Design and Development of PC based Multicast Count down Time Generator and Receiver Application for Timing activities at Test Range during Missions

Chandradeo Kumar Rajak
Scientist
ITR Chandipur, DRDO

ABSTRACT

Count Down time (CDT) is of crucial importance during launch of flight vehicles like Missile, UAVs in a test Range. A Remote tracking station is not able to know about status of Count Down activities without seeing CDT. Conventional CDT systems are hardware based. These systems are bulky, costly and very difficult to reconfigure. This paper discusses about software based Count down Time generator application which can run on any desktop or laptop. This application is GUI based which takes inputs from the users and generates the Countdown Time. The generated CDT is transmitted into the network using multicast technology which is received and displayed by Multicast CDT Readers deployed over the network. As this application is completely software based hence it can be configured to run on any of the Desktops or laptops which are easily portable. As delay and time jumps are crucial issues in Range Timing hence special mechanism are applied to address them.

Keywords

Range Timing, Application level Multicasting, Socket Programming, Count Down Time, GUI Design in Qt.

1. INTRODUCTION:

Due to advancement of computing power in Desktop and Laptops machines it has become convenient to develop mission critical applications and host on these platforms. IDEs (Integrated Development Environment) like Qt provide a rich set of Widgets for designing user friendly GUI. Qt. is also a right platform for developing real time Socket based applications because it provides APIs called QUdpSocket for receiving and transmitting data in real-time. The CDT generator application takes advantage of Multicasting for sending data in traffic efficient manner. So here first different types of communication over a computer network will be briefed, subsequently the design aspect of CDT generator will be described.

Communication over a computer network can be categorized broadly into following categories. Unicast, Broadcast, Multicast and Anycast. **Unicast:** Unicast is a communication technique in which there is one sender and one receiver. It is one to one communication in which the server transmits the data destined to a particular IP and port no and the client listening on that port receives the data. **Broadcast**: It is one-to-many type of communication in which the sender sends the data to all systems on the network. Here the data is sent on a particular port no but there is no specific IP addressed specified. Hence the data is pushed to all hosts on the network and hence other systems over the network get overwhelmed with the unnecessary traffic. Even if a system is not interested in data it is being forwarded to it .This type of communication is not suitable where bandwidth is limited because unnecessary traffic is pushed into the network. **Multicasting**: Multicasting is a data transmission technique on computer network which a limited broadcast type. Here the communication is from one sender to a group of receivers [1][2]. Data is sent to a particular group called multicast group. In this Instead of sending multiple copies of the same data to multiple receivers, multicast replicates the packet at the router where there is more than one receiver interested in data. Any receiver which wants to receive data can join the group and receive. In broadcast communication the data is sent to all receivers on the network. In **Anycast** [3][4] which is supported by IPv6 protocol the data is sent to the nearest receiver in the group. This is one-to-nearest association. There are many receivers which are within same group address but the data is forwarded to only that member which is topologically nearest in the group. Software based CDT generator and Receiver are applications developed in Qt. These applications have been developed in an effort to convert legacy hardware based systems into more flexible and IP enabled software based systems.

PC based multicast Count down Time generator generates UDP based multicast data and sends by multicasting it. In this way the data is sent to only a particular group of countdown time receiver clients. It provides advantages of bandwidth saving and minimal management overhead as it does not require keeping IP address record of all clients. So multicast is advantageous over broadcast. Multicast Technology uses class D addresses in the range 224.0.0.1 to 239.255.255.255. This class of IP addresses is used only for multicasting purpose. Client and server both use this IP address in their applications apart from the device IP address of network they belong to. In the forthcoming section system configuration, design of generator and receiver applications , implantation details of Applications will be discussed then and finally the paper will be concluded with the discussion on with results and their analysis.

2. SYSTEM CONFIGURATION

As shown below in fig-01 the PC Based CDT generator is interfaced with the Range Communication network. The multicast clients deployed at all remote stations receive count down time through multicast enabled applications. The PC based CDT generator application can be installed on any Desktop or laptop machines. In this case it has been run on intel core-i3 based processors on Laptop running windows 7 and windows 8. The network interface card (NIC) was of Netgear make and it was enabled for multicast by default.

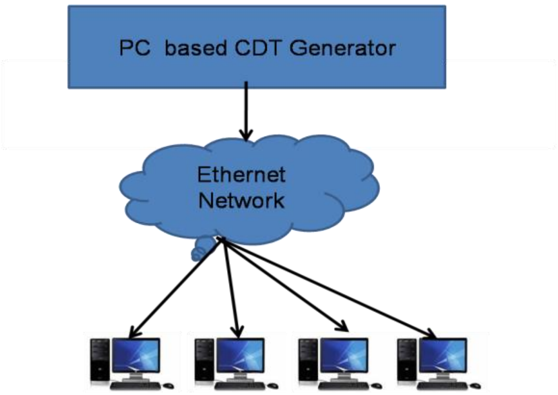

Fig 1: CDT Server and Client Configuration

The multicast clients are also PC based multicast applications. These multicast clients can be managed remotely using VNC remote desktop utility. There are 200 multicast clients deployed throughout the range network. They provide a real time display about Count down Time status in Range. The CDT generator is kept at a central control console .According to the mission requirements the outputs of the time can be formatted.

3. DESIGN OF COUNT DOWN TIME GENERATOR APPLICATION

Given below in fig-02 is the flow chart of the countdown time generator. The application takes inputs from the user through GUI in format hh:mm:ss. Initially the flag is kept 'd' i.e. down mode. If the user presses the hold button the flag is set to 'h' or if the all values of hour, minutes and seconds become zero the flag becomes 'u' i.e. counting Up mode. Hence depending upon the values of flag the time is generated in down, hold or up mode. The generated time along with the status flag of D, H or U is converted into a UDP packet of 14 bytes and sent to the network by multicast technique. Packet format of output time is like 00:05:03:20:0H where 0H indicates the Hold status of CDT. This packet is transmitted into network using a timer of 200ms. This same Timer is used to update the GUI of application in real time. The Socket Programming in Qt [4] has been used to send UDP multicast packets. There are Multicast clients deployed over the network which join to the multicast group receive the countdown time and display into GUI in Hour: Minutes: Seconds format. The status of the time is indicated using Led color by setting it to green or red.

The Design of CDT generator application is completely based on object oriented concept. Qt provides everything like Timer,Widgets ,UDPSocket implemented as Class. For using these we first create objects of the corresponding class and use in application. The Design of the User Interface has been kept as simple as it could be easily understood by a novice user. There is facility of user feedback about actions like button pressed by using QSound Class which gives a beep sound once any button is pressed.

4. IMPLEMENTATION DETAILS

PC based multicast CDT generator application has been implemented using Qt IDE[5][6]. Qt. is C++ based Graphical User Interface application development framework. It provides a rich set of GUI controls called Widgets. Qt. provides a rich set of API for network programming [7][8] like QUdpSockets , QHostaddress which has been used in this application for multicasting generated CDT. It also provides API like QTime and QTimer which is used for generating the CDT as well as sending the data at a regular basis in 200 milliseconds interval.

Shown above in fig-03 is the snapshot of Qt IDE. Qt has a module called QtCreator which provides rich set of features for designing user Interfaces. PC based CDT generator uses Widgets like Textbox, Buttons and Labels. The application GUI has a user friendly integrated Keypad through which users can directly provide inputs through mouse. The ease of use enables any normal computer savvy user to operate the application.

The Qt. IDE can be run on any platform like Linux and Windows. There are versions of Qt which can be used for software development for embedded devices and mobile device platform like android. This application has been compiled on both windows and Linux based machines. RHEL-6, Windows 7 and windows 8 versions satisfactorily run the CDT generator and receiver.

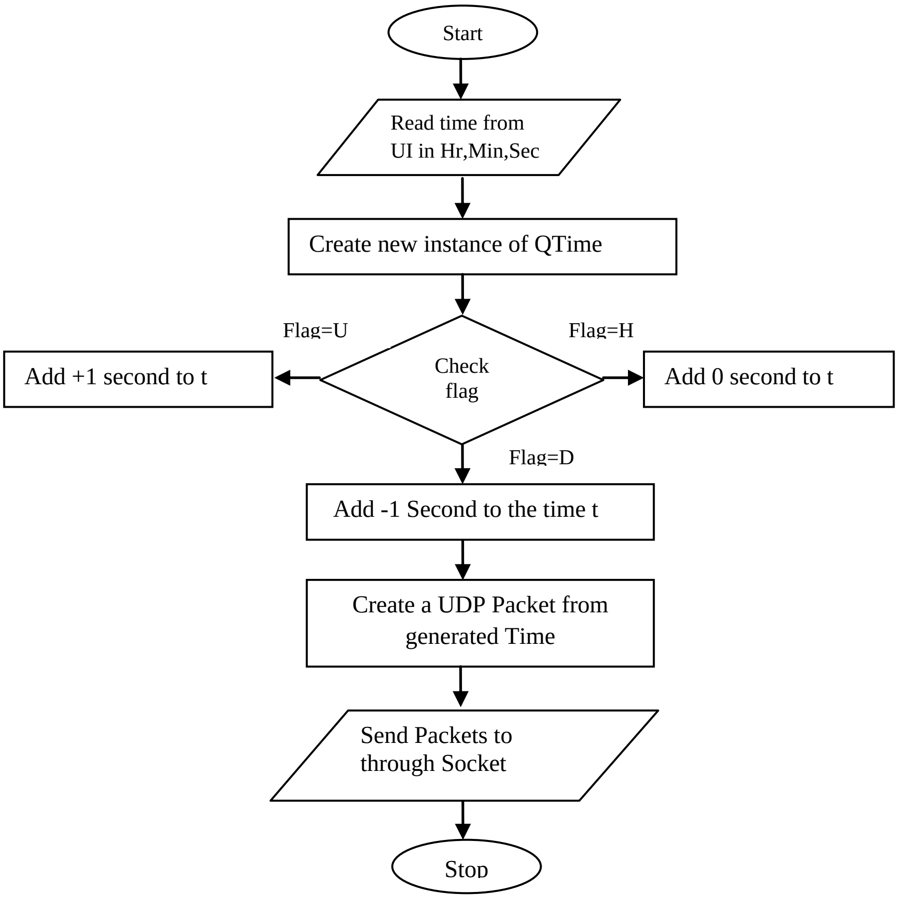
Start
Read time from UI in Hr,Min,Sec
Create new instance of QTime
Flag=U
Flag=H
Check flag
Add +1 second to t
Add 0 second to t
Flag=D
Add -1 Second to the time t
Create a UDP Packet from generated Time
Send Packets to through Socket
Stop

Fig 2: Flowchart of CDT Generator Application

5. OPERATING PROCEDURE OF CDT GENERATOR

The application is pre-configured with appropriate IP address of multicast group on which the clients are receiving. User installs this application on his computer. After launching the application user has to enter time in Hr,Min and Sec textboxes using inbuilt keypad. Then user has the choice of RUN and PRERUN buttons. If the clock has to be started from next full minute the RUN button is to be pressed and if it has to be started immediately the PRERUN button has to be pressed using mouse. The application starts generating and disseminating the time to network. If at any instant there is a need of hold to be given to the counting time the user has to press HOLD button. After issuing HOLD the same button text changes to CLR Hold so user can press the same button to clear the hold. Once hold is cleared any of the Run or Prerun button can be pressed according to requirement. Finally when the hr, min and sec become all zero then clock automatically goes to count Up mode. As The system time of the computer on which this application is running is synchronized using NTP server, accurate T0 time i.e. actual time when CDT becomes 0 is displayed into the GUI. Hence any user who has operated the conventional Clock can easily operate the same without additional training.

6. CDT RECEIVER APPLICATION

The time generated by CDT generator is transmitted into the network. This time has to be displayed by receivers. Receiver application is also a GUI based application developed in Qt. A UDP multicast socket is created in receiver application which continuously receives multicast CDT. A timer of 200ms is running at receiver which probes the network interface continuously. If data is available on a given port it receives that data and displays in GUI. Given below in fig-04 is the flowchart of receiver application.

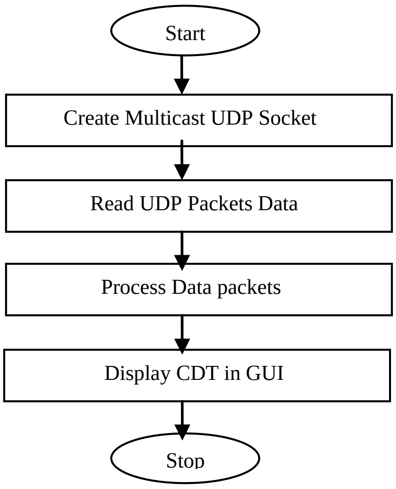

Fig 3: Flowchart of CDT Receiver application

The above steps after create UDP Socket are fired every 200ms. Receiver application receives the data in ASCII format from the network. If there is delay in the received CDT data then this delay has to be compensated at the client side. For doing compensation of delay first the ASCII characters are converted into integer and then apply the logic as.

Present_Sec=Presesent_Sec-1

In down mode and

Present_Sec=Present_Sec+1

When counting in Up mode. After doing this correction again these integral values are converted into the ASCII characters for displaying in GUI.

7. RESULTS AND ANALYSIS

The generated time was successfully transmitted into the network. All clients received data and successfully displayed them. It can be seen from the figure that the same time which is generated by generator application is received by the receiver application correctly. There were two issues namely (i) loss of data due to the nature of UDP protocol and (ii) delay occurred due to transmission to remote clients. Here the countermeasures for both problems are considered.

7.1 Repeated Transmissions

The Update rate of countdown time is 1sec i.e. new time packets are created at every one second. If there is a loss of packet the client application will not get that time packet and a jump will be observed by user. In order to compensate the loss of packet due to the unreliable nature of UDP the CDT generator sends same time 5 times. The redundancy at packet label has been implemented and using a QTimer API which fires at every 200ms and transmits the same CDT packets 5 times. Hence even if one or more packets are lost there are other packets to solve the purpose. This solution may be applicable to low bandwidth network where there is chances of congestion and hence a packet loss But in ITR Range the network is gigabit and hence chances of loss of packet is negligible.

7.2 Calculation of Delay

Let us take case of the client located at 400 km.

Propagation delay= distance/Speed of light [11]

$= (4000\text{x}1000)/ (3\text{x}10^8)$

$=0.013\text{sec}$

For a link of 2mbps leased line and the size of Ethernet frame to be 1518 bytes the transmission delay may be calculated as below

Transmission delay =packet length/link bandwidth

$=1518\text{x}8\text{bits}/2\text{x}10^6 \text{ bits/sec}$

$= 0.006072 \text{ sec}$

All other delay factors such as processing delay and queuing delay being of the order of microseconds as the network infrastructure are Gigabit network. If all these factors are combined the total delay is less than 500ms but update rate of timing is 1 seconds hence the user may not observer delay. If the stations are far located by satellite links then the delay may be noticeable, in this case the logic is implemented at client side at software level to compensate the delay.

This application has been tested during Range Trials and was found satisfactory with some suggested improvements regarding different types of outputs which can be used for interfacing it with launch Computer because the launch computer needs interface of MIL standard.

Shown below in fig-06 is the Conventional hardware based CDT Generator based on hardware along with software application based CDT generator in fig-04. The hardware based Time code readers which are RS232 interface based are shown in fig-0

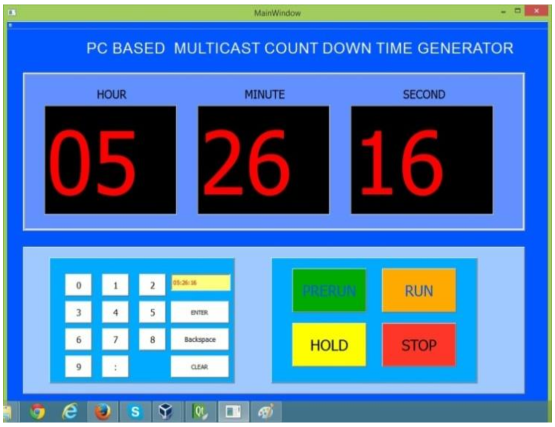

Fig 4: CDT Generator Application

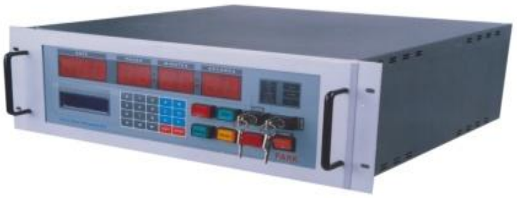

Fig 5: Hardware Based CDT Generator

The multicast CDT receiver client is shown in fig-05 below. Client has features like display status using led controls in Qt Creator. When there is hold ,the hold button in lit to red when time is counting down the down button is lit to green. In this way it provides a good feed back to user about reception of data as well as time status.

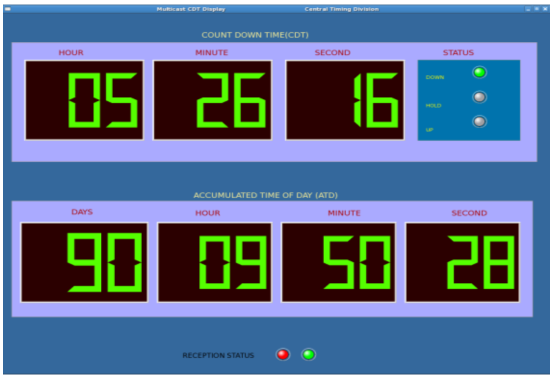

Fig 6: Multicast CDT Receiver

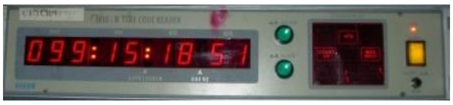

Fig 7: Hardware based CDT Receiver

8. CONCLUSION AND FUTURE WORK

The implantation of software based systems may introduce more processing delay than hardware based systems. This delay can be compensated using the logic in software. As in Range based Timing delay and time jumps are very critical issues these has to be taken care of very precisely otherwise this may give error in Rang measurement of flight vehicles. High speed Missiles travel several kilometres in one second and if there is a error of one second in timing then it will lead to error of several kilometres in Range. Here in this application both of these issues have been taken care of.

Hence it can be concluded that with appropriate knowledge of the frame work and available hardware one can convert any embedded devices into more flexible software applications. The PC based CDT generator application is such an example. The performance of software based CDT reader and generator was found satisfactory. It proved to be more useful than hardware based systems in terms of flexibilities it provides for compensation of time delay. After measurement of the delay from server to client the delay can be easily compensated at the client side. The processing time in software based implementation was found greater than the hardware based systems but that was compensated using software logic.

The generated Count down Time is sent over leased lines to various remote locations. These leased lines are maintained by the many public and private agencies. Hence there are always vulnerabilities of the CDT being intercepted by any intruder. Therefore in future it is being planned to encrypt the CDT packets using some Encryption algorithms so that it cannot be intercepted. The CDT generator application will generate the time and before sending it will encrypt the whole packet. After being received at client side there will be decryption algorithms running this will decrypt and display the received time. There are always challenges in development of Real Time encryption technology which will encompass minimal delay. Because delay become one of the deciding factors of any Range based Timing application to be implemented for Mission purpose.

9. REFRENCES

[1] Lin, John C., and Sanjoy Paul. "RMTP: A reliable multicast transport protocol." *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*. Vol. 3. IEEE, 1996.

[2] Castro, Miguel, et al. "SCRIBE: A large-scale and decen tralized application-level multicast infrastructure." *Selected Areas in Communications, IEEE Journal on* 20.8 (2002): 1489-1499.

[3] Szymaniak, Michal, Guillaume Pierre, and Maarten van Steen. "Versatile anycasting with mobile IPv6." Proceed ings of the 2nd international workshop on Advanced archi tectures and algorithms for internet delivery and applications. ACM, 2006.

[4] Walkowiak, Krzysztof. "Anycasting in connection-oriented computer networks: Models, algorithms and results." International Journal of Applied Mathematics and Computer Science 20.1 (2010): 207-220.

[5] Blanchette, Jasmin, and Mark Summerfield. *C++ GUI programming with Qt 4*. Prentice Hall Professional, 2006.

[6] Xue, Ming, and Changjun Zhu. "The socket programming and software design for communication based on client/server." *Circuits, Communications and Systems, 2009. PACCS'09. Pacific-Asia Conference on*. IEEE, 2009.

[7] .Law, KL Eddie, and Roy Leung. "A design and implement tation of active network socket programming." *Microprocessors and Microsystems* 27.5 (2003): 277-284.

[8] Stevens, W. Richard, Bill Fenner, and Andrew M. Rudoff. *UNIX network programming*. Vol. 1. Addison-Wesley Professional, 2004.

[9] Shneiderman, Ben, and Shneiderman Ben. *Designing the user interface*. Pearson Education India, 2003.

[10] Tang, Poi Loon, and Clarence W. De Silva. "Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control." *Control Systems Technology, IEEE Transactions on* 14.4 (2006): 707-718.

[11] Wu, Chengjie, et al. "Analysis of EDF scheduling for wire less sensor-actuator networks." Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of. IEEE, 2014.