Research and Implementation of Time Synchronization Mechanism in Simulation Architecture

Kunfu Wang¹, Wanfeng Mao¹ and Wei Feng¹

¹. System Engineering Research Institute of China State Shipbuilding Corporation, Beijing, China

Abstract. Modern warfare is changing from independent mode to cooperative mode, and time synchronization is the basis of cooperative operation. In this paper, the NTP based time service system (st simulation system) is designed and developed under the environment of windows QT 4.8.5 with the object-oriented thinking of C++, the time synchronization mechanism in simulation architecture is described, including the design of time synchronization algorithm of ST simulation system and the design of time service flow between ST simulation system and other simulation nodes in the architecture, it can effectively improve the time accuracy and accuracy in the simulation architecture.

1. Introduction

With the development of computer and communication information technology, the traditional combat mode has undergone fundamental changes: from an independent unit to a cooperative mode. The CEC (cooperative engagement capability) system is a model of cooperative defense operations. The system uses computer, communication data link, navigation and other information systems to combine their independent combat units to form a unified combat entity and achieve cooperative operations [1]. Time synchronization is the basis of cooperative operations. Traditional distributed time synchronization methods mainly include:

1) Set up special network to provide time[2] benchmark for each system;
2) Purchase special time receiving device;
3) Adjust the time manually.

The distributed system can not provide a unified global clock for the modules which are independent from each other. Usually, each process or module maintains its own local clock [3]. Therefore, the above methods will have different problems such as low efficiency, high cost and large error [4].

In this paper, the mechanism of time synchronization in simulation system is studied. ST simulator is designed to ensure the time synchronization of each simulation unit, and then related scientific research can be carried out in simulation system.

2. Related Research

2.1. Simulation System

The simulation simulation system mainly realizes the unified coordination of battlefield situation and combat state information by simulating the status data of each formation unit under different data links and the obtained situation data after information fusion processing during the execution of tasks by the simulation command node, and then carries out relevant verification work [5]. The composition of the simulation system is as follows:
The system is composed of Command node, Wireless communication link, UAV node, surface ship node, AWACS node, fixed wing aircraft node, submarine node, other external information processing system and director station. Through network switch, a distributed simulation system environment is formed. According to the characteristics of each communication data link, the interface protocols of different data links are defined. UDP / IP is used as the underlying network communication protocol to carry out the information exchange between different node units.

2.2. Time Synchronization

At present, the network timing in the simulation architecture is based on the Internet, which provides standard time information for other terminal devices in the Internet.

The network time synchronization provided by Windows system synchronizes to a specific external network server by changing the settings in the Internet time in the system date and time configuration, and there can be multiple free time servers. When a server is not available, it can automatically switch to other time servers. But the time is only accurate to the second level, and the client must always be connected to the Internet. For example: time.windows.com.

Through the time server of GPS or Beidou satellite navigation system, synchronization with the atomic clock of GPS or Beidou satellite navigation system is adopted. Its typical application is centralized synchronization, which synchronizes with the only time server.

Common network time service protocols include Daytime protocol, Time protocol, NTP protocol and PTP protocol. Among them, NTP is mainly used to synchronize the computer clock in the Internet. By specifying several clock source sites on the Internet, we can provide time service for users’ computers, and these clock source sites can be compared with each other to improve the accuracy.

3. Research on Time Synchronization in Simulation Architecture

3.1. Research on Time Synchronization in Simulation System

NTP protocol is usually described as a client server mode. UDP / IP protocol is used for sending and receiving time stamps, and 123 is used for communication port by default [6]. According to the requirement of time synchronization in the simulation architecture, this paper proposes the time system simulation system based on NTP.

The working principle of St simulation system is a two-level server and client system. It maintains the punctuality process in the server and synchronizes the time for each client through NTP protocol. The principle of synchronization algorithm[7][8] is shown in the following diagram:
Figure 2. ST synchronization algorithm schematic diagram

T1 is the time when ST client sends request message, T2 is the time when ST server receives request message, T3 is the time when server sends response message, T4 is the time when client receives response message, Set $\delta'$ as the path delay from sending message from time system client to receiving message from time system server, $\delta''$ as the path delay from sending message from time system server to receiving message from time system client, Delay$\delta_i$ as the total path delay, Offset$\theta_i$ as the clock offset estimation of server and client, then:

$$T2 - T1 = \theta_i + \delta'$$

$$T4 - T3 = \delta'' - \theta_i$$

$$\delta_i = \delta' + \delta''$$

If we assume that the message transmission path delay from the client to the server is equal to the message transmission path delay from the server to the client, that is, $\delta' = \delta'' = \frac{\delta_i}{2}$, the above three equations can be converted into:

$$T2 - T1 = \theta_i + \frac{\delta_i}{2}$$

$$T4 - T3 = \frac{\delta_i}{2} - \theta_i$$

It can be concluded that:

$$\text{Delay} \delta_i = (T4 - T1) - (T3 - T2)$$

$$\text{Offset} \theta_i = \frac{(T2 - T1) - (T3 - T4)}{2}$$

Due to the instability of the network environment and the instability of the network delay, the accurate offset $\theta_0$ of the clock of the server and the client falls in the interval $[\theta_i - \frac{\delta_i}{2}, \theta_i + \frac{\delta_i}{2}]$. For the estimation of a large number of samples, according to the minimum delay criterion specified by the network protocol, the offset with the minimum network transmission delay is selected from the sample data, namely: $\theta_o = \theta_i + \frac{1}{n} \min_{i=1}^{n} (\delta_i)$, the sample number N is the sliding window statistics.

3.2. Design and Implementation of ST Simulation System in Simulation System

The design of ST server, client and other simulation system timing process based on the above principle is shown in Figure 3:
ST simulator is divided into two levels of server and client mode, which uses UDP / IP as the transmission protocol[9][10].

![Workflow design of time series simulation system](image)

**Figure 3.** Workflow design of time series simulation system

The main function of ST server is to distribute time information to the specified clients in the network; the main function of ST client is to request time service from ST server; receive synchronous time signal, determine clock deviation of the machine; calibrate the clock of the machine; send time system message to other simulators, so as to achieve the time unity of each node unit in the simulation system.

### 3.3. Key Code Design of St Simulation System

This design is designed and implemented in the operating system: Windows 7, programming environment: Visual Studio 2010, programming language: QT 4.8.5. The flow chart structure of the program design is shown in the following figure 4:

Some codes are designed as follows: time setting structure:
typedef struct tagTimeStruct
{
  struct second
  {
    unsigned char one;
    unsigned char ten;
  } sec;
  struct minute
  {
    unsigned char one;
    unsigned char ten;
  } min;
  struct hour
  {
    unsigned char one;
    unsigned char ten;
  } hou;
  struct zoneModify
  {
    unsigned char one;
    unsigned char ten;
  } zone;
  struct day
  {
    unsigned char one;
    unsigned char ten;
  } da;
  struct month
  {
    unsigned char one;
    unsigned char ten;
  } mon;
  struct yearL
  {
    unsigned char one;
    unsigned char ten;
  } yL;
  struct yearH
  {
    unsigned char hundred;
    unsigned char thousand;
  } yH;
} TimeStruct;

The core base class of NTP time synchronization code CNtpBase:

class CNtpBase
{
  public:
    CNtpBase();
    virtual ~CNtpBase();
    int ConstructPacket(char * packet);
    virtual void CreateSocket() = 0;
    virtual void BindServAddr(struct sockaddr_in &h, const char s[]);
    //Bind socket
    virtual int getNtpTime(int sk, struct sockaddr_in

Figure 4. Time series design flow chart
* addr, struct NtpPacket * ret_time);//Get time server time
virtual int setLocalTime(struct NtpPacket * pnew_time_packet); //Correct local time
};
int CntpBase::constructPacket(char *packet)
{
    struct NtpPacket *ntp;
    ntp = (struct NtpPacket *)packet;  //Header part of assignment datagram
    ntp->leapVerMode = 35;
    ntp->stratum = STRATUM;
    ntp->poll = POLL;
    ntp->precision = PRECISION;       //First time stamp
    struct timeval tv;
    gettimeofday(&tv, NULL);
    ntp->referenceTimestamp.integer=htonl(tv.tv_sec+JAN_1970);
    ntp->referenceTimestamp.fraction= htonl(USEC2FRAC(tv.tv_usec));
    return 0;
}

Where, gettimeofday (& TV, null) function gets the exact time of the current system, whose value
is the number of seconds from 0:00 on January 1, 1970 to the current time. The NTP time stamp
obtained from GPS is the number of seconds from 0:00 on January 1, 1900 to the current time, so it
is necessary to add the number of seconds returned by the system to the number of seconds in 70
years.

#if def WIN32
int gettimeofday(struct timeval *tp, void *tzp)
{
    time_t clock;
    struct tm tm;  //The structure of time, minute and second
    SYSTEMTIME wtm; //Structure of system time
    GetLocalTime(&wtm); //Get system time
    //Time required to convert system time into NTP message
    tm.tm_year = wtm.wYear - 1900;
    tm.tm_mon = wtm.wMonth - 1;
    tm.tm_mday = wtm.wDay;
    tm.tm_hour = wtm.wHour;
    tm.tm_min = wtm.wMinute;
    tm.tm_sec = wtm.wSecond;
    tm.tm_isdst = -1;
    clock = mktime(&tm);
    tp->tv_sec = (long)clock;
    tp->tv_usec = wtm.wMilliseconds*1000;
    return 0;
}
#endif

SYSTEMTIME sysTime;//Convert seconds of NTP message format into system time
sysTime.wYear = 1900 + tv1.tm_year;
sysTime.wMonth = 1 + tv1.tm_mon;
sysTime.wDay = tv1.tm_mday;
sysTime.wHour = tv1.tm_hour;
sysTime.wMinute = tv1.tm_min;
sysTime.wSecond = tv1.tm_sec;
sysTime.wDayOfWeek = tv1.tm_wday;
sysTime.wMilliseconds=(WORD)newtv.tv_usec/1000;

So far, the time of the system has been corrected, and an interface function is provided to realize the time synchronization function of the whole network. After the IP address and time synchronization interval of the active and standby servers are configured, the interface function can be called to calibrate the time of the system where the program is located.

4. Conclusion
According to the importance of time unification in simulation architecture, this paper proposes a simulation system based on NTP protocol. With the object-oriented thinking of C++, the time service system (st simulation system) based on NTP protocol is designed and developed in the environment of windows QT 4.8.5. The application proves that it can effectively improve the time accuracy and accuracy in simulation architecture.

5. Reference
[1] Liu Peng, Chen Ying, Guo Xiao-li. A high precision time synchronization system design for naval formation[J].Ship Science and Technology, 2011.6:46-50,65
[2] Mills, D., D.Plonka and J.Montgomery. Simple network time protocol(SNTP)version 4 forIPv4, IPv6 and OSI.Network Working Group Report RFC-4330,University of Delaware, December 2005.27 pp.
[3] Yuan-Xi Yang, Yang-Yi Xu, Jinlong LI et al. Progress and performance evaluation of BeiDou global navigation satellite system: Data analysis based on BDS-3 demonstration system[J]. Science China Earth Sciences, 2018, 61: 614.
[4] Mitchell W S. Overdick, Joseph E. Canfield, Andrew G. Klein. A software-defined radio implementation of timestamp-free network synchronization[C]. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017: 1193-1197.
[5] Rakesh Kumar, Monowar Hasan, Smruti Padhy et al. End-to-end network delay guarantees for real-time systems using SDN[C]. 2017 IEEE Real-Time Systems Symposium (RTSS), 2017:231-242.
[6] Zhang Xin, Aliaosha Ye, Liu Xuan, et al. Research on Site Diagnosis Technology of Electric Energy Data Acquire System[J] Electrical Measurement & Instrumentation, 2014, 51( 8) : 116-119.
[7] LIU Chuan-bo, WANG Hang-yu, QIU Zhi-ming. Research on time synchronization protocol of cooperative engagement simulation[J]. Computer Engineering and Applications, 2010.46(23): 223-226
[8] Liu Lei, Zhang Lei. Optimization and simulation of improved NTP protocol in asynchronous communication network[J].Computer Engineering and Applications,2011,47(6):75-77,92
[9] Wang Wenjun, GongYuchang, ZhuJianming, A Time Synchronization Protocol in Local Network Based on UDP[J].Computer Applications and Software, 2007.5:133-135
[10] Benjamin Hesmans, Olivier Bonaventure. An enhanced socket API for multipath TCP[C]. Proceedings of the 2016 Applied Networking Research Workshop, 2016:1-6.