IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

Link Aggregation for Embedded Applications

Nandini .S[1], Prof. Sankar Dasiga[2]

M.Tech Scholar, Department of ECE, NMIT, Bangalore, India[1]

Professor, Department of ECE, NMIT, Bangalore, India[2]

Abstract: Connectivity is a key feature for digital devices to exist in this connected world. Every device sends and receives the data. As the demand of data communication is increasing day by day has resulted in enhancements of capacity and QoS of interfaces and devices supporting more than one communication types. The subsequent evolution to enhance the data communication speed is to aggregate the multiple physical channels, aka Link Aggregation. Link aggregation is already in use for faster data syncs across servers using dedicated links and, IEEE has already come up with the standards for the same. With the rapid strides in the integration technology etc yet another trend that is emerging currently is the ever increasing capabilities and applications of embedded systems. Embedded Systems too now simultaneously support different types of communication interfaces!! It is envisaged that "Link Aggregation" would soon become a requirement for embedded systems, and this project work was taken up towards studying the possibilities. Raspberry Pi is an open and also regarded as a good reference platform for Embedded Systems development was chosen for the study in this project work. Ethernet and Wi-Fi are the most common and standard communication interfaces supported in almost every digital system, and, TCP, UDP and SCTP are the very commonly employed protocols for data communication. Therefore these interfaces and protocols have been selected for the study and comparison of performance of data communication between computer and embedded platform in this project work. Having collected the data and done some analysis, few recommendations for further enhancements and work were also made from this work.

Keywords: Link Aggregation, data rate, client server communication, point to point communication, Raspberry pi, protocols, performance.

I. INTRODUCTION

In the present world, embedded systems play a very important role. Embedded systems perform various tasks to make our life easy and cost- effective solutions are offered by them. A number of development platforms are available along with tools, libraries and peripheral devices for the development and hands on study of embedded systems. Microprocessor / Microcontroller is the heart of embedded system. Embedded Systems are very commonly used in home, automobiles, workplaces, PD's etc. Mobile phone is one such embedded device. Mobile with single processor can perform wide variety of tasks, from answering a call to data communication through the internet. One such task is the Link Aggregation.

Link Aggregation is the combination of multiple physical links to single logical link. A group of physical links making a single aggregation is called Link aggregation group. IEEE 802.3.ad-2000 specifies LACP protocol to outline the aggregation. Link aggregation increases the bandwidth, availability, load sharing, auto configuration, reduces the duplication risk, deterministic behaviour.

Benefits of link aggregation:

1. Better connection availability in case of single link failure.
2. Due to load balancing, there is a better utilization of physical links and deterministic behaviour.

While link aggregation is widely in use with general purpose / computer systems, its use with embedded systems is not prevalent

II. LINK AGGREGATION

Link Aggregation is the combination of multiple physical links to single logical link. A group of physical links making a single aggregation is called Link aggregation group. IEEE 802.3.ad-2000 specifies LACP protocol to outline the aggregation. In earlier days dial-up modems were limited to 3-5KB/s. A variety of techniques were developed for increasing the speed. One such technique was "shortgun".

This technique consisted of two modems, each operated in parallel at 5.5 KB/s in order to get the aggregated throughput of 11KB/s. Both the modems operated in parallel to almost double the speed. Thus link aggregation is joining different various system connections in parallel to expand the past what a solitary association could maintain furthermore gives repetition on the off chance that if any of the connection comes up short.

Link aggregation increases the bandwidth, availability, load sharing, redundant, auto configuration, reduces the duplication risk, deterministic behavior. It also prevents mis-ordering or duplication of frames and traffic is distributed packet by packet. Some of the pre-requisites for link aggregation are

1. The links used should be in full duplex mode.
2. The links should be parallel to each other.
3. The links should be point to point connected.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

III. CLIENT SERVER COMMUNICATION

Server and Client are main elements in all kind of Internet Applications. A request is sent to server by client and server listens to these requests and server will fulfill the request and respond back to the client. For this client server communication a common protocol is been used. Here the concept of client server communication is used for link aggregation where my laptop acts as client which uses Ubuntu operating system and raspberry pi acts as server which uses Raspbian as its operating system. Request–response messaging pattern is used to exchange messages between the clients and servers. The client sends a request, and the server responds to it. Client Server Communication is an example for inter-process communication. For this communication between a client and server, a common language and some set of rules are followed and theses rules are defined in a communications protocol. Protocols being used for the work are TCP, UDP and SCTP.

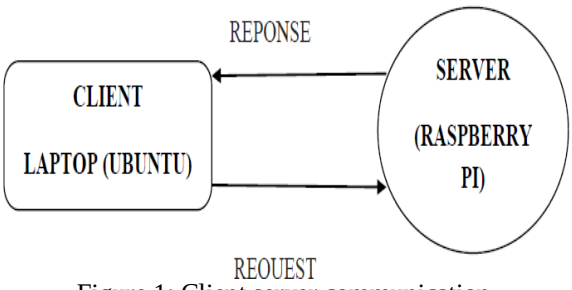

Figure 1: Client server communication

A. Point to Point Connection.

As we know one of the prerequisite for link aggregation is that, the links should be point to point connected. Hence communications between two nodes or endpoints is called point –point communication. An example is a telephone call, where one telephone is connected with other, and whatever is spoken by one caller can only be heard by the other. P2P is point to point communication. For both wired or remote association, point to point association can be set up. Since point to point connection links is one of the prerequisite of link aggregation, in this work P2P connection is done between the raspberry pi and the laptop using an Ethernet cable (wired) and Wi-Fi(wireless), where raspberry pi acts as a server and laptop acts as a client. Ubuntu OS is installed in laptop.

B. Concurrent Server

Since we are working on Link aggregation where multiple need to handled. Hence many connections are handled simultaneously by the concurrent server. While processing the previous requests, it can accept the new connection requests. Concurrent server is implemented by calling a fork () function. Each time a fork () is called, it creates a separate process. That is a current process is divided into two, a parent process and a child process. The ways in which the child process varies from parent process and child process is that child process has unique process ID and it has only one user thread. Along these lines, for long exchanges, an alternate kind of server is required — the simultaneous server, as appeared in Figure 2. Here, Client A has officially settled an connection with the server, which has then made a youngster server procedure to handle the exchange. This permits the server to process Client B's solicitation without sitting tight for an exchange to finish. More than one youngster server can be begun along these lines.

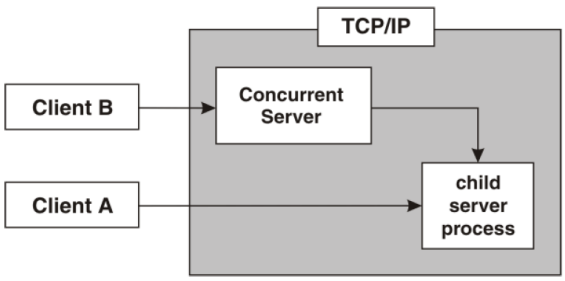

Figure 2: Concurrent server

C. Socket Programming

Once the point to point connections are done, my task was to transfer the data between the client and server. Socket programming is used to establish a connection and transfer data between client and server for protocols TCP, UDP and SCTP protocols. Socket is key advancements of PC systems administration. Applications speak with the assistance of sockets utilizing some standard instruments which are incorporated with system equipment and working frameworks. Socket innovation has been utilized for about two decades.

A solitary socket speaks to a solitary connection between two bits of programming/software. Yet, more than two software can be spoken with customer server framework or appropriated frameworks (for instance, numerous Web programs can s speak with a solitary Web server all the while) however we require different attachments to do this . Thus customer server model is utilized. Sockets are bi-directional, that is both the sides of the connections are equipped for sending and accepting information simultaneously.

Operating systems like Unix, Windows, MAC and many other operating systems support sockets. Socket is a combination of port and IP address of a particular system. Sockets also support many languages like C, C++, Python, Java, etc. Socket is an abstraction through information can be sent or received by an application and also provides access to inter process communication services.

There are two types of sockets,
(i) Stream sockets and (ii) Datagram sockets
Reliable byte-stream service is provided by the stream sockets and datagram sockets provide best-effort service. TCP uses stream sockets and UDP uses datagram sockets. Stream socket is also called connection oriented socket and datagram socket is also called a connectionless socket.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

IV. DESIGN APPROACH

The Block diagram of the proposed system is shown in figure 2.

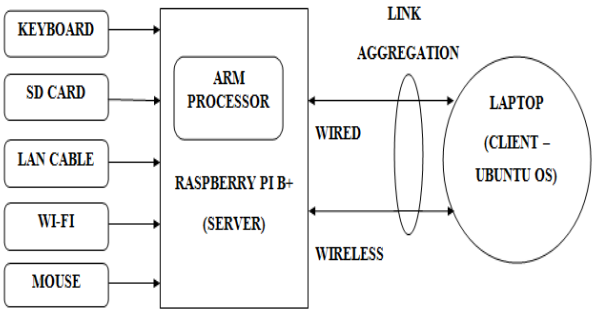

Figure 3: Block diagram of the proposed system

This design approach consists of the following:

A. Raspberry PI

Raspberry Pi is an open and also regarded as a good reference platform for Embedded Systems. In the project at first Raspbian OS is been installed into SD card and is placed in the SD card socket. The keyboard and mouse connected to the hardware platform through USB which is used for operating Raspberry pi and programming purposes. HDMI to VGA converter is connected to the HDMI port of the pi to display the OS on to the monitor. Once these initial connections are done, hardware is powered up through the nano USB port. Some amount of time is taken by the operating system to boot and check whether all the interfaces are working fine. Connection to the internet is possible by connecting the Wi-Fi adaptor via USB PORT or the LAN via Ethernet port. Both the LAN cable and Wi-Fi are used in our project.

B. Link aggregation

Link Aggregation is the combination of multiple physical links to single logical link. A group of physical links making a single aggregation is called Link aggregation group. Link aggregation increases the bandwidth, availability, load sharing, auto configuration, reduces the duplication risk, deterministic behaviour. As we know that the prerequisite of link aggregation is point to point connection, therefore once installing the OS for the embedded platform is done our next task is to establish point to point connection between the client(laptop) and server(Raspberry pi).Here we are using two interfaces(links) (i) wired link(LAN cable) and (ii) wireless interface (Wi-Fi).Both these links are point to point connected between Raspberry Pi (server) and laptop (client), which are then aggregated to form one logical interface in order to transfer a huge file. Huge files of different sizes are been transferred between server and client using socket programming concept for TCP, UDP and SCTP protocols for both link aggregation and single link transfer . the results are compared to show the better performance of link aggregation when compared with single link transfer.

V. RESULTS

Both the server codes and client codes for link aggregation and for single link are executed by the server and client respectively for all the three protocols (TCP, UDP, SCTP). The codes are executed for six different file sizes, 2MB, 2.5MB, 3MB, 3.5MB, 4MB, 5MB. Once the codes are executed, total time taken to transfer file and throughput which determines overall performance is calculated. Each file is transmitted ten times, each time the file is transmitted throughput and time taken is noted down. Finally average of ten attempts is calculated for both time taken and throughput. The same is followed for both link aggregation and single link transfer. Once the averages of ten attempts are calculated for both link aggregation and single link transfer for all three protocols, the results are compared separately for each protocol. The readings calculated and the graphs obtained are as below:

A.Transmission Control Protocol (TCP)

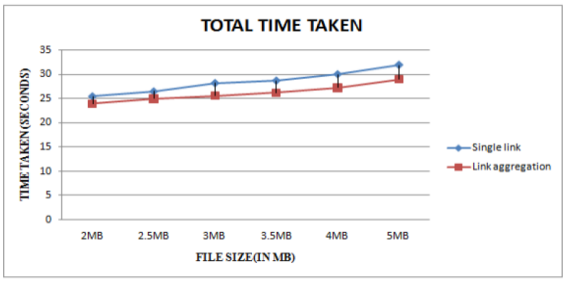

Figure 4: Total time taken for both link aggregation and single link using TCP.

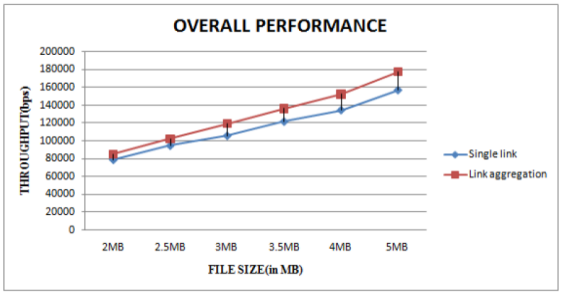

Figure 5: Overall performance of link aggregation and single link using TCP.

B. User Datagram Protocol

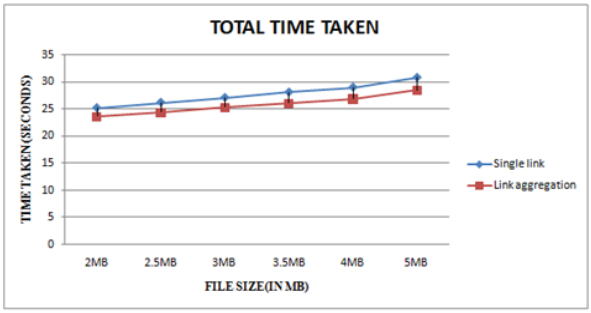

Figure 6: Total time taken for both link aggregation and single link using UDP.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

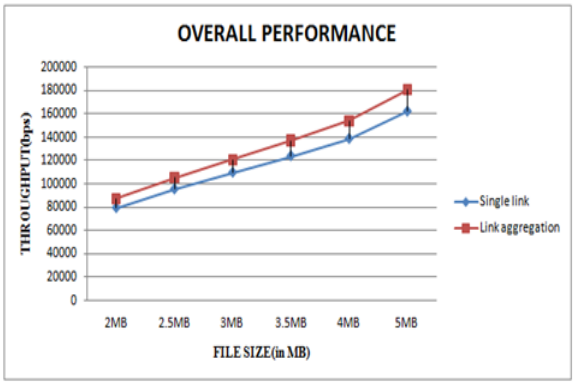

Figure 7: Overall performance of link aggregation and single link using UDP.

C. Stream Control Transmission Protocol

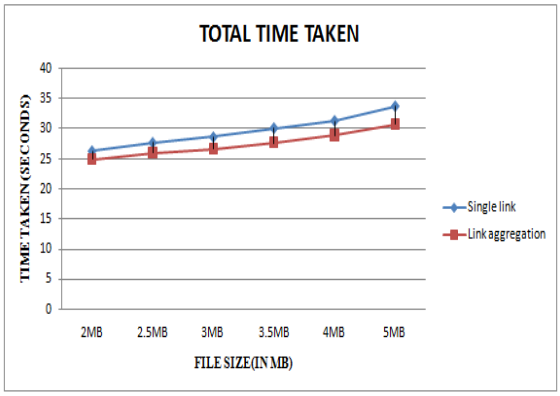

Figure 8: Total time taken for both link aggregation and single link using SCTP.

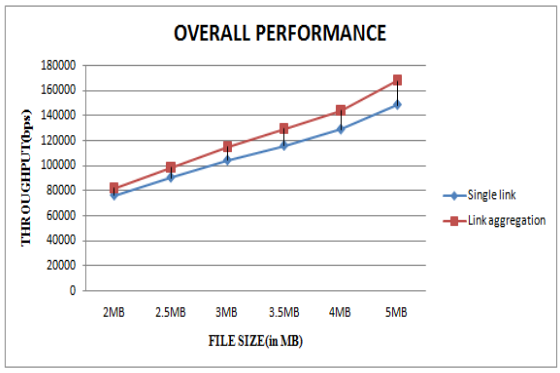

Figure 9: Overall performance of link aggregation and single link using SCTP.

VI. CONCLUSION

In this project i have compared the performance of link aggregation and single link transfer in the context of Embedded Systems. This is done with help of coding part, where the codes were written to transfer the files and to calculate the performance. The codes were written for both client and server for all the three protocols using socket programming. In socket programming, sockets were used to communicate the data between client and server. The performance was observed by transmitting files of different file sizes.

Once the results were calculated, it was observed that the performance of the link aggregation is better than single link transfer. The amount of time taken by link aggregation to transfer different files of different sizes is less when compared to single transfer. Since we are checking the results for three different protocols(TCP,UDP and SCTP), it was observed that for all the three protocols performance of link aggregation is better than single transfer and amount of time taken by link aggregation is also less when compared to single link transfer.

It is observed that the benefit of link aggregation while transferring smaller files is not significant due to overheads involved in the handshake etc in the connection establishment process. As size of the file for transfer increases the benefits of link aggregation became more and more evident.

We have successfully tried link aggregation in the context of embedded systems where as we all know support for multiple links is becoming a commonality, like the Wi-Fi and BT availability with our cell phones. However the transmission is limited with a limitation of computational power and availability of memory. As we can see embedded systems are growing day by day and are becoming very powerful. They can soon be candidates for transferring even more complex and huge files and many more complex tasks whereby requiring link aggregation.

REFERENCES

[1] A Novel Bandwidth Aggregation System Using Multiple Physical Links" by Soma Bandyopadhyay Innovation Lab Tata Consultancy Services, Arpan Pal Innovation Lab Tata Consultancy Services, Shameemraj Nadaf Innovation LabTata Consultancy Services.
[2] " LTE-WiFi Link Aggregation at Femtocell Base Station" by Yuji Kojima1, Junichi Suga1, Takeshi Kawasaki1, Masato Okuda2 and Ryuichi Takechi1 1FUJITSU LABORATORIES LTD, 2FUJITSU LIMITED.
[3] "LTE-WiFi Link Aggregation at Femtocell Base Station" by Yuji Kojima1, Junichi Suga1, Takeshi Kawasaki1, Masato Okuda2 and Ryuichi Takechi1 1FUJITSU LABORATORIES LTD, 2FUJITSU LIMITED ,japan.
[4] "Introduction to Sockets Programming in C using TCP/IP " Professor: Panagiota Fatourou ,TA: Eleftherios Kosmas.
[5] http://www.makeuseof.com/tag/ubuntu-an-absolute-beginners guide/
[6] http://www.instructables.com/id/Raspberry-Pi-B-boot-from-SD-Card/
[7] http://elinux.org/RPi_Easy_SD_Catprd_Setup
[8] http://askubuntu.com/questions/22835/how-to-network-two-ubuntu-computers-using-ethernet-without-a-route
[9] .http://www.speedguide.net/faq/how-to-connect-two-computers-directly-using-a-wireless-275
[10] http://www.linuxjournal.com/article/2333?page=0,1
[11] http://www.raspberrypi.org
[12] http://www.elinux.org