

ANALYTICAL STUDY ON PNG IMAGE TRANSFER ON
BLUETOOTH PICONET WITH RESPECT TO NODES
AND TIME TAKEN

Mr. Sujit Jhare*

Mr. Divakar Singh**

Dr. Shailendra Singh***

Abstract:

The PNG image is the another one basic image format in digital world that is a enhance version of GIF, the limitation of GIF is, it is support only 256 colors and the PNG support true color, here we are going to present an analytical study to sharing a PNG image on multi hop Bluetooth network over piconet, and evaluate the result according to time taken by image transfer on network, this strategy provide us the result for PNG image, is how much efficient on multiple node under piconet circumstance, and the result showing, how much piconet nodes are suitable for the PNG sharing with less times when the network load increases or decreases.

* Developer, Mahindra Satyam.

** Asst. Prof. (CSE), BUIT, Barkatullah University, Bhopal.

*** Proff, Dept. of Com. Eng. App., N. I. T. T. R. Bhopal.

Introduction:

Bluetooth is a recently proposed standard for short range, low power wireless communication [1]. Now days several devices including personal computer, possibly a laptop, keyboard, mouse, joystick, printer, scanner, etc. that is equipped with a Bluetooth Connectivity card. There are no need to connections between these Bluetooth devices, Bluetooth enables seamless communication between all them, as the data carrier in between two mobile devices (text, audio, video, and file) Bluetooth Work on 2.4 GHz in between Less-than 10 Mt. Range.

[2]The Piconet formed via connecting eight different Bluetooth devices. And, the Slaves can communicate with each other in a Piconet via the Master. There are two types of connection link to transmit data between Bluetooth devices. For file transmission, it uses an ACL (Asynchronous Connection-Less) link when it requires reliable data transmission service, and it uses SCO (Synchronous Connection-Oriented) link when it requires real-time service, such as voice, multimedia data stream.

The Bluetooth piconet is a form of broadcasting a file on different node, when a master node open a connection stream and all slave node that is connected to the master node is establish the connection, in the inquiry search mechanism all slave devices search a master node, and after searching slave node going to connect one by one to the master node, the Star topology is working behind on this Network, up to eight node are connected successfully with the master node and remaining node are keep in pool area in master node, once any one slave node is disconnected from master node then the remaining pooled one slave node connected to the master node and this is again performing the piconet. As the piconet is establish for all time in broadcasting mechanism [3].

The Master is the Bluetooth device that sets the frequency hopping sequence. The Slave synchronizes to the Masters in time and frequency by following the Master's frequency hopping sequence. Every Bluetooth device has a unique

Bluetooth device address and a 28-bit Bluetooth clock, the baseband part of the Bluetooth System uses a special algorithm, which calculates the frequency hop sequence from the masters' clock and device address. In addition to controlling the frequency hop sequence, the Master controls when Slaves are to transmit using Time Division Multiplexing (TDM) [4]

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because in this case JPG files are usually smaller than PNG files. The Adam7-interlacing allows an early preview, even when only a small percentage of the image data has been transmitted.

PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel.

PNG is designed to work well in online viewing applications like web browsers so it is fully streamable with a progressive display option. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors. Also, PNG can store gamma and chromaticity data for improved color matching on heterogeneous platforms.

Some programs do not handle PNG gamma correctly, which can cause the images to be saved or displayed darker than they should be.[12]

Animated formats derived from PNG are MNG and APNG. The latter is supported by Mozilla Firefox and Opera and is backwards compatible with PNG. [5].

Once an image is taken, during its lifetime many actors will operate on that image and its metadata. Because of this, some issues may arise with rationalizing the region metadata stored within the image. When non-compliant actors have processed the image, the integrity of the region metadata may be invalidated. It is important that compliant actors validate region metadata as well as update it such that the region metadata becomes valid.

If an actor does not properly update the region metadata, issues will arise when certain operations are performed on an image.

Based on a 2006 proposal by Microsoft, the Metadata Working Group (MWG) organization was created in 2007 by 5 founding members: Apple, Adobe, Canon, Microsoft and Nokia. Sony joined this initiative in 2008.

The goals are:

- Preservation and seamless interoperability of digital image metadata
- Interoperability and availability of metadata to all applications, devices, and services
- Enabling emerging metadata-based workflows

The organization is based on a formal legal framework and royalty free intellectual property policy that allows member companies and other industry leaders to collaborate on a solution to the above problems. The efforts of the MWG are organized into initiatives. The continuation of MWG's first initiative (covered in this document) addresses issues of digital imaging metadata for typical consumers. Future initiatives might deal with metadata for professional photography, audio and video metadata, etc. [6]

Implementation Strategy:

The Bluetooth piconet is a collection of Master and Slave devices, and our object is to create an analytical study on PNG image on piconet according to distribution time frame, so for the implementation of this idea, we Create a PNG image using Photoshop, because of simulation environment having memory constraint so we need to create a PNG with exact size that is suitable according to simulation environment, one great advantage with PNG image is that the simulation environment takes image data from PNG file up to their supporting memory and remaining file data is automatically discard by the environment, but for the constant result we chose the appropriate size of file. after creating a PNG image, we put the image on Master device, and place the path in JAD file, and start the simulation, on the master device the PNG image file is being open to send to all slave device, for this first we need to select the image file and after selection, keep this image file on writing stream, i. e. on distribution stream, now all slave device search for the object that is open for distribution on master, and found the name of

this image file, after searching complete, when all server demands the image file form the master, from here we observe the exact time of distribution of the image from master to all connected slave node, and this approach provides us the analytical output for the PNG image transfer on piconet under various number of nodes, when the PNG image move on two connected slave device then the distribution is much faster than three connected node, but as the node increase and reach up to seven connected node then the difference is much less or equal time taken.

Development Environment:

Net bean [7] is a free, open-source Integrated Development Environment for software developers. All the tools needed to create professional desktop, enterprise, web, and mobile applications with the Java platform, as well as with C/C++, PHP, JavaScript and Groovy.

NetBeans IDE 7.0 introduces language support for development to the proposed Java 1.6.0_02; Java HotSpot(TM) specification with the JDK developer preview, support for GlassFish 3.1, Oracle WebLogic, Oracle Database, Maven 3, HTML 5 and more.

Simulator Environment:

for the simulation process of the entire application, Sun Java™ Wireless Toolkit for CLDC Version 2.5.2_01 for mobile device, sun java simulator is a third party tool to test and simulate the mobile application developed on Net Bean, it is internally associate with the Net Beans, and work like a inbuilt tool of Net Bean.

The Sun Java Wireless Toolkit for CLDC [8], formerly known as the J2ME™ Wireless Toolkit, is a set of tools for creating MIDP applications. It consists of three main components:

- The toolkit user interface automates many of the tasks involved in creating MIDP applications.
- The *emulator* is a simulated mobile phone. It is useful for testing MIDP applications.

- A collection of *utilities* provides other useful functionality, including a text messaging console and cryptographic utilities.

Scope of the work:

In the related work the PNG images is taken as our primary data and Bluetooth Piconet as network environment, and Bluetooth enable mobile devices as master & slave node, Net Bean 7.0 as IDE for development and simulation of entire process.

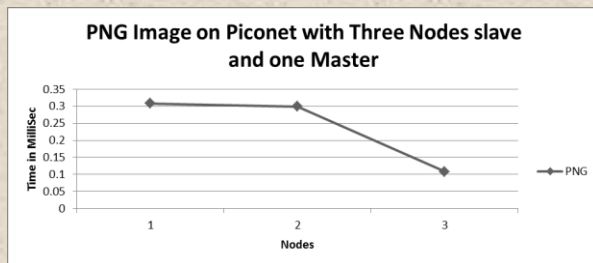
The entire application will be developing using JSR-82 and Midlet suite [9], at the very first we create a J2ME application for piconet establishment [10]. After creating the application we keep PNG image in resource folder and update the resource file for JAD with path of testing image, after running the application when the first mobile screen appears from this screen we need to select the Master or Slave Device. After selecting a device as master device, then from that device we can send the images to the network or the other slave devices. Now after selecting master device, on the simulator we open again a new mobile device, and select this device as slave device, after selecting the slave device, we need to establish the Bluetooth network connection in between Master and Slave, as the master connect with slave, all slave device according to Bluetooth address, stored in master device remote list, once our piconet is completed, then we choose the images list form the master device, in the current stage, all the images will be shown on master device.

Output result:

for getting the more closer result for PNG image and piconet operation we test the operation under various circumstance and find the result for all possible condition, in this testing condition we also merge the noise at simulation level and network barrier on the simulation level so we can found the most appropriate result same as the real environment, where noise and other barrier break down the network.

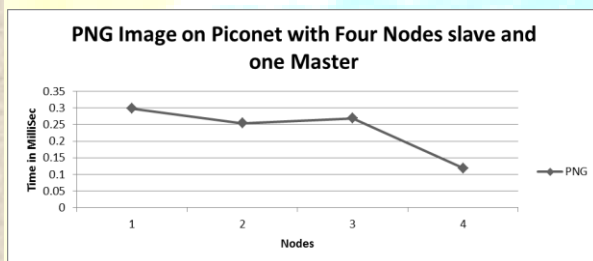
We calculate the output on basis of these given circumstances, that test reports and desired output is showing with graphs.

Ist Pattern Analysis: In the first pattern we chose the three slave node with one master, and transmit the PNG image file on those nodes, and we accumulate the given output.



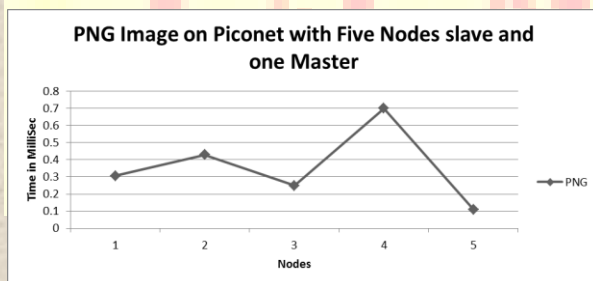
According to the above chart it is showing when we move image on three nodes, the time taken to complete the operation from 0.1 seconds to 0.3 seconds.

IInd Pattern Analysis: In the second pattern we chose the four slaves with one master, and transmit the png image file on those nodes, and we accumulate the given output.



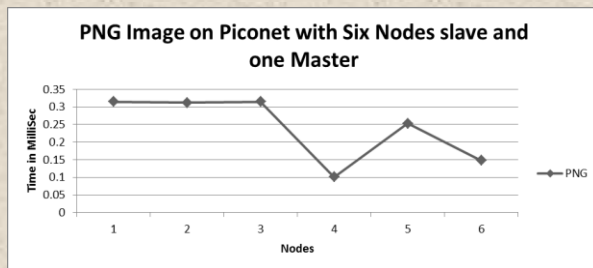
According to the above graph it is showing when we simulate with four nodes the time taken from 0.11 seconds to 0.3 seconds.

IIIrd Pattern Analysis: In the third pattern we chose the five slaves with one master, and transmit the png image file on those nodes, and we accumulate the given output



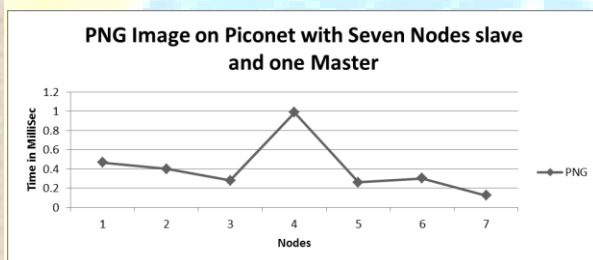
According to the above graph it is showing when we simulate with five nodes the time taken from 0.1 seconds to 0.7 seconds.

IVth Pattern Analysis: In the fourth pattern we chose the six slaves with one master, and transmit the png image file on those nodes, and we accumulate the given output



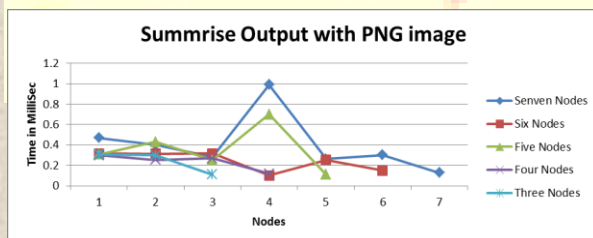
According to the above graph it is showing when we simulate with six nodes the time taken from 0.1 seconds to 0.3 seconds.

Vth Pattern Analysis: in the fifth pattern we created the complete piconet having seven slave nodes and one master node, and then move our png image file on this network, and the time taken as in result is showing in this graph.



According to the above graph it is showing when we simulate with seven nodes the time taken from 0.01 seconds to 0.5 seconds. With pick point 1 Sec.

Final Output: when we merge all the output values in one graph to cover the complete analytical study, we found very new and results according to our test criteria, the combine graph is given as:



When we calculated the output with all seven nodes and master device having network noise and merged all the calculated result into one graph, then we get the clear picture of output with all the

compression, in this graph it is clearly visible if we test the criteria with less nodes then the file transfer takes less time overall, and up to five nodes the results are much more constant, but as we add sixth and seventh node the results varies mostly, and with the seven nodes the total time is equal to each other but the peak point is going near about to 1 Seconds. This is because of noise and actual network traffic that we apply by the simulation.

The actual values for all the calculation is given in below table.

0.468	0.401	0.28	0.99	0.261	0.301	0.126
0.315	0.312	0.315	0.101	0.253	0.148	
0.306	0.429	0.25	0.7	0.11		
0.298	0.254	0.269	0.119			
0.308	0.299	0.108				

According to calculated value, for all nodes the start and end time is variant, while for the middle nodes value having less variance, this is due to network and noise and the file format that we used.

But according to the overall strategy the png image have variant on network if we calculate the total average time for all pattern then we got the average patterns time is:

For Seven Nodes	0.403857
For Six Nodes	0.240667
For Five Nodes	0.359
For Four Nodes	0.235
For Three Nodes	0.238333

Based on the overall average output for all the nodes we found that, when we use full piconet using seven nodes and one master then in the average file takes more time to transfer on network. But if we consider overall output then we noticed the point the png image could be a good option to store image data and move on network.

Conclusion:

When we apply these test on the piconet with png image, and found that the image takes nearly equal time to move on the network, and all the time is nearby to each other, so we can say that the gif image could be a better option to move on network, the concept is, when we creates a png image so it is compressed and suitable image for internet also, so if we move some pictorial information with the help of images, so png image is an another good option. So based on the above outputs and calculations we found the png images is good for piconet network.

References:

- B. Miller and C. Bisdikian: *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*. Prentice-Hall, 2000
- ByoungKug Kim, SungHwa Hong, JeongSik In, JaeWan Kim and DooSeop Eom : A Method to Support High Data Transmission Rate in Ad-Hoc Networks based on Bluetooth, Seoul, 136-701, Korea
- United States Patent Patent No: US 7532594 B2, May 12, 2009
- <http://netbeans.org/features/index.html>, introduction of net bean
- http://en.wikipedia.org/wiki/Image_file_formats#GIF
- Bluetooth Piconet Applications J. MANDER and D. PICOPOULOS
<http://www.ee.ucl.ac.uk/~afernand/Example1.pdf>
- Release notes of sun java toolkit for CLDC Version 2.5.2_01.
- http://www.metadataworkinggroup.org/pdf/mwg_guidance.pdf GUIDELINES FOR HANDLING IMAGE METADATA Version 2.0, November 2010
- Developing Applications with the Java APIs for Bluetooth™ (JSR-82), published by Sony-Ericson in January 2004.
- J2ME Bluetooth Programming, Master's Thesis, André N. Klingsheim, Department of Informatics, University of Bergen, 30th June 2004
- Knowl Inf Syst (2008) 14:1–37, DOI 10.1007/s10115-007-0114-2, Received: 9 July 2007 / Revised: 28 September 2007 / Accepted: 8 October 2007, Published online: 4 December 2007, © Springer-Verlag London Limited 2007
- <http://hsivonen.iki.fi/png-gamma/>