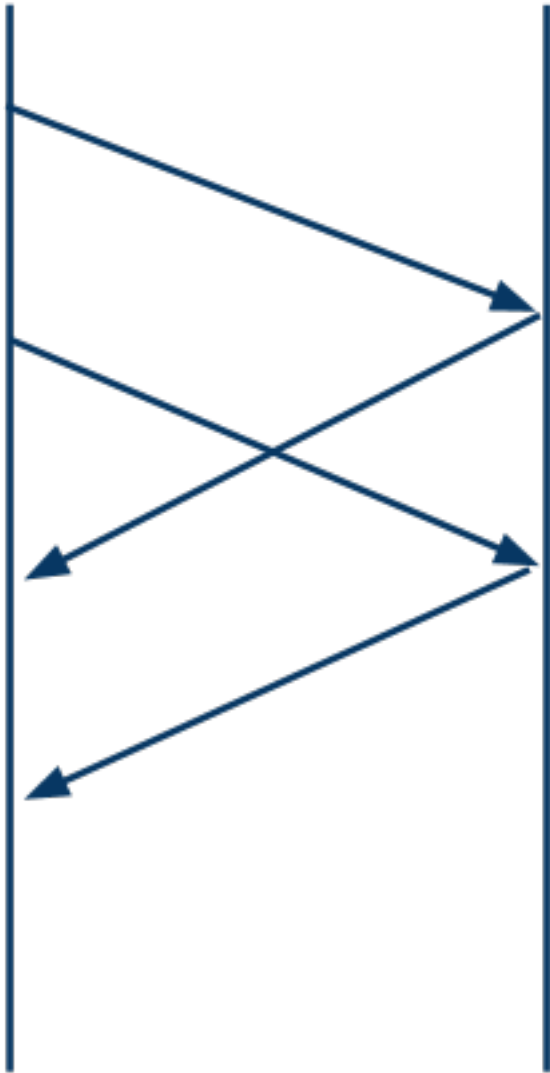
- server \$ usbipd --debug
- server \$ usbip\_bind\_driver --list
- server \$ usbip\_bind\_driver --usbip 7-1
- *Build linux kernel modules: CONFIG\_{STAGING, USB\_IP\_COMMON,USB\_IP\_VHCI\_HCD}=m*
- client \$ insmod vhcd-hcd.ko
- client \$ insmod usbip\_common\_mod.ko
- client \$ usbip --list server
- client \$ usbip --port
- client \$ usbip --attach server 7-1

# USB/IP Server State Machine

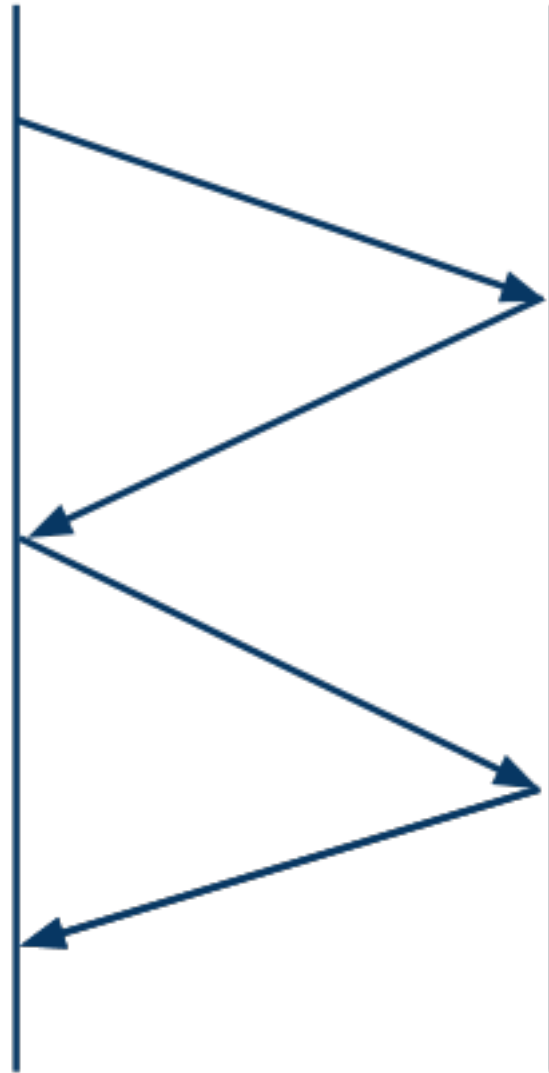


# TCP\_NDELAY: Nagle Congestion Control

TCP\_NDELAY



DEFAULT



# Current Status

- Non-blocking USB/IP server in qemu partly implemented
- Need to decide which USB peripheral controller to hook to.  
I'll probably work from qemu-neo1973
- Getting the right kernel tree, kernel config, qemu tree combination is sometimes non trivial

# Challenges to Making it Just Work<sup>(TM)</sup>

- VHCI: error handling when the USB/IP server hangs up:  
"usbip: vhci\_hcd: device ffff8801116051a8 seems to be still connected..."
- VHCI: allow attaching to a unix domain sockets in addition to TCP connections
- Security: USB packet injection permissions, allow for console user, PolicyKit?
- USB 3.0: unclear impact on USB/IP protocol (USB/IP currently only have weak)