International Journal of Computer Networking,
Wireless and Mobile Communications (IJCNWMC)
ISSN 2250-1568
Vol. 3, Issue 1 Mar 2013, 171-190

MOBILE PRINTING SOLUTIONS

ARUL SELVAN R[1] & K. PORKUMARAN[2]

[1]Software Engineering, Novell, Bangalore, India

[2]Principal, Dr NGP Institute of Technology, Coimbatore, India

ABSTRACT

The printing paradigm has changed with the arrival of the number of hand held devices. We identified the various mobile printing data types, printing methodologies, the unsolved problems in mobile network printing and the remote printer discovery for the hand held devices. And we developed solutions and proposals for the problems identified. In addition to this, we explored and the next generation printing solutions like driverless printing, location based print services and printing as a service.

KEYWORDS: Driverless, Rendering, Mobile, Printing, Discovery, Driver

INTRODUCTION

The recent growth in popularity of laptops and other small devices such as Net Books and smart phones is changing the landscape of network printing. Where it was previously normal to only allow printing from in-house systems there is now a growing need to support casual printing from the devices consorted with the users. And there is no easy method to print from the hand held devices because the hand held devices do not have well established print solution through printer drivers. A printer driver is a computer program that allows higher-level computer programs to interact with hardware devices.

The printer driver is a software entity that directly supports a physical printer, exposes the printer capabilities to the application, and enables it to carry out its functions. Hardware vendors develop printer drivers that are specific to printer. Most printers require different printer drivers depending on the operating system, or work station, or devices. Image or document rendering, whether performed immediately or during print processing, is performed in the print driver.

OBJECTIVE

The objective of this paper is through light on how to eliminate printer specific drivers on work stations or other devices and produce printable data on devices where print graphics is absent. The print rendering pipeline and the print sub system varies across work station computers. The need for the driver increases since the number of work stations are increasing and more importantly, the graphics print quality varies across print sub systems. Most of the mobile computing devices do not have the print rendering pipe line and the drivers associated to the printer. They are dependent on external applications and print rendering systems for quality print needs. In this paper, the solutions for the above mentioned unsolved problems are discussed, and recommendations and proposals are made.

PRINTING LANGUAGES

Like any other computing device, printer has its own language to produce the print outs. A printer, as a hardware understands one or many language types such as PCL, HPGL, PS, ESC, PRN, UFR II and so on. Out of these languages, PCL and PS are widely used in the printing industry, while- HPGL is used primarily in plotter devices.

Print Control Language

PCL was developed by Hewlett-Packard as a simpler, faster and less expensive alternative to PostScript-based laser printers. PCL retains all those advantages today and has fewer features than PostScript and its descendant, PDF. - However, the vast majority of business documents do not require the power of PostScript. Fewer features means simplicity;- Simplicity is the biggest reason most workgroup printing is PCL and most printers sold today are PCL based.PCL is device dependent. This means that the drivers for this language utilize the printer hardware for creating some of the printed data, usually graphics data such as fill areas, underlines or fonts.

Hewlett-Packard Graphics Language

HPGL is the standard print format for HP plotters, i.e., the data stream sent to HP plotters. If you are plotting to a Hewlett-Packard plotter, you are using HPGL. Most CAD programs output HPGL data directly because they can print on HP plotters. HPGL is an open language vector graphics file format assuring optimal file size with very fast and accurate rendering. HPGL/RTL can be output by all CAD/CAM/CAE systems. All HP printers starting with the LaserJet include HPGL as an integral part of PCL support. RTL (raster transfer language) is a raster-based specification and a subset of standalone HPGL but not within PCL.

Printer Job Language

The Printer Job Language was also developed by Hewlett-Packard. Printer Job Language resides above the languages like Printer Command Language or Post Script. It can be used to set printer parameters such as switch the printer interpreter to the desired page description language, usually either PostScript or PCL and set the printer ready to receive either PS or PCL or Raster data.PJL is also used for status read back between the printer and the host computer. PJL offers application programs an efficient way to remotely control printers. Using PJL, developers can provide applications with the ability to monitor printer status, request the printer model and configuration, change control panel default settings, modify control panel messages, and more. It is more appropriate to say that PJL is a printer control language rather a printing language, because it is primarily used for managing the printers.

Post Script

PostScript is a programming language optimized for printing graphics and text, irrespective of whether it's on paper, film, or CRT. It is also a page description language, introduced by Adobe. The main purpose of PostScript is to provide a convenient language to describe images in a device independent manner. Postscript is a programming language that was designed to specify the layout of the printed page. Postscript printers and postscript display software use an interpreter to convert the page description into the displayed graphics.

Other Languages

Ultra-Fine Rendering: Ultra-Fine Rendering is Canon's new proprietary super-fast rendering technology. It is designed for use with applications such as Microsoft Office.

Prescribe: Prescribe is a page description language created by Kyocera. The advantage of this language is that you may embed it in any other currently selected printer emulation on the Kyocera machines.

Samsung Printer Language: Microsoft and Samsung developed the Samsung Printer Language (SPL). This enables the windows GDI language to be converted into bitmap while printing. The image is rendered during printing process, which greatly reduces the amount of processing power required from the PC. Though many standards are evolved

overtime, Post Script and PCL dominated the printer hardware world and have become the prevalent languages that are supported by all manufactures.

PRINT DATA TYPES

Typically the printable data falls into three broader categories namely text, image and algorithmic data or a combination of the above. In this chapter we will explore the print source, different formats and also discusses the best format for print rendering.

Tagged Image File Format

TIFF is an industry standard designed for the handling of raster or bitmapped images. TIFF, tagged image file format is one of many file formats that are purely graphical, i.e., bitmap, raster zed, pixilated images. TIFF is commonly produced by scanners, fax machines, and other imaging hardware and software.

TIFF is designed for high resolution printing of images.TIFF files are rendered pixel-by-pixel and are typically large files, often forcing use of lower-resolution or less color than the original electronic or paper document (e.g. converting to FAX resolution bitonal).

Portable Document Format

The Portable Document Format (PDF) is a file format for representing documents in a manner independent of the application software, hardware, and operating system used to create them, as well as of the output device on which they are to be displayed or printed. Adobe PDF is a widely used document exchange format which is readable by anyone using Adobe's freely available Acrobat Reader. It is distantly related to Adobe's PostScript. Because PDF is rich in features to support photographic images, graphics, and color profiles.It has started to supplant Post Script, another good choice for users who require accurate color reproduction. Over the years, PDF has grown as a “precise file exchange” format in the fields of prepress and electronic learning.

The important benefit of PDF is, it allows the font to be embedded in the document, so type and layout both preview and print consistently wherever the file travels.PDF is used differently in document and imaging applications. Scanning and imaging applications are beginning to produce a raster-only, subset of PDF, commonly referred to as "image PDF".

JBIG2

JBIG2 is an International Telecommunication Union format. Image compression experts from a variety of companies joined to develop the JBIG2 standard for compression of black and white (bitonal) scanned documents.JBIG2 encoding shrinks the size of PDF images many-fold, making them 3 to 10 times smaller. The power behind JBIG2 technology is its ability to support both lossless and perceptually lossless black and white image compression. The format is flexible, and is designed for easy embedding in other image file formats, such as TIFF.

Portable Network Graphics

Portable Network Graphics (PNG), pronounced "ping," is an open, extensible file format for the lossless, portable, well-compressed storage of raster (bitmap) images. PNG was designed as a patent-free replacement for the older and simpler GIF. It is less complex than TIFF and a suitable replacement for many uses, especially small images. PNG supports indexed-color, grayscale, and true color images, plus an optional alpha channel for transparency. Also PNG supports up to 48-bit true color or 16-bit grayscale. Saving, restoring and re-saving an image will not degrade its quality,

unlike standard JPEG. Unlike TIFF, the PNG specification does not let implementers choose what features they'll support, so a PNG image saved in one application is readable in any other PNG-supporting application.

Encapsulated Post Script

Encapsulated PostScript format (EPS) is postscript with a few restrictions and comments in a specified format that provides more information about the postscript that follows. It was designed to make it easier for applications to include postscript generated elsewhere within their own pages. While PostScript is primarily a printing language, Encapsulated PostScript is a graphic file format derived from PostScript and is often used for transferring files between various graphics applications. EPS graphics are placed on single-page PostScript files inside documents destined for PostScript or PDF printing.EPS files contain PostScript code for describing the font and vector images. EPS files will print identically on all PostScript-compatible printers and will appear the same in all applications that can read the PostScript format. EPS files are more or less self-contained, predictable PostScript documents that describe an image or drawing and can be placed within another PostScript document. EPS files are generated from several sources and are structured to published code constraints used for storing font and vector image information.

Windows EMF

Enhanced Metafile (EMF) is the core display formats of the Windows operating environment and the native graphics formats for MS Office applications. EMF is a file extension for Enhanced Metafile, a spool file format used in printing by the Windows operating system. Enhanced metafiles provides true device independence. Since the enhanced metafile format is standardized, pictures that are stored in this format can be copied from one application to another. The EMF pictures are device independent and expected to maintain their shape and proportion on any output device.EMF is also consumed by the printer drivers as a graphics language. Windows Metafiles are intended to be portable between applications. LibreOffice, XnView and Cristal light WMF Converter Pro are the non-Windows applications which support EMF.

Application Emitted Print Data

PDF and XPS (XML Paper Specification) are the two dominant desktop publishing standards. In Linux and Mac world, PDF has become the de-facto standard for submitting the print job to the Common UNIX Printing System. Important upstream desktop applications such as GTK, GNOME, Qt, KDE, Firefox, and Libre Office have learnt to emit print jobs in PDF X format. Vector file mandates the PostScript language to draw its image. You must have a specific software application to create an EPS file, such as Adobe Photoshop, Adobe Illustrator and Quark Xpress. Microsoft promotes the XPS data from every application, and implements the XPS print stack. The XPS stack support mandates a new printer driver, and offers advanced printing features such as ticket based printing, high-fidelity, and improved color printing etc. XPS rasterisation service provides an inbox solution for the printer driver. Host based printing benefits from this feature.

Mobile Print Data

Existing print solution providers have their proprietary way of printing from the hand held devices. For example, Hewlett Packard assigns an e-mail ID to the printer, print jobs are sent from the corporate e-mail address to one of the company's networked printers from any e-mail capable device. Print job rendering is done using native applications in the print transformation server. Apple uses IPP (Internet Printing Protocol) for print management from their hand held devices such as iPhone and iPad. Any software printer on the work station (Mac and Windows) can be enabled for Apple's Air-

Print through the mDNSpublishing protocol. Air-Print requires a wireless(Wi-Fi)network. This is achieved by submitting the print job as a PDF. Mac/Linux CUPS framework, the application data is always filtered by the specialized filter programs. Microsoft Windows provides real time printing through GDI or XPS print path. In a nutshell, no specific format has been evolved for printing from hand held devices. Though there are many solutions being worked upon, there is no everlasting solution which eliminates the external print sub system and the application that provides the real time rendering to create print jobs compatible for the printing hardware.

PRINTING METHODOLOGIES

Host-Based Printing

The term Host-Based Printer is used to refer to a printer that has very little intelligence. It does not have a Page Description Language (PDL) interpreter, and built in fonts, or print straight ASCII text. Only a completely formatted image (rasterized data image) is sent to the printer. Therefore, it does not require a powerful processor or large amounts of RAM or ROM. To overcome this lack of printer intelligence, the work of generating each page shifts from the printer to the PC, specifically the host-based printer driver on the PC. The host-based software uses the computer's resources to process print commands and raster data, taking advantage of the computer's memory and processing power. This is where the actual page (raster) data is created and sent down to the printer. The data needs little to no processing on the printer (binary data in a stream or simple packet protocol).

PDL Based Printing

Host-based printing requires a software print engine in the host operating system, and unlike a PDL printer, cannot accept ASCII text directly from a computer. This means the host-based printer only works in Windows and Macintosh environments that are specifically supported with the print engine written for that environment. Users of unsupported Windows and Macintosh environments, as well as users of Linux, UNIX, or OS/2 should consider a PDL printer like the HP Color LaserJet 2500 or HP LaserJet 3700 series printer. Print Font substitution is added advantage in PDL based printing. In comparison, page description language (PDL)-based printers use the processor and memory resources of the printer. Citrix ICA, PC Anywhere or remote desktop connections are not supported on a host-based printer. Hewlett-Packard suggests using a non-host-based printer for this type of environment.

Pass through Printing

Applications themselves generate the postscript code and send that for printing. Printer driver is bypassed for the printing need. The main reason for an application to generate its own PostScript code is for platforms where the driver is not available, such as for UNIX®. In the past, some applications have had to generate their own PostScript code because the driver did not support PostScript Language, or because it was necessary for cross platform compatibility. One might need a printer driver that supports the PASSTHROUGH printing. PASSTHROUGH printing escapes native printer language codes to the data stream sent to the printer, so application can send data directly to the printer. However, few hardware vendors recommend that applications not perform functions that cause printer memory consumption, such as downloading or replacing the font at the hardware.

Device Independent Printing

The PostScript language is a programming language designed to convey the description of virtually any desired page to a printer. PostScript page descriptions are programs run by an interpreter. PostScript programs are usually generated by application programs running on other computers. This allows the output to be consistent when printed on more than one type of printer or print device. Specifically, the graphic objects will be consistent and in some cases of

higher quality than PCL. PostScript goes beyond the typical printer control language and is a complete programming language of its own. Many applications can transform a document into a PostScript program whose execution will result in the original document. The printer interpreter converts the Post Script program into the printed document. Also the Post Script program can also be viewed in the computer screen by the Post Script compatible application viewer. Since the document-program is the same regardless of its destination, it is called device-independent.

PDF Printing

PDF (Portable Document Format) is the most wonderful thing that could happen to the graphics and prepress industry. The file format is a compatible format in such a way that it can collect vast amount of information on the computer application. The compression algorithms can compress the size of the file, as a result, you can accumulate large amounts of information in the format.

These algorithms play a crucial role in enabling a user to incorporate the graphics or images in the file. In other words, a person can incorporate the use of graphics in the document and transfer the files across systems as these algorithms reduce the file size which makes the transfer of files easy and convenient.PDF is a computer application which is used for the purpose of archiving and exchanging different type of information. PDF is also independent of any application or software for running on the system. One of the format's key features is the ability to embed a font directly inside a PDF file; this, in turn, makes it possible to accurately render a document's fonts on computers where the font is not installed.

Raster Printing

RIP (raster image processor) is a component used in a printing system which produces a raster image which is nothing but a dot matrix data structure representing a generally rectangular grid of pixels. The raster image is then sent to a printing device for the paper output. The input may be a page description in a high-level page description language such as PostScript, Portable Document Format, XPS or another bitmap of higher or lower resolution than the output device. In the latter case, the RIP applies either smoothing or interpolation algorithms (upscale while retaining the quality) to the input raster to generate the output raster.

Raster image processing is the process and the means of turning vector digital information such as a PostScript file into a high-resolution bitmap image. A RIP can be implemented either as a software component (host based printing) of an operating system or as a firmware program executed on a microprocessor inside a printer, though for high-end typesetting, standalone hardware RIPs are sometimes used. Ghost Script and GhostPCL are examples of software RIPs. Every PostScript printer contains a RIP in its hardware/firmware. the following are the stages of rasterisation.

Raster vs. Vector Data Types

Vector data has algorithms used for extensively in the blue prints, drawings and plotter devices.

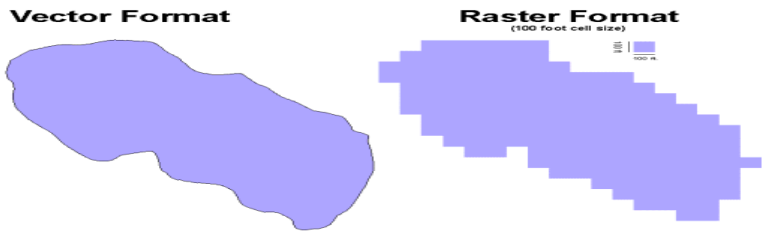

Figure 1: Data Types

Raster images can also be scaled on some occasions using the sophisticated algorithms.Raster images are primarily used with photos, which is why Photoshop is a raster editing program. Adobe Illustrator, on the other hand, is a vector drawing program that automatically creates your vector formulas as you draw. Here is an example: The following shapes represent the same water pond. The raster format is created by taking the vector version and converting it into 100 foot cell sizes. Raster format is easier for analysis and computation. Both formats are important because each has their own strengths and used in different scenarios.

Print to File

In the Print to file option, Windows saves the formatting and layout information of your file so a printer can reproduce the document without the program or computer that created it. This is merely a postscript file for the printer, not a raster image. There are many reasons to do this. When you do not have a printer in handy but want to archive a document and print it later or you want to send a document to someone who has the same printer, but does not have the program you used to create the document. However, Print to file has several limitations. It is a souvenir from the days of DOS, and it is primarily designed for parallel printers. It does not work easily on newer USB printers. Also, the computer used to print the saved file must have the same printer driver installed as the computer used to create the file. The Print to file option is a holdover from earlier days of computing. It does not work easily on more common USB printers, and you can accomplish many of the same goals by creating a PDF or an XPS document.

Transport Layer Printing

Most of the enterprise printers support a form of raw printing commonly referred to any of the following: raw sockets, port printing, standard TCP/IP, Port 9100, Port 2500, AppSocket, and IP direct. These names represent the same printing process, that is open a TCP socket-level connection on the specified port, and dump the code (PJL) or send a print ready file to the port and then close the connection. For the purposes of this document, Port 9100 refers to this type of printing. Port 9100 printing occurs at the transport level of the OSI model, bypassing the upper three layers and lowering the overhead for the print submission subsystem. Applications have to create print ready code and pass it into the TCP/IP stack for delivery. This is the most suitable printing model for the hand held device printing because the associated over heads are minimum compared to other printing ports.

Fax Printing

Some mobile printing applications workdifferently.They send the documents to fax machines rather than printers. The process works just like faxing over e-mail. The user selects an e-mail message that has a document attached and then enters a fax number. The Printing service translates the document into a format a fax machine can read. It sends the document to the fax number the user entered. Usually document formats for fax services are in PDF, TIFF or Word text forms. In another embodiment, the document is directly sent to the target printer. The section below has more details.

IPP (Internet Printing Protocol)

Still some legacy printing applications uses IPP 1.x, IPP2.x has superior specification which is way ahead of IPP 1.x.AndIPP Everywhere specifications are in the final stage, about to be frozen for the implementation. IPP everywhere has definition for MFD and MFP's, hence collaboration can be enhanced, like print from email, scanner to email and print etc. IPP 2.x has specification for fine grained printer management like shutting down or restarting the printer, printer control panel access, subscription based job management, promote job, re-process job, driverless specifications and lot more. Obviously it is a superior printing protocol for printer management, compared to SNMP Mibs and Print Job Language way

of printer management. And in recent times, printer hardware manufactures started supporting inherent IPP stack and the associated functionalities.

AIRBORNE PRINTING PROBLEMS

It is essential for people to work in a mobile, flexible way, by being able to access and print important documents whether they are at work, in different office, or in transit. Mobile printing problems are either unsolved or partially solved. Though we have many solutions available for hand held printing, none of them is able to fulfill the complete consumer needs.

Print Accuracy and Quality

Print ready content, images and PDF files are printed relatively easily, but other formats such as office documents and proprietary formats suffer. Examples of some of these issues include: Excel spreadsheets losing formatting, 3D charts and graphics getting flattened to 2D, fonts getting replaced, data missing. Content transformation is one key element to retain accuracy and quality of the original document. Documents printed from handheld devices don't always look like how they do when printed from desktop or laptop computers.

Absence of Print Graphics Engine

None of the handheld devices have the embedded print sub-system. No proper print sub system has evolved, even though the need for printing is increasing. Due to the absence of a print graphic engine, the data has to leave the application and undergo multiple translations before hitting the dead tree. Due to lack of entire application knowledge, the scope of the format conversion cannot be ruled out. This results in loss of print quality or a cluttered print.

If the content to be printed is not compatible to what printers accept, it is necessary to transform it into a print ready format. So there is a need for print ready conversion from the external world. In some cases, the content format may be compatible to what a printer can receive and hence the content is sent directly to the printer to print. These are typically higher-end expensive printers and restricted to a limited set of file formats.

Dynamic Print Device Discovery

The hand held device is expected to move across home, offices, cafes, and airport lounges etc. The expectation is to detect the printers on the current zone. The printers could be published on the LAN or individual PC's or through the Print servers (USB or Parallel or Network) or through Bluetooth or through LDAP print services schemas. Majority of the hand held devices do not have the ability to detect the printers dynamically (there are few but the discovery protocol sets are limited or proprietary). Location aware printer detection and job submit and management becomes necessary for the mobile handheld devices.

Printer Interaction Intelligence

Though AppSocket interprets print ready data, there are certain intelligences need to be built into the handheld device for the successful print job initiation and control. Instructing the printer about the type of data arrival and setting the interpreter mode for the PDL type is one of the challenges.

The absence of print a sub-system and printer driver means printer specific feature like selecting paper, color, number of pages in a page and many other options would not be possible from the hand held device. In case of Bluetooth interaction, printing profiles need to be negotiated. These intelligences are either completely unavailable or partially available. Hence it does not meet the printing requirements.

Source of Application Data Generation

Handheld devices do not have the knowledge about the origin of the data. Some data formats are generated by the device and most of the other data formats are received by the device. Typically, complex data formats are not generated by the hand held device. They are sent from the application which resides on the desktop computers. In a nutshell, the data on the handheld device can be classified into two categories, generated by the handheld device (iNote, PDF correction etc.) and generated elsewhere and arrived to the hand held device.

AIRBORNE PRINTING

Remote application rendering or deferred rendering is one of the easiest and most uncoiled method for handheld device printing. The data is transferred to the remote application's rendering service. The output from the application rendering, the geometrical data is fed in to the print sub system on which the printable data is generated and eventually sent to the printer on the network.

Application data is transferred to the print sub system through email or internet printing protocol. Data from the handheld device has to undergo format conversion before it reaches the printer. Since there is no real time rendering, there is a chance for the data loss. Since application and data are decoupled, real time printing is not possible. Though the above mentioned uses cases are very common in real world, majority of the organizations employs the remote rendering as a solution for the airborne printing from hand held device.

SOLUTION- FORMAT TRANSLATION AT SOURCE

A desirable and foolproof solution to handheld printing is to generate printer neutral languages files such as post script or PCL from the application on the hand held device. One of the obvious methods to achieve this is to generate the desktop published data in a portable format. Common portable formats are PDF and XPS, PDF is the most widely used format. XPS is being promoted by Microsoft and gaining attention in the desktop publishing world.

PDF/X Generation

Although data is digitized, the actual data representation will vary, it could be text, graphics, Lamda-calculus or algebraic equations. Applications can have choice of the data type but when it comes to desktop or print publishing, it has to be either vector or raster images. Though PDF/X was invented by Adobe, it is now an ISO standard for print publications and the community is actively working on improvements and new requirements.

Although mobile devices such as android do not have the ability to read all forms in the PDF documents, they have the inherent capability for PDF generation. iText® is a library that allow a user to create and manipulate PDF documents. All image formats, database and xml formats are supported in iText; this is available as a library in java and C sharp versions. But the fact is that not all applications are coupled with the above said libraries for PDF generation.

In fact the hand held device's document processing applications are scaled down versions of the desktop standard. At times it would be difficult or impossible to generate the desktop published data. To be specific, none of the java or C sharp PDF libraries can generate the PDF/X document which is meant for print publication.

PDL Generation

Though actually PDF files are usually better and in particular more compact, many applications desired to produce portable postscript files as a publishing idea. The other motivating factor for creating the postscript is simply to transform the postscript file into an established desktop published data. As explained above, although it is better and easier

to create the PDF file directly from windows applications creating postscript and then turning them into PDF files is still very popular.

Creating a post script file is easy from the Windows application. Select the print and the post script printer and using the printing option `print to file`. However, the resulting postscript file may not be completely portable – after all it was created for the specific printer. And you may not have a postscript printer at all. The solution is to install a virtual postscript printer, i.e. install printer drivers without actually having the corresponding printer after all you just print to file anyway. Another way of achieving this is to configure the output mode to optimize the print for output portability. But this configuration may not always yield the desired output depending on the printer driver. And PCL is not recommended since it is specific to printer hardware and portability is not accomplished.

AIRBORNE PRINTER DISCOVERY

Pervasive computing creates many challenges for service discovery, one of them being remote service discovery. Service discovery is an important feature for integrating service mobility into wireless networks. Mobile devices can use service discovery to automatically detect services in foreign networks, maintaining their set of subscribed, home network, services. As mobile printing becomes more widely used, the need to locate unfamiliar printers and printing devices is met by identifying such devices with location-enabled mobile phones, and portable digital assistants. Nearby printers and printing devices can be identified by such portable computing devices if the printer's location information is stored along with its make, model, IP address and other forms of identification. There are many service advertisement methods evolved along with the protocol namely, SLP, mDNS, IPP, SDP, SNMP, Blue Tooth etc. This section explores the remote printer discovery which does a service discovery in the network.

SNMP Discovery

Simple Network Management Protocol (SNMP) is an IP based protocol for managing devices such as router, servers, work stations, printers etc. SNMP itself does not define which information a managed system should offer. Rather, SNMP uses an extensible design, where the available information is defined by management information bases (MIBs). A Management Information Base (MIB) is a virtual database containing any number of characteristics of a managed device. The SNMP protocol is part of the industry standard Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. The printer should be SNMP enabled and should have implemented the Printer-MIB'sand should respond to query from the SNMP manager. SNMP can discover printers on more than one sub network by using multi-hop discovery.

Bluetooth Service Discovery (SDP)

Bluetooth technology offers a convenient way for mobile professionals and telecommuters to send documents to print without having to connect laptops, mobile phones or other devices physically to the printer. Service Discovery Protocol (SDP) is the part of Bluetooth that allows this to happen. SDP provides the means to find a device that will print for you or to browse through the range of other services Bluetooth devices in the area.SDP operates in Client-Server model; an SDP server is any Bluetooth device that offers a service or services to other Bluetooth devices (a service is a feature usable by another device). There is no question of remote discovery since the Bluetooth wireless range is about 30 feet.

DNS and mDNS Discovery

Using DNS interface for the network service discovery. DNS can associate various kinds of so called Resource Records with a given domain.DNS Remote Discovery via DNS-SD is designed to occur by making dynamic updates to a

DNS server. The use of unicast DNS allows for remote service discovery. If the DNS server is known for the given domain, (for example, google.com) the services are discovered by making standard DNS queries for and within the domain. Multicast DNS (mDNS) provides the ability to perform DNS-like operations on the local link in the absence of any conventional unicast DNS server. Multicast DNS is a protocol that uses similar APIs to the unicast DNS system but implemented differently. Apple's Airprint uses DNS and mDNS in the iOS platform.

Service Location Protocol

The Service Location Protocol (SLP) service discovery framework is composed of 3 components namely, User Agents (UA), Service Agents (SA), and Directory Agents (DA). UAs operate on a user's behalf to discover services that meet the user's specific requirements. SAs operate on the behalf of services, in order to promote them to other agents. DAs are entities that operate as a central repository to store or retrieve service information via service registrations and service requests. SLP can be configured with or without DA. Though alternative discovery mechanisms exist, SLP remains the most generic and versatile solution for service discovery on TCP/IP networks, SLP is both unicast and multicast protocol.

LDAP Discovery Services

In Light Weigh Directory Access Protocol (LDAP) discovery, the printer service discovery is done through an LDAP search request to the LDAP server. The LDAP request contains the LDAP server name, LDAP port, credentials, the context or group to search, and whether or not to recursively search sub-containers or sub-groups. The attributes of the printer objects can be read and the printing choices are made. Remote printer discovery is quite easy since the directory has replicated data base. Based on the user position (geography and network segment) the printers can be discovered through LDAP search. Not suited for discovery foreign networks.

Proposed Solutions

The goal of this study is to recommend the preferred printer discovery mechanism for the remote domain. Service discovery in a given foreign sub network is governed either by Unicast or Multicast.DNS unicast based remote domain's service discovery is appears to be a good solution since it eliminates the need to the IP multicast routing. Forwarding of multicast traffic from the local router to the sub network which is connected in the internet has inherent challenges though many multicast protocols are developed for the multicast routers. IP multicast is not, in general, used in the commercial Internet backbones.

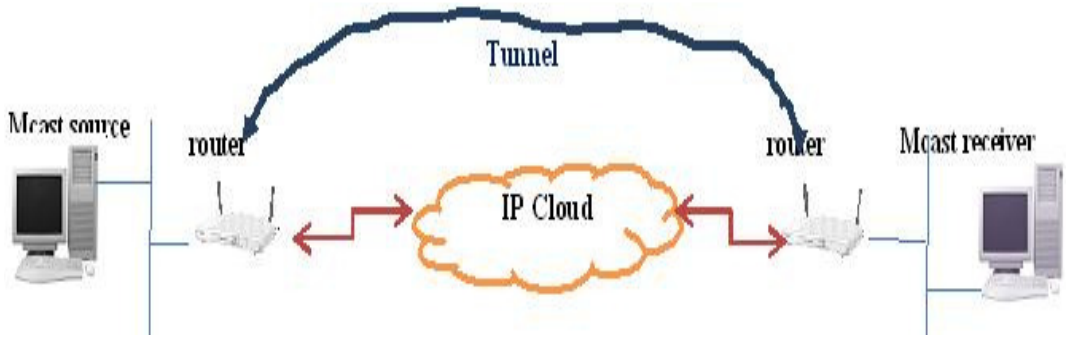

Figure 2: Remote Discovery

Considering the multicast limitations, IP tunneling is a good approach than enabling all the routers for IP multicasting. If SLP is employed for the service domain and Directory agent is deployed in the remote foreign domain, then the number of IP multicast packets will get reduced to a larger extent.

AIR BORNE PRINTING MODELS

Based on the use case, there are several airborne printing models possible. And in some scenarios combination of more than one model is mandated by the organization. The models are explained and the merits and de-merits are analyzed.

Self-Discovery and Self -Transformation – Direct Print

Associating the data with an appropriate printing profile and the data is transferred to printer through Bluetooth or through Wi-Fi network (Bluetooth preferred Wi-Fi need a special application for SDP Discovery/BPP/HTTP Print Job

Figure 3: Direct Print

the discovery and print job submission). This type of printing is straight forward as no external entity or processing required. SDP is the preferred printer discovery through Bluetooth baseband. Users on high mobility and low in print volumes prefer this type. No security release code since the user is near the printer and transmission security is addressed by the Bluetooth L2CAP layer.

External Transformation and Self-Discovery

Application data is converted into print ready data in the transformation server which has the printer drivers installed.

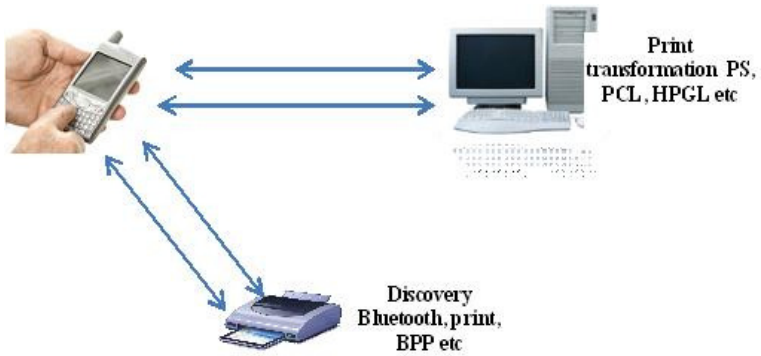

Figure 4: External Rendering and Direct Print

This approach essentially creates the PC loaded with the drivers and based on the target printer device, an appropriate driver is selected and the output is sent to the printer through the mobile device. The mobile or hand held device just associate the PDL data with the BPP profile or choose to dump the data in the printers App socket provided the port has the capability. The cons of this approach is the accuracy of selection of the right printer driver amongst thousands of options, and the large size of the PDL output that needs to be sent from the cloud to the device.

The advantages of this approach is the ability to support for new and emerging formats of content across different verticals – e.g., graphic arts, architecture and construction, etc., to add to the capability of the offering. With the processing power offloaded, the print outputs can be accurate and of the desired quality. Discovery is achieved by self. Self-discovery is preferred in public printing like airport and cafe lounges. Network discovery is most desirable for mobility users frequently change the work location within the organization for business need.

External Transformation, Self-Discovery and Reference Print

The PDL output data size is usually high compared to the application data. Print by reference is a requirement for devices that have limited local storage. Reference print supports the passing of a network reference to a printer, which then processes this reference in a manner specific to the implementation of the printer.

This interaction model expresses the notion of a print portal, where a print provider provides access to a local or Internet-based print service that supports printing for users who do not have native printing capability. The basic architecture of such an interaction is shown below.

Figure 5: External Rendering and Reference Print

For Print-By-Reference, two different reference formats are defined. Simple Reference: printer deals with a request that only contains a URL.XML Reference: The XML reference specification may be used to support the inclusion of Meta information that is passed along with the URL.

This meta-information may be used to improve the descriptive qualities of the reference (e.g., provide security hints, alternative access paths etc.).Reference List: A reference list is syntax defined to represent a list of XML references that can be sent as one transaction.

External Transformation and External Discovery

In this model, the job of the mobile user is very simple, upload the data and select the printer to print. Print for others or into your office printer from a remote location. The remote location (where print job is initiated) could be inside the corporate (different work place or geography) or from a public

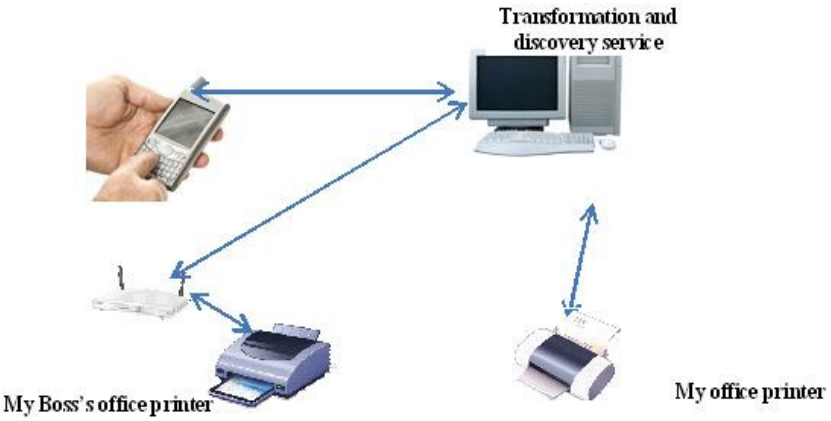

Figure 6: External Rendering and External Print

net work like home office, cafe or airport lounge. This is equivalent to a fax printing within the organization. User can submit the application data for print through e-mail or simple http upload to the transformation server where the print ready data is created. Then the transformation server requests for a target printer (physical) by negotiating with the user. There is no special negotiation protocol; it is based on the choice made by the user. In some cases the desired printer lists are pushed to the user from the server based on the learning or the predefined user configuration. The security release code (if the printer is shared) is transferred to the printer receiver through out of bound communication. **Merits**: This is very simple fax printing within an organization. User has a choice to choose the target printer. **Demerits**: Printing for others not for the mobile user, immediate print need is impossible, security and printing options are limited

AIRBORNE PRINT SOLUTION – ON SCREEN TO DEAD TREE

Often times the end device type is unknown and there is a compelling business need to send the data for print and the most demonstrated external print rendering or conversion system is not reachable. Here the business case is not revolved about the quality, scaling or the quantities rather the print service reach ability. In this section the rationale and the relation between the screen rendering and print rendering has been analyzed and an attempt has been made to capture the screen rendering for the printing purpose. No attempt has been made to capture the final screen rendered data rather to capture the meaningful intermediate data which can be consumed by other end devices. Here the end device type is not known, it could be anything, it could be a different display device, monitor, plotter or print device.

Rendering Pipeline

Real time print rendering pipeline consists of three stages; every stage has several sub stages, the rendering work flow is depicted below. Rendering is the process by which a computer creates images from models. These models, or objects, are constructed from geometric primitives such as points, lines, and polygons that are specified by their vertices. The final rendered image consists of pixels drawn on the screen, a pixel is the smallest visible element the display hardware can put on the screen.

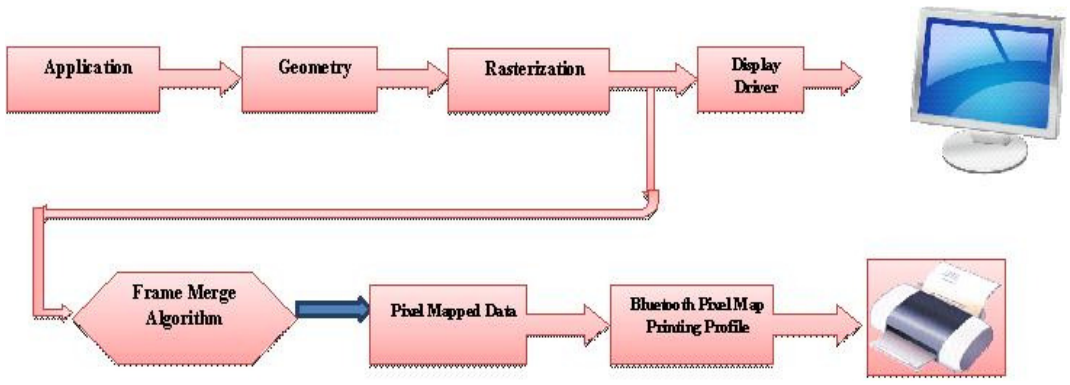

Figure 7: Screen Rendering Print

The application program developer defines the application stage. Application generates the geometry primitives and fed into the geometry stage. The rendering primitives are lines, vertices, polygons that might eventually end up in the screen or print device. Geometry is responsible for pre-polygon and pre-vertices operations. There are lot many operations like vertices shading, projection coordinates. On its' way to screen, the given model is transformed into different spaces and coordinate systems. Originally the model resides it's won space meaning that it have not transformed. Each model can be associated with the model transform to get positioned and oriented in the screen, duplicates are also removed. In Android application screen rendering is done by either 2D interface or OpenGL interface. In the rasterisationstage, all

primitives are raster zed that is converted into pixels. It is a process of converting vector-based graphics (such as lines, text and rectangles) into the pixel-based representations that can be directly displayed on a screen.

Video RAM

In general Video RAM is a random access memory (RAM) used to store image data for a computer display. It is a special type of dynamic RAM (DRAM). Video RAM is really a buffer between the computer processor and the display and is often called the frame buffer. When image is sent to the display, they are first read by the processor as data from some form of main (non-video) RAM and then written to video RAM. From video RAM (the frame buffer); the data is converted by a RAM digital-to-analog converter (RAMDAC) into analog signals that are sent to the display presentation mechanism such as a cathode ray tube, Liquid crystal display or thin film transistor display. Every application writes or transfers the raster data into the video memory, so all the time the video memory would have the raster data with embedded true color. This is nothing but a pure bitmap file which does not have any application specific data rather a printable image in raster specification.

Screen Resolution

For images viewed on hand held computing device or typical work station computer screens, scan resolution merely determines image size. The scanned image decides the size on the screen. More the scanning resolution, larger the on screen image size. It is quite natural to think of greater resolution as showing more detail and while that is generally true because it makes the image larger. But we are always greatly circumscribed by our output device capabilities and oftentimes we cannot take advantage of maximum image resolution. The images are huge, and our screen sizes are simply not large enough to accommodate.

Images are dimensioned in pixels and screens are also dimensioned in pixels, and video systems show pixels directly. And our video memory always has images and documents rendered in pixels. For example, image of 500 pixel image width is displayed at a different size on our different monitor screens. We do not necessarily see the same image. On the video screen, we may see the 500 pixel image anywhere between 4 and 8 inches wide, depending on our screen size and settings. That is a very wide size range for the same image on different screens, and it shows that there is no concept of exact size in inches on the screen.

So the hand held device computing device's video memory size is influenced or determined by the screen display resolution. If you have more screen resolution, to take advantage or exploit the high screen resolution, you need to have more video memory. From the printing industry statistics, 300Dpi image or document is the ideal candidate for printing. If the image density is more than 300Dpi, it might not make any difference in the press. Of course, screen also will not handle and will start horizontal and vertical buffering. In a nutshell, device with 300Dpi would produce a pixel layout that can produce optimal print. Most advanced hand held devices are coming up with screen resolution of more than 300Dpi, the optimal resolution suitable for optimal print. For example, iOS 4 comes with 326Dpi and Samsung note comes with 329Dpi.

Achromatic Rendering

In non-host based printing, bitmap and image files are transferred to printer as a PDL file. Image is in bitmap format, not the raster format for printer consumption. But in Bluetooth printing, image and text files are associated with a specific profile on which the handshake happens, file transfer and interpretation happens at the receiving end. Generating or capturing the vector data would not be possible from the video RAM. But there is a way to read the video RAM for the

raster data and of course we need a**sophisticated**frame map algorithm to extract the raster data and to eliminate the page overlaps and form the pixel data stream for the application data such as MS office document (This is a conceptual idea). The raster data or bitmap data from video RAM is dependent on application and not specific to the specific end device. This can be very well transmitted as a raw data to the host based printer provided the printer has the knowledge to understand and the raw data and choose not to send it to raster image processer. Non host base printer, we need to create a profile and associate the data accordingly or achromatic driver to generate the PDL data for the printer consumption. Of course this achromatic driver is colligated to pixel print and a mini driver not specific to any printer.

VIRTUAL, DRIVERLESS, LOCATION AWARE, SERVICE ORIENTED PRINTING MODEL

In this model, no remote rendering and printing is done in real time. By keeping the traditional printing attributes, the associated driver complexities are eliminated. And undifferentiated printing experience is achieved to a greater extent for all consumers.

Virtual Printer

A virtual printer is added to the file/print menu of the user's application. These printers convert the print job into post script data and submit them for further processing to the network print server. Printable data is created including all required resources like fonts, logos, images etc. with no user intervention. When we say virtual, the printing is abstracted and the work station user need not install the appropriate printer driver for all printer hardware in the world. This will help to get rid of printer drivers, version updates and associated problems. Work station user just need to install the virtual printer and done for the lift time or until the next version of PS specification arrives. Virtual driver outputs the generic post script format, take away the post script data on any platform software printer and pass the printing with the printer specific options as tickets. No need for the platform specific driver, the source of the print data is post script code. In addition to this the virtual printer must have the capability to generate the EPS code for print preview.

Location Aware Printing

And when we say location aware, the printing ecosystem is inherited based on the user’s current or specified location. Based on your current network operating zone and your printing preference, the user will get a list of recommended printers. The recommended print options and the printer lists will get automatically preserved as long as the network territories are unchanged. The user should able to select the most appropriate printer and would retain the print options. No static or predefined configuration. As and when the user migrates to a new location, the printing preferences also changes and the user wish to retain the printing options for future. Microsoft has somewhat similar options but that needs a static configuration.

Universal Printing Experience

This is not about the universal printing drivers. How the printing experience differs from desktop to desktop and desktop to mobile platforms like android and tablets etc.? For example in Mac and Linux world desktops Common UNIX Printing System (CUPS) provides the printing support through several filter programs and these platform applications are learnt or came to an agreement to emit the print jobs in the form of PDF data. By rendering print jobs using the Adobe PDF standard, print jobs can be saved or distributed to any client platform, creating a Universal Print Solution. But this may not be a true Universal Print Solution since the Application is decoupled for the PDL generation. Preserving the print data in an intermediate format always does not ensure the information correctness. Windows solutions use’s EMF formatting which is only available to Microsoft Windows clients, in fact this is sometimes not compatible across windows flavors and

needless to say there is no significance in non-windows desktops. All that we need is that common intermediate printer language which can be taken to any printer in the world through the superior print sub-system.

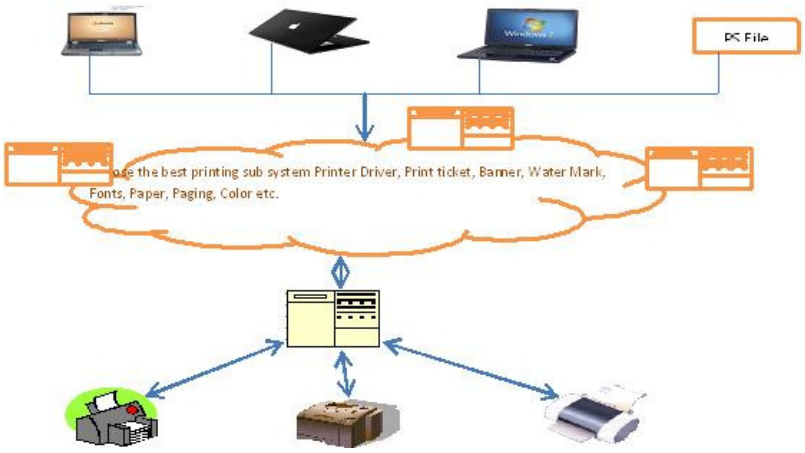

Figure 8: Universal Rendering & Driverless Print

When we talk about universal print experience, primarily it is about the various printing options at hand, like per page configuration, color specifications, recommended print paper size etc. These options are specific to printer drivers and not every desktop player has an advanced hosting environment to expose these advanced printing configurations to the application. By bringing up the printable data from the desktop application environment to the printer world and move it to the best printing sub system will greatly improve the printing experience. The best printing sub system need not be limited Windows, it could be print provider. Like, Xerox or HP can have their own pre-press print subsystem which consumes the page description language along with the advanced printing configurations. Also the print sub system can choose to emit the EPS data back to application for the final confirmation for the print specifications.

In another embodiment, the printable data can be fed to the print driver for the pass through printing in which the PDL generation is avoided and the printing configuration and preferences are encoded into the printable data. This can be achieved anytime by choosing the best print system and bypass the PDL conversion.

How to provide the universal printing experience across different desktop and devices? How about the print quality? Is it possible to provide the universal print quality experience across different print sub systems? This may not be possible since the application data has to geometry conversion and subsequent algorithmic and PDL conversion will vary in individual print sub-systems. This cannot be influenced externally (all the time) and it is the user's choice to select the desktop operating system which has the best geometry conversion.

Driverless Printing

Driverless means eliminating printer specific drivers at the base station where the application resides or the desktop published data resides from where the printing is initiated. Driverless is not limited to hand held devices but all types of work stations. The need for the printer driver comes into picture when we like to convert the application data into PDL data with the printer specific control codes. The moment we have a virtual driver or application is able to emit the printable data, now the job is simplified. Yes, we do not need a printer driver where the print job is initiated. This will eliminate the printer driver dependency on the desktop or the hand held device. In another configuration, some printing vendors already came up with the inbuilt SDK which adds value to the printable document before the RIP (Raster Image Processor) consumption. Like setting the page size, bordering, page specific print options, color option etc.

Usually every driver vendor has their own way of exposing the printer capabilities to the end user. There is no standard way of doing this. Print tickets are generated for every print job and merged with the default or forced ticket from the administrator. In another configuration, independent Print Job Ticket is provided through PJL or transport layer. Tickets can be submitted for every job or the printer can always configured with a default generic print ticket.

The print job ticket is usually prepended to the file, and can be recognized by the special characters at the beginning. Though the print takes the form of PostScript comments, it is used for submission of PDL data as well. The print job submitter should have the knowledge about the printer capabilities unless the default or generic job ticket would be associated by the printer for all the print jobs. This is where the print service provider provides the service to the mobility user.

Remote, Dynamic Discovery

Printing on the move, dynamic printing mandates the remote and dynamic discovery of print devices. In either case the discovery is initiated from the remote machine or a print server. You do not have an installed discovery client on the device. It is not about delivering the discovery application

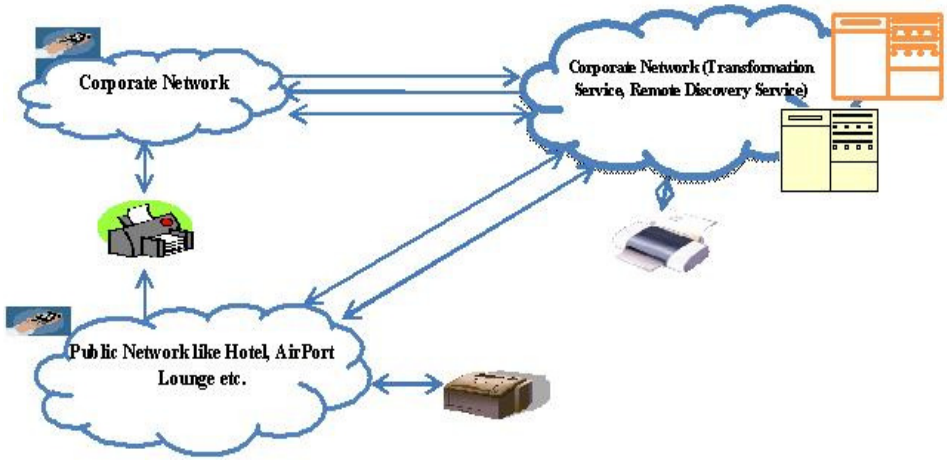

Figure 9: Remote Printer Discovery and Print

through browser. Here, the real need is that, the mobile user has no knowledge about the printers around and would like to print to the available choices nearby and my device has limited processing and storage capability. The access point to the network would be derived or obtained by the mobile user and the domain is identified. And the user has a choice to specify the remote domain on which the printing service is needed. Once the device access point is obtained, the print service provider will initiate a remote discovery of printer on the domain. And the associated printer attributes are gathered as part of the printer discovery process. The response to the user is sent as a set of printers and the service capabilities along with the geography. The geography could be a map itself or the chargeable print office. The type of discovery advertisement should be preloaded or sent dynamically for the remote discovery service to initiate the discovery.

Service Oriented Mobile Printing Model

Printing can also be served as service from the cloud. Like application rendering service, remote discovery service, print rendering service etc. Though these terms are new to the print solution but makes great sense when the print sub system is not around or inaccessible.

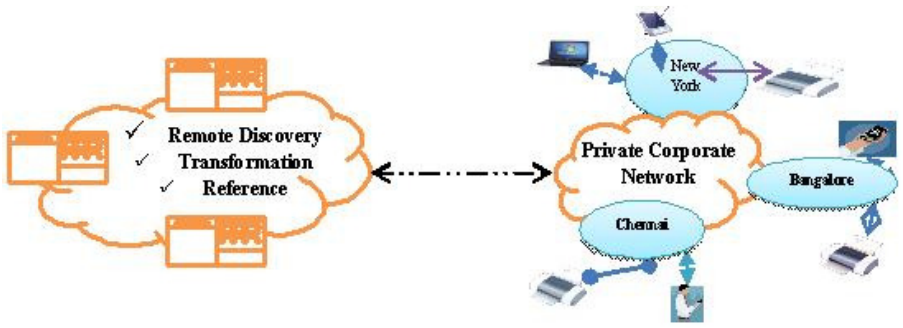

Figure 10: Printing as a Service

And print as a service will let people choose the superior printing sub system than going with the platform's default print sub system. Source device need not have the print sub system, printing services are hosted by the typical print servers or multi-tenant enabled printer servers for the cloud environment. In a nutshell, print as a Service is not only addresses the hand held device print computation rather it facilitates an alternate means to select the desired print sub system irrespective of user's platform choice. Like if Windows GDI or XPS print path provides more printing features the Mac OS X user may wish to print using the windows print sub system. Similarly CUPS (Common UNIX Printing System) from Mac OSX provides better print qualities and features, the windows users may wish to print using the Mac OSX print sub system etc. The following are the example for print as service, user has the application data, looking for a printable data (PDL+ticket) with printer discovery and user has the Post Script data and looking for a print ticket with printer discovery.

REFERENCES

1. http://www.google.com/patents?id=B2jJAAAAEBAJ&printsec=abstract&zoom=4#v=onepage&q&f=false (US 7,609,402 B2)
2. http://www.undocprint.org
3. http://www.linuxfoundation.org
4. http://www.faqs.org/patents/app/20120002980#b#ixzz1th4yZEeM
5. http://www.bluetooth.com
6. Q-CAD: QoS and Context Aware Discovery Framework for Mobile Systems - Licia Capra, StefanosZachariadis and Cecilia Mascolo, Dept. of Computer Science, University College London
7. A Distributed Architecture for Remote Service Discovery in Pervasive Computing – by FarzadSalehi A thesis submitted to the Department of Computer Science in conformity with the requirements for the degree of Master of Science Bishop's University, Canada, December 2011

AUTHOR'S DETAILS

Master of Engineering in Applied Electronics Engineering from PSG Technology, Coimbatore, India. Over a decade experience in IT industry, includes disaster recovery, data protection, desktop storage management solutions and now into enterprise print solutions. Specialized in storage management, file sharing, data synchronization, storage stack performance analysis and software mobile print solutions. Owning many patents in US patent office, and published many research papers.http://www.linkedin.com/pub/arul-selvan/2/487/a72