



Provided by the author(s) and NUI Galway in accordance with publisher policies. Please cite the published version when available.

Title	PTP-IP a new transport specification for wireless photography
Author(s)	Corcoran, Peter M.
Publication Date	2005
Publication Information	P. Bigioi, G. Susanu, E. Steinberg, P. Corcoran, (2005) " PTP-IP a new transport specification for wireless photography", IEEE Transactions on Consumer Electronics, Vol. 51, No. 1, pp. 240-244.
Publisher	IEEE
Item record	http://hdl.handle.net/10379/295

Downloaded 2021-04-06T01:15:20Z

Some rights reserved. For more information, please see the item record link above.



PTP/IP - A New Transport Specification for Wireless Photography

Petronel Bigioi, George Susanu, Eran Steinberg and Peter Corcoran

Abstract – PTP (*Picture Transfer Protocol ISO-15740*) is an *International Standard for connectivity and interface of digital photography devices*. In this paper we describe how PTP can be extended to work over generic TCP/IP networks. As digital cameras and printers becomes wireless, PTP/IP will provide the necessary flexibility of applications infrastructure for networked digital photography¹.

Index Terms — digital camera, DSC, digital printers, connectivity, PTP, PTP/IP, wireless, PictBridge, ISO-15740.

I. INTRODUCTION

The Picture Transfer Protocol (PTP) [1] was developed to address the necessary industry driven requirements for two way communication between a digital still cameras (DSC) and external devices such as printers, host desktop computers, hand held devices etc. PTP is an internationally published standard (ISO-15740) designed and refined as part of the electronic photography group, WG-18, of the photography technical committee, TC-42 of the International Standards Organization as ISO.

This activity was successfully conducted at an industry-wide level and involves practically all major digital camera manufacturers and additionally, imaging chipset manufacturers, and digital imaging and operating systems software companies. As of late 2004, at least 95% of all consumer digital cameras include PTP as the sole means of camera connectivity or in parallel with USB mass storage solutions.

An important validation the PTP standard is the fact that PTP became the underlying protocol for PictBridge [3] (Standardized by CIPA - Camera and Imaging Product Association, Japan) which is the high level protocol for direct connectivity between cameras and printers.

Recently, Microsoft ® Corporation announced MTP (Media Transfer Protocol) which is a superset of PTP standard. The ability of PTP going over networking transports enables also the MTP to work over IP networks. This enables a number of new usage models for portable media players and devices in the context of wireless 802.11 networks.

PTP include two components which may not be symmetrical: an *initiator*, namely the desktop device or the printer and a *responder*, namely the digital camera.

The features supported by PTP include defined mechanisms for image referencing, for command responses, events, device properties, datasets, and data formats to ensure interoperability. In addition to standard operations and formats, optional operations and formats, as well as extension mechanisms, are supported allowing manufacturer-specific customizations of imaging device behavior while still conforming to the PTP standard.

II. THE NEED FOR ADITIONAL TRANSPORTS

An important guideline for the WG-18 working group has been to define PTP as a transport independent protocol.

Thus in theory solutions are available for PTP over multiple transport layers. For example, discussion of the broader applicability of PTP was presented at ICCE 2002 [4]. This paper introduced some preliminary ideas for implementing PTP over Bluetooth and TCP/IP networks. Other reference of PTP includes PTP over 1394.

Practically, due to the fact that at most consumer level cameras in the last 3 years are USB based, current product support of PTP is limited to USB (1.0, 1.1 and 2.0) devices. As part of related activities by the USB standards working group a *Still Image Device Definition* was presented in conjunction with the PTP specification [2]. This specification approved by the USB-WG became the basis for the implementation of PTP over USB. However, it is important to note that PTP was developed to be a much more flexible standard than the current USB based implementations would suggest.

As consumer electronic devices become wireless, there is an obvious need to support this trend in the digital imaging domain, in particular DSCs, Digital printers and supporting desktop devices. As such devices begin to require access to IP transport layers, most noticeably WLAN and TCP/IP based, there is a need to extend the capabilities of PTP to operate over such transport layers.

In this paper we present and discuss the specifications of a new PTP transport, namely PTP/IP. This transport is designed to transparently support a superset functionality over a dual TCP/IP network connection as is available for PTP running over a USB transport. This will provide a significantly more flexible infrastructure for digital photography applications, especially in the context of the growth of home WLAN networks.

III. THE SOLUTION

A. Transport Requirements

PTP is a transport independent protocol. Referring to the PTP Standard [1] Clause 7 the definition of the transport requirements, include:

¹ Petronel Bigioi is Director of Camera Technology at FotoNation Ireland Ltd, Galway (e-mail: petronel@fotonation.com).

Eran Steinberg is CEO of FotoNation Inc, San Francisco, CA (e-mail: erans@fotonation.com).

George Susanu is a Senior Research Engineer at FotoNation Ireland Ltd, (e-mail: george@fotonation.com).

Peter Corcoran is the Director of IP at FotoNation Ireland Ltd, (e-mail: pcor@fotonation.com).

- : Disconnection Event – this clause requires that the transport layer has to report device disconnection to higher layers. The implementation of the transport layer has to generate a *DeviceDisconnection* notification. ([1] Clause 7.1)
- Reliable, Error Free Channel – this clause requires a reliable, error free channel, meaning that those issues have to be solved in the transport level protocol. The PTP is not dealing with them. ([1] Clause 7.2)
- Asynchronous Event Support – this clause requires support for Asynchronous Events. The transport has to provide a separate physical or logical way of transferring the events asynchronously to transactions. ([1] Clause 7.3)
- Device Discovery and Enumeration – this clause requires transport support for device discovery and enumeration. ([1] Clause 7.4)

The PTP/IP transport fulfills all of the above requirements but one –Device Discovery and Enumeration (clause 7.4). Due to irreconcilable reasons beyond the scope of the Specification, it was unwise to select a single device-discovery and service-enumeration protocol. Instead, the decision was to separate the device-discovery from the internals of the PTP/IP specifications. An informative annex describes how the PTP/IP discovery can be implemented using Rendezvous. However, discovery is by no means limited to Rendezvous. Current implementations include manual (no-discovery), proprietary UDP based discovery, UPnP™ etc as described in Section E of this paper.

The PTP/IP transport specification is based on the use of TCP (Transfer Control Protocol) layer as its own transport layer, as shown in Fig 1. TCP (in the TCP/IP protocol stack) is the natural transport layer to provide reliable, error free data communication services to the PTP/IP layer. TCP is a stream based transport layer that provides multiple communication channels (TCP connections) and error free data delivery.

B. PTP/IP as part of the Communication Stack

PTP/IP supports the requirements of the PTP layer on a network stack. It was determined that mapping PTP onto the TCP layer using TCP stream sockets would provide a suitable means of communication between two PTP devices (i.e. the *Initiator* and *Responder* of PTP).

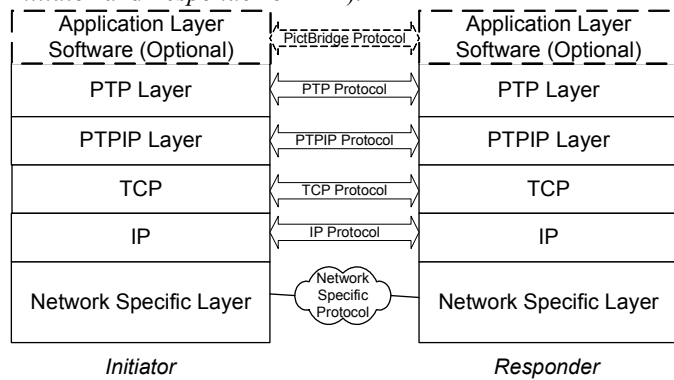


Fig 1: PTP/IP over TCP/IP

The place of the PTP/IP layer in the communication stack is presented in Fig1. The possibility to open multiple concurrent sockets between the devices would provide a mechanism to separate (i) the command, data & response (CDR) and (ii) the event-based, PTP primitives. The use of TCP is attractive from

two points of view: (i) support in almost all types of LANs (ii) provides error free data delivery and concurrent logical connections support.

C. PTP/IP Connection

In order for the PTP layer on the *Initiator* to be able to establish a session with the remote PTP layer on the *Responder* device, a PTP/IP connection needs to be first established. This consists of two TCP/IP connections (Fig 2), one for a CDR channel and one to provide an event channel. The former carries PTP commands, data and responses while the latter carries PTP events related to the CDR channel asynchronously.

If an *Initiator* requires multiple PTP sessions to the same *Responder* device, then multiple PTP/IP connections need to be established. In a networking environment, multiple *Initiators* may attempt to open connections with the same *Responder*. Thus special care needs to be taken in order for the *Responder* to make sure that dual TCP connections are associated with the same PTPIP connection.

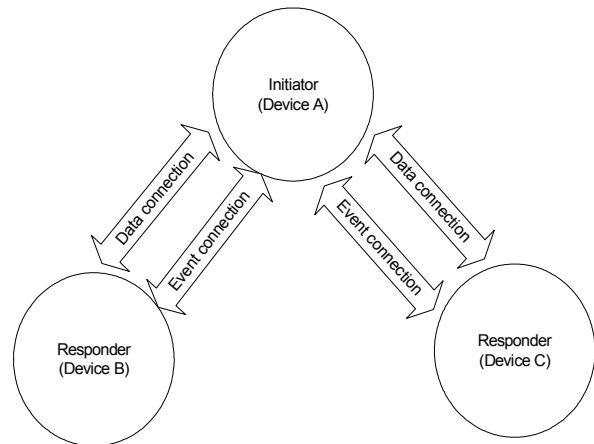


Fig 2: PTP/IP Connections

A PTP/IP connection should be fully established before any event and data communication can take place. This is described in Fig 3. The *Initiator* begins by establishing a CDR TCP connection with the *Responder*, (1) and immediately afterwards sends the *Init_Command_Request* PTP/IP packet (2) that will contain its unique device identifier and friendly name.

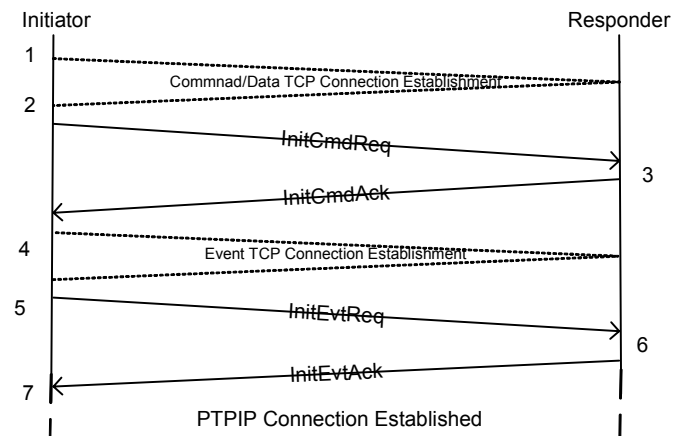


Fig 3: PTPIP Connection Establishment

The *Responder* may answer either with an *Init_Command_Ack* which returns a unique connection number, its unique device identifier and its friendly name, and lets the *Initiator* know that it can continue with the connection establishment procedure; or with an *Init_Fail* which will deny the *Initiator* access to *Responder* services and will close the CDR channel. When the *Initiator* receives the *Init_Fail* PTP/IP packet, it will also close its end of the CDR channel.

If the *Initiator* received an *Init_Command_Ack* packet it will initiate the event channel TCP connection (4), and the *Initiator* will then send the *Init_Event_Request* PTP/IP packet (5) which carries the previously assigned connection number. The *Responder* uses this number to associate the two TCP connections, CDR and event, with the same PTP/IP connection and the same PTP session. In response, the *Responder* will send an *Init_Event_Ack* packet on the event channel (6).

The PTP/IP connection is now considered established (7) and further data communication can take place.

Note that as opposed to USB where a responder is limited to a single initiator (a camera can only interact with one tethered device), in the wireless world, the responder, such as a digital camera, a concurrently interface with more than a single initiator (e.g. a camera may communicate with a PC to download images and with a wireless printer to make prints.)

D. Transport Configuration

PTP/IP is executed on top of TCP/IP. Correct implementation of the addressing is a key component of the implementation. The *Responder* and/or *Initiator* must obtain valid IP addresses using one of the following methods:

- Manual configuration – the IP address and other associated information is configured manually by the user, to reflect the topology and address schema of the local area network in which the imaging devices will function
 - DHCP server running in local network. In this case, the imaging devices should implement a DHCP client that will automatically obtain an IP address from the DHCP server running in the system. There may be a case where DHCP is not available (such a camera connecting to a single PC at home via ad hoc wireless connection, or a LAN that does not implement DHCP)
 - Dynamic Configuration for example IPv4 Link-Local Addresses (v4LL) - standard that describes how an IP address is automatically configured in order for a new device to work in a local area network, without having to setup a DHCP server
- PTP/IP specification doesn't state nor limits the method to be used. It is up to the specific platform what method will be used to configure the TCP/IP stack for normal functionality.

E. Device and Service Discovery

Although the PTP/IP transport for the PTP protocol opens a number of interesting new usage scenarios for an imaging device, if no device and service discovery protocol is used, then the PnP features of a device will be lost.

The PTP standard requires device discovery and enumeration. In PTP/IP this requirement can be fulfilled by using a few approaches:

- Manual Configuration – instruct the *Initiator* manually where to connect (*IP:port* for the remote *Responder*)

- Rendezvous – a device discovery and service discovery mechanism defined by Apple® Computer, Inc.
- UPnP – the family of protocols defined by UPnP Forum [5] to facilitate automatic device configuration, service discovery and invocation. Only the device and service discovery features (SDP) of UPnP are to be used.
- A custom device and service discovery mechanism, specific for PTP/IP, based on UDP protocol. Each *Responder* (Camera) arriving in a new network and periodically thereafter will use UDP protocol to broadcast a defined packet that will contain information about itself (Such as supported protocol – i.e. PTP, port numbers where it waits for incoming connections, its IP address, and other useful information). All the devices that may need to act as *Initiators* will run background processes that would intercept this message and configure the PTP/IP layer.

The PTP/IP layer can work without a service discovery protocol, therefore specifying such service is outside the scope of the specifications of the PTP/IP.

F. Device Bonding

The PTP/IP protocol also provides support for device bonding, better known as device pairing. The protocol provides transfer mechanisms for device unique identifiers as well as device friendly names during the PTP/IP connection establishment.

Each device in PTP/IP context will have a unique GUID (16 bytes unique number) that will represent the device in the communication process. The exchange of their GUIDs, allows to the application layer on each PTP/IP device to implement device level access mechanisms. *Initiators* have the ability to filter out the *Responders* they are not configured to connect to and *Responders* have the ability to reject a PTP/IP connection request from unknown *Initiators*.

This mechanism is required in order to avoid accidental picture transfer to an unauthorized *Initiator* for example when a *Responder* (digital camera user) enters the wireless-proximity of an *Initiator* (the neighbor's printer...).

Despite what service discovery and enumeration is used, it is recommended that the device GUID will be also part of the announcement packets, so the *Initiator* will be able to filter (if it wants) specific devices, and generate selective notifications.

G. Implementation

An initiator and responder reference designs as well as devices based on PTP/IP are currently in initial stages of production. Such devices have implemented PTP/IP over TCP/IP over 802.11b/g.

A possible PC software architecture for PTP related services is presented in Fig 4. The operating system running on the PC is Windows XP. Windows XP has its own imaging architecture (WIA – Windows Imaging Architecture). The PTP logic is implemented as a standard WIA minidriver, for integration with the imaging support from the operating system. Such reference software is available by the authors.

The PTP Transports (only PTP-USB and PTP/IP are presented) are accessed through a custom transport API that is the same for all the available transports. In this way, same PTP logic can be used to access different types of PTP devices (USB devices, IEEE1394 devices, Bluetooth devices or PTP/IP devices).

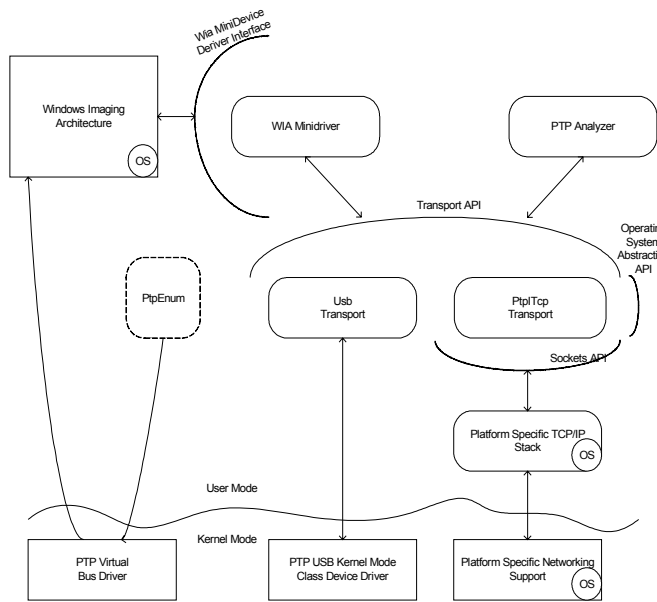


Fig 4 PC WinXP Software Architecture for PTP

For transports that don't have built in device discovery and enumeration, the authors have implemented an external plug and play manager (PtpEnum) that can accommodate various plug and play mechanisms, external to the transports. This is the case with the PTP/IP transport. PtpEnum has two main functions: plug and play manager, in charge with receiving and enumerating all PTP devices in the local LAN and filtering of those devices (based on GUID of the devices and/or the device friendly name).

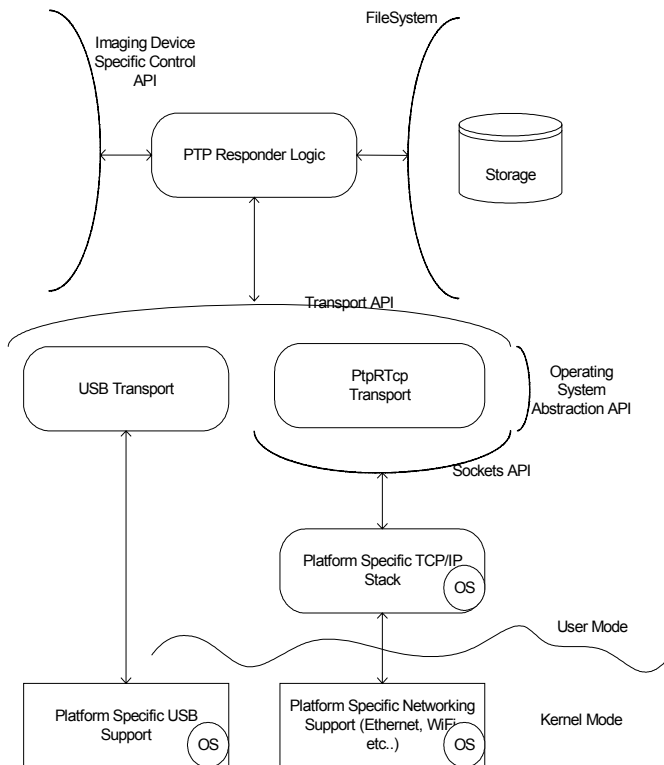


Fig 5: Responder Software Architecture

If a device arrival notification is received (either using Rendezvous™ or the custom PnP mechanism), PtpEnum will

decide if the device is or not allowed to be reported. If the device is allowed to communicate with the host PC, then PtpEnum will notify a kernel mode bus device driver that will generate the Windows XP specific PnP events that will trigger the creation of a local imaging device (WIA will load the PTP WIA mini-driver). All the XP imaging applications (including the Explorer) will be able to work with the newly arrived device.

The virtual bus device driver and its children don't contain any PTP logic at all. Their existence is somehow forced, but necessary for an easy, seamless integration with Windows XP imaging architecture.

Possible software architecture for PTP related services in a Responder device is presented in Fig 5. The PTP Responder that implements the PTP logic can work with different transports, through the transport API. Same PTP logic can be used when the imaging device connects over the USB cable or over a networking interface (Ethernet or WIFI physical connections).

IV. CONCLUSIONS

The PTP/IP transport provides the description of how the PTP data structures need to be transported over a TCP/IP network protocol and is the foundation for establishing wireless camera and printer communication.

PTP/IP provides the necessary basis for direct printing to wireless printers as well as wireless connection to desktop devices.

A detailed specification draft of the specification is available by the authors [6].

Reference implementation as well as feasibility tests were conducted and successfully concluded by the authors.

Initial devices using PTP/IP were introduced by Fall-2004. PTP/IP is scheduled to be presented to Standard bodies by Winter-2005.

REFERENCES

- [1] PTP/ISO-15740, "Picture Transfer Protocol Specification", http://www.iso.org/downloads_it10.html
- [2] USB Device Working Group, "USB Still Image Capture Device Definition", http://www.usb.org/developers/devclass_docs/usb_still_img10.pdf
- [3] CIPA, "CIPA DC-001-2003 Digital Photo Solutions for Imaging Devices", http://www.cipa.jp/pictbridge/contents_e/03overview_e.html
- [4] P. Bigioi, G. Susanu, P. Corcoran and I. Mocanu, "Digital Camera Connectivity Solutions using the Picture Transfer Protocol (PTP)", *ICCE 2002 and IEEE Transactions on Consumer Electronics*, vol. 48, number 3, pp.417-427, August 2002
- [5] UPNP Forum <http://www.upnp.org>
- [6] PTP/IP Draft Specification – for review purposes only www.fotonation.com/



Petronel Bigioi received his B.S. degree in Electronic Engineering from "Transylvania" University from Brasov, Romania, in 1997. At the same university he received in 1998 M.S. degree in Electronic Design Automation. He received a M.S. degree in electronic engineering at National University of Ireland, Galway in 2000. He is currently Director of Camera Technology at FotoNation. His research interests include VLSI design, communication network protocols, digital imaging and embedded systems. He is a member of IEEE and a member of technical committee of ICCE.



George Susanu received his BS degree in microelectronics from “*Kishinev Politechnical Institute*”, Kishinev, Republic of Moldova. With an experience of eight years in RTOS and embedded systems, having a wide experience in C/C++ programming he currently is working as senior R&D engineer with FotoNation Ireland Ltd. His research areas include real time operating systems and device connectivity.



Eran Steinberg received his BSc. in mathematics from HU, Jerusalem, Israel. He received his BFA in Photography from School of Visual Arts, NY. He received a MSc. in Imaging Science from RIT, Rochester, NY. With a vast experience of over 15 years in industry, project leader of ISO/TC42/WG18 – 15740 & 12 granted patents. he is currently CEO of FotoNation. His research interests include digital image processing and connectivity.



Peter Corcoran received the BAI (Electronic Engineering) and BA (Math's) degrees from Trinity College Dublin in 1984, where he continued his studies at TCD and was awarded a Ph.D. in Electronic Engineering for research work in the field of Dielectric Liquids. In 1986 he was appointed to a lectureship in Electronic Engineering at NUI, Galway. His research interests include microprocessor applications, home networking, digital imaging and wireless networking technologies. He is a member of IEEE.