Study of voice data communication using network streams on dataflow programming

Barlian Henryranu Prasetyo*, Dahnia Syauqy, Rizal Maulana, and Gembong Edhi Setyawan

System and Robotics Lab, Faculty of Computer Science, University of Brawijaya, Indonesia

Abstract. In this paper, we implement a voice data communications between sender and receiver through Wi-Fi. The system was designed and implemented using network stream protocol which was developed by National Instruments on Lab-VIEW between two devices. The device which acts as the sender was My-RIO board and the receiver device was Personal Computer. The voice data communication system was tested by observing the effect of various buffer sizes on the throughput, latency and CPU usage. The result showed that the larger the buffer size being used, the throughput values were also getting bigger. However, particular buffer size made the throughput became saturated. The change on the buffer size also directly affects CPU usage. It seemed that the buffer size did not affect latency significantly. It showed that the latency was stable with an average of 13 milliseconds on voice data transmission.

1 Introduction

Authentication process is a required part in a smart home system to ensure the safeness of its control system [1]. Set of codes implemented as authentication such as Personal Identification Number (PIN), smart cards and passwords methods are shown to have weaknesses [2]. Authentication system using human voice as a password code replacement has become new alternative in Smart Home [3-4].

Generally, human voice which is used as biometric authentication is captured by microphone embedded into node device. Since node device normally does not have high specs hardware, the captured voice will be sent to processing device which can perform faster computation. Therefore, methods to send the voice data become interesting [5].

The effectiveness of a data communication system depends on four fundamental characteristics, namely packet loss, throughput, delay and jitter [6]. In order to support real time communications between nodes, other than it requires lossless type of communication, it also requires type of communication that has the characteristics of a high throughput. In addition, for a command based system, low latency is also preferable to achieve dependable system [7].

In our design, there are 2 nodes which acts as sender and receiver. The sender nodes captures any voice command and directly send it to the receiver nodes that recognize the command. Since both nodes have already had complex function, CPU utilization in both nodes are also need to be monitored to ensure all function works properly [8].

There are three models of communication that have

special characteristics. Network communication model characteristics are shown in Table 1. Table 1 can be used as a reference to determine the model and the network protocol used in the application configuration and system requirements [9]. By analyzing the system requirement, the appropriate type of voice communication is using streamed/ buffered model type.

Table 1. Communication model characteristics [9]

Communication Type	Configuration	Characteristics	Requirement
Message Base	1:N	Even driven, command	Low Latency, Guaranteed Delivery
Process Data	N:1	Single Point, Current Value	Latest Value Instead of Guaranteed Delivery
Streaming/Buffered	1:1	Continuous Data Transfer	High Throughput, Guaranteed Delivery

The data types in communication such as TCP and UDP is formatted in string. It means that the sender must flatten all type of data into a string and the receiver must unflatten the received data from string. Therefore, additional layers need to be added to the complexity of the data transmission of non-string. In this paper, we used Network Stream Protocol in Lab-VIEW which was developed by National Instruments because it can transfer most data types including clusters and arrays. But before implementing on a real machine, we need to test the system performance to see the influence of varying buffer size to the throughput, latency and CPU usage.

Fig 1. Sequence message diagram lossless data communication

2 Algorithms

Network stream communication is a point-to-point where the data are maintained until all data is sent or until the stream is disconnected. The data transmissions are obtained through the use of FIFO buffer on the endpoint and through the acknowledgment and flow control on message protocol. The FIFO buffers keep the application that contains the writer endpoint of the data overwrites if the data writing process is faster than the network transfer rate. Message acknowledgment serves to prevent loss of data from the network termination, and message flow control prevents the application that contains the writer endpoint from overwriting the data if the data is generated faster than the data reception by the application that contains endpoint reader. Figure 1 shows a sequence message diagram lossless data communication [10].

3 Research Methods

The Lossless communication by using network stream protocol requires two devices that will be used to transmit data to each other, which in this case are My-RIO and a PC. My-RIO is an evaluation board which is made by National Instruments (NI). NI My-RIO provides FPGA integrated with processor which can be programmed using Lab-VIEW. This evaluation board is also equipped with onboard accelerometer, analog and digital I/O, USB port and also audio jack [11].

The voice data is sent from the microphone that is connected to a 3.5mm port on My-RIO board. The voice data input will be perceived as analog data on the single writer section to write in the network streams on My-RIO. Before single writer transmit the voice data, first it requires creating endpoint at the initialization block on each side of the network stream. The initialization was done by setting parameters including the size of the buffer, sender/recipient ID and sender address/recipient data while creating the endpoint. After the initialization is done, the data can be sent from a single write My-RIO towards single read PC using Wi-Fi communications.

The process is done by sending the data directly from single write block using Wi-Fi communication to the single read contained in My-RIO. Once the data is received single read My-RIO, the destroy endpoint blocks will remove all endpoints in My-RIO. To perform the latency test, the received data on the PC need to be sent back to the My-RIO to determine the round trip time two-way communication. Therefore, we need to create Single Write and Single Read block in both devices. Block diagram of network stream system can be seen in Figure 2

Fig 2. Network stream system block diagram

Fig 3. Network stream state diagram

The process of connection creation from one host to another is quite complex and involves multiple exchange of packets between two endpoints. Therefore, it requires management connection between endpoints. The state diagram for a network connection process stream can be seen in Figure 3.

The process is started from system initialization. In its default state, the system had an idle status. Then, in idle status, network stream makes connections to the endpoint. If the endpoint has not been established, the network stream will make the first endpoint. After that, the system back to the default status idles. Then, the system will try to make a connection because the endpoint has been made so that network stream check the condition of the destination endpoint. There are 3 possible conditions, i.e. write, read and destroy. When the endpoint is ready to write, the system performs a write stream. When the endpoint destination is ready, then system will read the stream and when the connection is timed out, the system will destroy the endpoint. The endpoint uses a FIFO buffer to receive and transmit data.

The system was implemented in Lab-VIEW 2014.

Before implementing on real voice data, first we simulate the system using sensor data generated by onboard sensor. After getting appropriate parameters, we simulate the system using real voice data. Overall, the system was simulated in two phases:

- Simulation of Data Sensor (data z axis accelerometer sensor)
- Simulation of Voice Data (voice from microphone)

4 Result and Discussions

Network stream is a lossless, unidirectional, one-to-one communication consisting of a writer and a reader endpoint.

4.1 System Modeling

The system block diagram modeling at the sender which was implemented in My-RIO device can be seen in Figure 4. The Figure shows that the system will perform a loop for 60 seconds. This duration will be used to count how many data can be passed within each second. Before the looping process it initialize parameters including buffer size, sender ID and sender address in the endpoint creation block. During the process of looping occurs, the system retrieves the data on the z-axis accelerometer and then writes in a single write block. The data are then transmitted to the Reader Endpoint. The data retrieval process from the accelerometer and data writing on a single write block rely mainly on the processor speed which is being used without implement any delay. After the looping process is complete, the system will flush and destroy the buffer stream.

Fig 4. Blok diagram modeling of the system at endpoint writer

Fig 5. The block diagram modeling of the system at endpoint reader

The block diagram modeling of the system at reader endpoint can be seen in Figure 5. The Reader Endpoint initializes the buffer size, sender ID and the sender address. The buffer size of Endpoint Reader must be equal to the size of the buffer in the Endpoint Writer. Similar to the Writer Endpoint, the system will loop for 60 seconds. During the loop run, the system will receive the network stream data from the Writer Endpoint. After the loop is finished, the system will flush and destroy the buffer stream.

4.2 Testing Using Sensor Data

The testing focused on throughput, latency and CPU usage while varying the buffer size parameters which is used. In the first phase, we use sensor data as the input, and in the second phase we use real voice data as the input.

The network stream throughput testing aims to determine how much data can be sent in the system every second. The Z-axis accelerometer sensor data contained in My-RIO were sent continuously to the PC for 60 seconds. The z-axis accelerometer data on My-RIO were sent without delay. The data were sent as elements in which one element refers to DBL data type (64 bits). The test was done by giving variation in the buffer size that is used, and observe the effect on the throughput. The throughput testing result can be seen in Figure 6.

Fig 6. The throughput and CPU utilization testing result

Figure 6 shows that the throughput is linearly influenced by the size of the buffer. However, when the size of the buffer is above 100000, the throughput will be

saturated despite the enlarged size of the buffer. In addition, it should be considered that by increasing the size of the buffer, and then the CPU usage on My-RIO also will increase.

The Latency test aims to determine the processing time required to transmit a data from My-RIO to the PC. The data packets from My-RIO were sent to a PC, and then after the data were received by the PC, the data were directly sent back to the My-RIO. The data transmission was done as many as 1000 iterations. The round trip transmission was assumed to be similar therefore the result will be divided by 2. The latency testing result can be seen in Figure 7.

Fig 7. The latency testing result

4.3 Testing Using Voice Data

The voice was captured by a microphone which was connected to a 3.5 mm port on My-RIO. The voice duration is only set at 3 seconds since the voice is only used as short command. In this model, the buffer size is made permanent with value of 100000 as the results from previous throughput sensor data testing.

Figure 8 shows the front panel for testing the voice data. The voice data recording can be performed after the light turns on. The user is given only 3 seconds to record sound through a microphone. The top graph signal is a voice signal which has been recorded and is ready to be sent My-RIO to the PC. While bottom charts signal is a sound signal that has been sent back from the PC to My-RIO.

Fig 8. The front panel display using voice data

Figure 9 shows that the voice data latency testing is done by using four different commands, with each command is repeated three times. From the results, although there are differences in command, but the value of latency is not much different. The average latency value of this system is 13 milliseconds.

Fig 9. The latency voice data testing result

5 Conclusion

Network stream protocol supports transmission of all data types without normalizing the data to the intermediate data type. Network streams are designed and used for lossless communication, where data is maintained until all data sent and certainly there is no data transmission failure.

The throughput test is done by sending the data on MyRIO heads to PC while varying the size of the buffer. From the test results, the larger size of the buffer was being used the throughput values are also getting bigger. However, in certain circumstances, although buffer size is enlarged, the throughput is saturated.

The change on the buffer size also directly affects CPU usage. The larger the buffer size, the CPU usage on MyRIO will also increase. So the selection of the buffer size must be adapted to the complexity of the system to be designed.

In Latency test using the sensor data, the data were sent as many as 1000 iterations while varying the size of the buffer used. From the test results, the size of the buffer does not affect the speed of data transmission process. The latency is always stable with an average of 12 milliseconds even though the size of the buffer increases. In the latency test using sound data, the test is done by sending 3 seconds voice duration data using four different short commands and with buffer size of 100000. The result shows that the latency is stable with an average of 13 milliseconds.

References

1. S. Chitnis, N. Deshpande, A. Shaligram, An Investigative Study for Smart Home Security: Issues, Challenges and Countermeasures, *Wireless Sensor Network*, 8, 61-68. (2016)
2. Lawton, Biometrics: A New Era in Security, *IEEE Computer*, pp.16-18. (1998)
3. O. Joymala, N. Khare, Securing a Smart Home Network using Voice Biometric, *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*, 5:2. (2016)
4. S. Amrutha, S. Aravind, A. Mathew, S. Sugathan, R. Rajasree, and S. Priyalakshmi, Voice Controlled Smart Home, *International Journal of Emerging Technology and Advanced Engineering*, 5:1. (2015)
5. T. Obaid, H. Rashed, A.A. El Nour, M. Rehan, M.M. Saleh, and M. Tarique, Zigbee based Voice Controlled Wireless Smarthome System, *International Journal of Wireless & Mobile Networks (IJWMN)*, 6:1. (2014)
6. A.F. Behrouz, *Data Communication and Networking*, USA: McGraw Hill. (2007)
7. T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S.

- Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura and J. Yamato, Low-Latency Real-Time Meeting Recognition and Understanding Using Distant Microphones and Omni-Directional Camera, IEEE transactions on audio, speech, and language processing, vol. 20, no. 2, February (2012)
8. S. Nejati, S. Di Alesio, M. Sabetzadeh and L. Briand, Modeling and Analysis of CPU Usage in Safety-Critical Embedded Systems to Support Stress Testing, Model Driven Engineering Languages and Systems pp 759-775 (2012)
9. National Instruments, Using the Right Networking Protocol, <http://www.ni.com/white-paper/12079/en/>, access on April 15th 2016. (2013)
10. National Instruments, Network Streams Functions, http://zone.ni.com/reference/en-XX/help/371361H-01/lvcomm/ns_networkstreampalette/, access on April 15th 2016. (2011)
11. National Instruments, NI MyRIO-1900 user guide and specifications, <http://www.ni.com/pdf/manuals/376047a.pdf>. (2013)