Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 3, March 2013, pg.26 – 34

RESEARCH ARTICLE

Effect of Twisted Pair Cable Arrangement on Manual Termination with RJ 45 and the Quality of Communication in Auto-Sensing Port Devices

Samuel King Opoku[1], Malik Adams[2], Edward Akaadom[3]
[1]Computer Science Department, Kumasi Polytechnic, Ghana
samuel.k.opoku@gmail.com
[2]Computer Science Department, Kumasi Polytechnic, Ghana
johnmorocco@gmail.com
[3]Computer Science Department, Kumasi Polytechnic, Ghana
kingeddy2012@gmail.com

Abstract— Current Network devices have been implemented with sensors that automatically detect and accordingly manage the various functional cables of a twisted pair cable. This paper research into the effect of randomly arranging twisted pair cables on communication and the complications associated with manual termination with RJ 45 connectors. Different sets of unshielded twisted pair cables were randomly terminated. These cables were used to set up communication channel between homogeneous and heterogeneous network devices. With the help of communication detection mechanism and a Java desktop application implemented with Java SE, the quality of communication was analyzed. The manual termination processes were also closely monitored and analyzed.

Key Terms: - Cable Arrangement, Colour Coding Scheme, Ethernet, LAN, MDI, MDIX, RJ 45, Twisted Pair.

I. INTRODUCTION

Network administrators and designers over the years have stuck to two different types of cable specifications, namely, crossover and straight through ([1]-[3]). Twisted Pair ports are generally connected so that the transmit pair on one end is connected to the receive pair on the other end, and vice versa [4]. Hubs and switches are deliberately wired opposite the way end stations are wired ([4], [5]). When a hub or switch is connected to an end station, a straight through Ethernet cable is used and the pairs will match up properly ([1], [4]). When two hubs or switches are connected to each other or two end stations are connected to each other, a crossover cable is used to make sure that the correct pairs are connected ([2], [4]). The standard wiring for end stations are known as Media Dependent Interface (MDI) and the standard wiring for hubs and switches is known as Media Dependent Interface with Crossover (MDIX) [4].

Current network devices are implemented with auto-sensing ports that automatically detect and accordingly manage the various functional cables ([4], [6], [7]). This mechanism is achieved through the implementation of auto-MDI/MDIX [7]. This research analyses the effect of random cable arrangement on the quality of communication in an auto-sensing port device. It also determines whether time variation exist between different cable arrangements during communication between homogeneous and heterogeneous devices. The level of difficulty in terms of manually terminating network cables forms integral part of this research. Communications between devices were tested using pinging. There were five major devices used: two laptops (PC's), two Sky Routers and a D-Link Switch. A Java desktop application designed with MySQL Database Management System was used as a testing system. A request is sent by one of the PC's (called PC 1) to query a database stored on the other PC (called PC 2). The time spent in connecting and fetching data through the cables were recorded and analysed.

Auto-MDIX automatically detects the required cable connection type and configures the connection appropriately, removing the need for crossover cables to interconnect switches or connecting PCs peer-to-peer ([4], [5]). Similarly, auto-MDI automatically detects and configures the connection between a hub or switch and an end station. For auto-MDI/MDIX to operate correctly, the data rate on the interface and duplex setting must be set to "Auto" [4]. In [6], a pseudo-random number generator decides whether or not a network port will attach its transmitter, or its receiver to each of the twisted pairs used to auto-negotiate the link. Auto-MDIX was promoted within 1000BASE-T standard in [7]. Algorithms were developed to force auto-MDIX mode which allowed links to be automatically established even if the port does not auto-negotiate in [8]. Newer routers, hubs and switches use auto-MDIX to automatically switch to the proper configuration once a cable is connected. The other four wires are used but are not crossed since auto-MDIX is mandatory at the higher data rates [4]. Thus 1000BASE-T transmits and receives on all pairs simultaneously without any dedicated send/transmit pairs ([4], [5]).

Twisted pair is a type of wiring where two conductors are twisted together in order to cancel out electromagnetic interference from external source ([9]-[11]). There are two types of twisted pair, namely, unshielded twisted pair and shielded twisted pair ([10]-[13]). Unshielded twisted pair is a cable type with one or more pairs of twisted insulated copper conductors contained in a single sheath. UTP cable is manufactured in various categories. Each category indicates the performance capability of that cable [9]. Category 3 (preferred name is CAT 3) uses 10BASE-T standards [11], CAT 5 uses up to 100BASE-T standards with the enhanced CAT 5, called CAT 5e supporting 1000BASE-T standards ([9], [12]). CAT 6 and its enhanced CAT 6, called CAT 6e, can handle up to 10GBASE-T standards [9]. Shielded twisted pair cable has extra cover to protect the transmission line from electromagnetic interference [13]. It comes in three forms, mainly, screened-unshielded twisted pair (S/UTP), screened-shielded twisted pair (STP) and screen-shielded twisted pair (S/STP) ([10], [13]). S/UTP has an overall shield and is good for protecting the signal from external interference. It does not protect the signal from any internal crosstalk. STP has four individual shields, one for each pair inside the cable. It is suitable for protecting the signal from internal crosstalk, but does not have an overall shield to protect the signal from external interference. S/STP has four individual shields, one for each pair and an overall shield. It is the best cable to protect signals from any type of interference ([9]-[13]).

II. DESIGNING NETWORK CABLES AND TESTING SYSTEM

Ten sets of cables were designed for the research work. The cables were grouped into four types, namely, Type A, Type B, Type C and Type D. Unshielded Twisted Pair category 5 cables were used throughout the experiment. The various wires in CAT 5 twisted pair cable are coloured: Blue (Bl), White-Blue (WBl), Orange (O), White-Orange (WO), Brown (Br), White-Brown (WBr), Green (Gr) and White-Green (WGr)

A. Type A and Type B Cables

Two sets of cables were designed for both Type A and Type B. Type A cable follows the usual straight through specification and they were labelled 1 and 2. The figure below shows how the ends of straight through cables are made.

Fig. 1 Straight Through Cable Design

Type B cables follow the usual crossover specification and they were labelled 3 and 4. The figure below shows how the ends of crossover cables are formed.

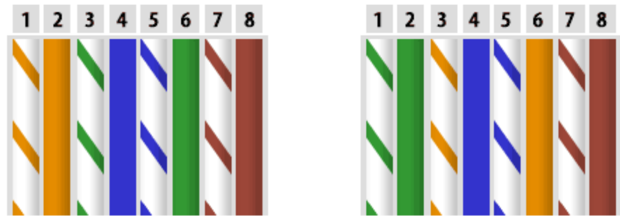

Fig. 2 Crossover Cable Design

B. Type C and Type D cables

Three sets of cables were designed for both Type C and Type D. Colour coding schemes were randomly picked. Pieces of papers with cable colours were placed into a bowl and they were randomly picked one at a time to determine the colour coding scheme. Type C cables had the same colour coding scheme for both ends whereas Type D cables had different colour coding scheme for both ends. The following algorithm was used to design both Type C and Type D cables

```
00 Begin
01      Get 8 pieces of papers and label them WBl, WO, WBr, WGr, Bl, Br, O, Gr
02      Place all the papers into a bowl
03      Do
04              Pick a paper from the bowl
05              Write the colour which is the next cable to use during termination
06      Loop While (Bowl is not empty)
07      If (coding scheme obtained is not equal to previous coding scheme) then
08              Label the colour coding scheme
09      Else
10              Discard the current coding scheme
11      End If
12      If (there are three sets of cables) then
13              If (they are designed for Type C) then
14                      Other ends of the cables follow the same colour coding scheme
15              Else If (they are designed for Type D) then
16                      Goto 02 //determine coding scheme for other ends of the cables
17              End If
18      Else
19              Goto 02 //continue with coding scheme to obtain three cables
20      End If
21 End
```

From the above algorithm, six terminated cables Labelled 5 to 10 were obtained. The figure below shows the various arrangements obtained for Labels 5, 6 and 7 with A and B indicating the ends of the cables

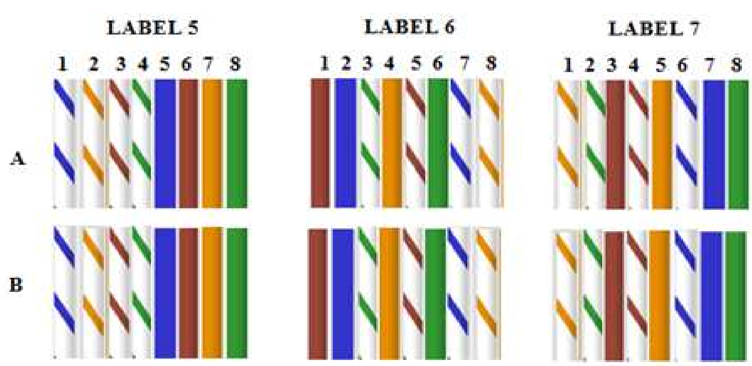

Fig. 3 Type C cables

Labels 8, 9 and 10 cables obtained using the above algorithm with A and B indicating the ends of the cables is shown in the figure below

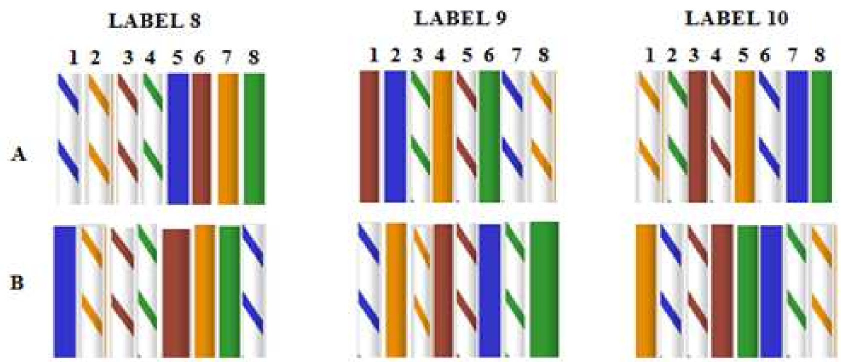

Fig. 4 Type D cables

Unfortunately, after scores of hard work, none of the Type D cables was successfully terminated with a cable terminating tester called PN-S Multi-Purpose Lines Communicating Scanning and Testing Device. Thus Type D cables were rejected for further analysis. This problem was associated with inability of the cables to reach the end of the RJ 45 connector during termination of the "B" ends of the cable. These cables usual drew back when the hand which clenched them was released when the cables were in RJ 45 connector.

C. Testing System Design

Two Laptops (PC's) were set up to communicate with each other. The PC's first pinged each other to determine whether there existed communication between both PC's. A simple Java desktop application was designed on one of the PC's (called PC1) to query a database created with MySQL Database Management System on the other PC (called PC2). The results were fetched back to PC1 through the communication medium and they were displayed on the console of PC1. The simple Java desktop application was implemented using the following algorithm:

```
Begin
    conStartTime = System.currentTimeMillis();
    connectToDatabase(server_IPAddress); //a method to connect PC 1 to database on PC 2
    conEndTime = System.currentTimeMillis();
    System.out.println("Connection time is: " + (conEndTime - conStartTime));
    fetchStartTime = System.currentTimeMillis();
    fetchData(queryStatement); //a method for fetching data from the database
    fetchEndTime = System.currentTimeMillis();
    dataFetchedTime = fetchEndTime – fetchStartTime;
    System.out.println("Data Retrieval Time: " + dataFetchedTime);
End
```

The data fetched were StudentID, First name, Last name, Address and Age. The same data was fetched throughout the experiment. The PC's were connected directly with cables and indirectly via a Switch and Routers.

D. Algorithm for Selecting Type C Cables

In order to select one or two cables randomly from Type C, the calculator's Ran() function was used with the following algorithm

```
Begin
    //Ran() is a function for generating random numbers between 0 and 1 inclusive
    code = int (Ran() * 3) //converts floating point number into integer by truncating the number
    If (code = 0 AND first cable Label <> 5) then
        choose Label 5
    Else If (code = 1 AND first cable Label <> 6) then
        choose Label 6
    Else If (code = 2 AND first cable Label <> 7) then
```

```
08              choose Label 7
09        Else If (number of cables selected < 2) then
10              Goto 02
11        End If
12 End
```

III. TESTING SYSTEM IMPLEMENTATION

This section describes how the experiment was carried out. Each subsection describes a particular procedure that was used to achieve the objective of this work. Communication between PC's was tested using pinging. Each computer is pinged from the other computer. The next activity was to run the Java program. This process was repeated four times and the average time was computed.

A. Direct Connection between Computers

The figure below demonstrates how the system was set up

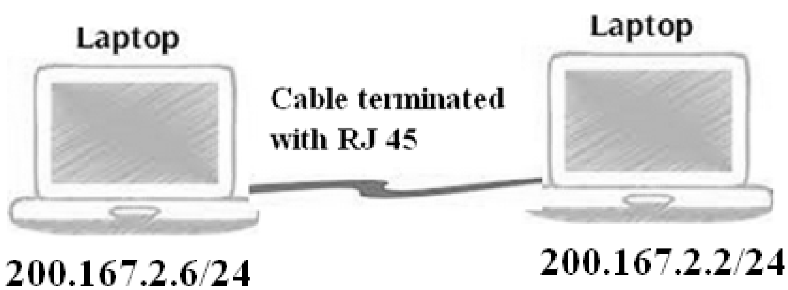

Fig. 5 Setup for Direct Connection between PC's

In the above set up, a cable was selected from Type A and Type B. Two cables Labelled 6 and 7 were selected from Type C using the algorithm in II (D) above. The various steps carried out under this set up were:

Step 1: Connect a cable from Type A
Step 2: Connect a cable from Type B
Step 3: Connect Label 6 cable of Type C
Step 4: Connect Label 7 cable of Type C

In all situations, communication between the PC's was successful. The figure below demonstrates the establishment of communication through pinging

```
C:\Users\KOTR3>ping 200.167.2.6

Pinging 200.167.2.6 with 32 bytes of data:
Reply from 200.167.2.6: bytes=32 time=2ms TTL=128
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128

Ping statistics for 200.167.2.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

Fig. 6 Ping Information on PC2

The figure below demonstrates the successfulness of running the Java Program

```
C:\Users\Sammy\Documents>java Student

Enter the IP address of the host: 200.167.2.2

Trying to connect to source...

Database Connected!

Connection time is: 421 ms

Fetching information from database

Displaying Information in the Particulars Table
************************************************
CSH0501001      Emmanuel Frimpong      Tafo    23
CSH0501002      Vicentia Brimpomaa     Tafo    21
CSH0501003      Joyce Owusu-Dapaah     Suame   20
CSH0501004      Bismark Acheampong     Adum    20

Data Retrieving time is: 7 ms
```

Fig. 7 Results from the Java Program

B. Connection through Switch

The figure below illustrates set up of the testing system. A D-Link switch with five ports was used.

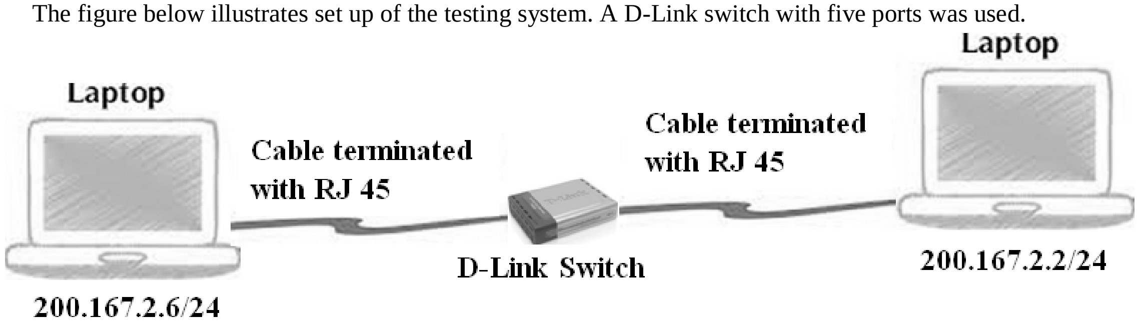

Fig. 8 Setup for Communication between PC's through a Switch

In the above set up, two cables where used from each type. Type C cables were randomly selected using the above algorithm in II (D). Labels 5 and 7 were used. The various steps carried out under this set up were:

Step 1: Connect the systems using cables from Type A
Step 2: Connect the systems using cables from Type B
Step 3: Connect the systems using cables from Type C
Step 4: Connect the systems using a cable from Type A and another cable from Type B
Step 5: Connect the systems using a cable from Type A and another cable from Type C (Label 5 was used)
Step 6: Connect the systems using a cable form Type B and another cable from Type C (Label 7 was used)

In all situations, communication between the PC's was successful. The figure below demonstrates the establishment of communication through pinging

```
C:\Users\KOTR3>PING 200.167.2.6

Pinging 200.167.2.6 with 32 bytes of data:
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128
Reply from 200.167.2.6: bytes=32 time=1ms TTL=128

Ping statistics for 200.167.2.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Fig. 9 Ping Information on PC 2

The figure below demonstrates the successfulness of running the Java Program

```
C:\Users\Sammy\Documents>java Student

Enter the IP address of the host: 200.167.2.2

Trying to connect to source...

Database Connected!

Connection time is: 420 ms

Fetching information from database

Displaying Information in the Particulars Table
***********************************************
CSH0501001      Emmanuel Frimpong        Tafo     23
CSH0501002      Vicentia Brimpomaa       Tafo     21
CSH0501003      Joyce Owusu-Dapaah       Suame    20
CSH0501004      Bismark Acheampong       Adum     20

Data Retrieving time is: 7 ms
```

Fig. 10 Results from the Java Program

C. Connection through Routers

Two Sky Routers, each with five ports were used. All the seven cables were used in this set up. The various steps carried out under this set up were:

Step 1: Cables from Type A connected PC to Router and a cable from Type B connected the Routers
Step 2: Cables from Type B connected PC to Router and a cable from Type A connected the Routers
Step 3: All devices were connected using cables from Type C
Step 4: Cables from Type A connected PC to Router and a cable from Type C connected the Routers
Step 5: Cables from Type B connected PC to Router and a cable from Type C connected the Routers
Step 6: Cables from Type A and B connected PC to Router and a cable from Type C connected the Routers
Step 7: Cables from Type C connected PC to Router and a cable from Type A connected the Routers
Step 8: Cables from Type C connected PC to Router and a cable from Type B connected the Routers

The figure below illustrates the set up of the testing system

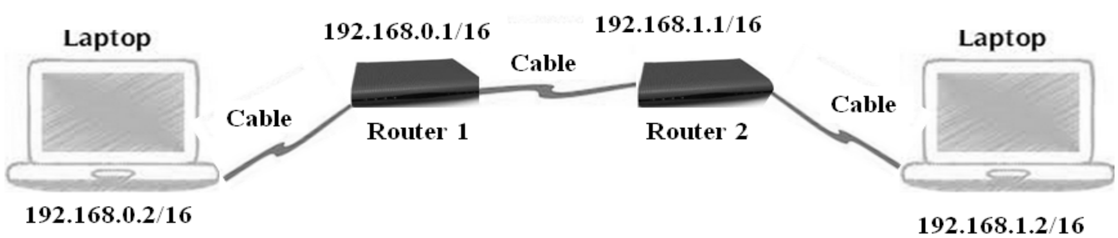

Fig. 11 Setup for Communication between PC's through Routers

In all situations, communication between the PC's was successful. The figure below demonstrates the establishment of communication through pinging.

```
C:\Users\Sammy\Documents>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

Fig. 12 Pinging Router 2 from PC 1

The figure below demonstrates the successfulness of running the Java Program

```
C:\Users\Sammy\Documents>java Student

Enter the IP address of the host: 192.168.1.2

Trying to connect to source...

Database Connected!

Connection time is: 434 ms

Fetching information from database

Displaying Information in the Particulars Table
*****************************************************
CSH0501001      Emmanuel Frimpong       Tafo    23
CSH0501002      Vicentia Brimpomaa      Tafo    21
CSH0501003      Joyce Owusu-Dapaah      Suame   20
CSH0501004      Bismark Acheampong      Adum    20

Data Retrieving time is: 25 ms
```

Fig. 13 Results from the Java Program

.

IV. CONCLUSIONS

The performance of an auto-sensing port network device using unshielded twisted pair does not depend on how the twisted pair cables are arranged and terminated. The time for connecting to the database for the testing system in the various activities is (420 ± 15) ms and the time for retrieving data from the database after establishing connection is (16 ± 10) ms. Auto-sensing port devices are flexible to work with since network designers need not to memorise how straight through and crossover cables specifications are designed. In a situation where a crossover cable is made instead of straight through cable, the cable does not become useless. A twisted pair cable well terminated works very well but it is sometimes difficult to move cables over others and get perfect termination. Scores of manual terminations are sometimes required which usually affect the length of the cable and also results in destruction of RJ 45 connectors.

REFERENCES

[1] W. Wu, "*The Design of a New Network Cabling Experimental Instrument Based on Embedded System*." Advanced Materials Research 328, pages 2427-2431, 2011

[2] A. Tanmay, "*Share Wireless Internet Connection on LAN Using Ethernet Crossover Cable*", 2011.

[3] B. Savitri, R. Goyal, "*Performance Analysis of Secured VoIP Local Area Network*." Proceedings of the 13th WSEAS international conference on Mathematical and Computational Methods in Science and Engineering. World Scientific and Engineering Academy and Society (WSEAS), page 204-209, 2011

[4] "*HP Auto-MDIX technology*", Hewlett-Packard web site. Retrieved February 14, 2013.

[5] HP ProCurve 2910al Installation and Getting Started Guide, March 2010, Pub No. 5992-3084

[6] D. J. Dove, B. W. Melvin, "*Apparatus for Automatically Configuring Media Connectors of a Node Interface*". US Patent 6,460,078 filed October 18, 2000 and issued October 1, 2002
D. Dove, "*1000BASE-T Automatic Crossover Algorithm*". Presentation to IEEE 802.3ab working group. Retrieved on February 15, 2013, Available: http://www.ieee802.org/3/ab/public/feb98/ddmdix1.pdf.

[7] D. J. Dove. "*Apparatus & Method for Automatically Switching Media Connections when Operating in Forced Speed and Duplex Mode*". US Patent 7,366,771 filed March 12, 2002 and issued April 29, 2008.

[8] TrueNet, "*Unshielded Twisted Pair (UTP) Today's Technology for Tomorrow's Network*", ADC Telecommunications, Inc, 2009
Available: http://www.adckrone.com/in/en/library/White_Papers/Enterprise/401262IN.pdf.

[9] Minicom, "*Understanding Twisted Pair Cable Technology*", Connectivity Information Sheet, page 1-7, 2005.

[10] B. Morris, "*Twisted Pair Cabling for Digital Signal*", Kramer Electronics Ltd, pages 1-3, August 2010

[11] J. H. Caltenco, D. E. Cisneros, R. Linares, M. J. Lopez-Bonilla, "*Electromagnetic Couplings in Unshielded Twisted Pairs*", Apeiron, Vol 16. No. 3, pages 439-454, July 2009

[12] T. Robertazzi, "*Introduction to Networks*", Springer Briefs in Electrical and Computer Engineering, page 1-14, 2012

