

# The Design and Implementation of Bus LAN Simulation System Based on Object-Oriented Technology

Meijuan Kang

School of Informatics, Linyi Normal University, Linyi 276000, China

E-mail: kmj\_2005@163.com

## Abstract

Computer simulation is one of methods used in evaluating the performance of LAN. In order to decrease network response time, the simulation system of P-persistent CSMA/CD Bus LAN is designed and implemented using Visual C++, by using the object-oriented analysis. Finally, the effect of the send probability for the network performance has been analyzed.

**Keywords:** Bus LAN, Computer simulation, Object-oriented

## 1. Introduction

The computer simulation of LAN is simulating the dynamic work process of LAN by running computer program. Through it we can obtain the characteristic parameters of network work, analyze the effect of relevant factors for the network performance, seek the strategies that can make equipments produce the best possible results and realize the best control of the amount of LAN information.

The system simulation subject is time slot P-persistent CSMA/CD Bus LAN. In order to analyze the effect of the send probability for the network channel utilization, the Bus LAN simulation system is designed and implemented using Visual C++ 6.0. Because Visual C++ 6.0 has visual and object-oriented programming techniques and it is powerful, user-friendly and easy to use, it is used in the design.

## 2. Simulation System Design

### 2.1 Simulation System Introduction

The slot P-persistent CSMA/CD Bus LAN mainly consists of LAN, workstation, message, message queues and event table. The LAN is composed of a number of workstations connected by the bus and it has a simulation clock that is used to control the propelling of the simulation process. The message is send by workstation and is stored in send message queue. The LAN is in charge of the arriving, distribution and receiving of the message. The message inter-arrival times is distributed as Poisson distribution. The send probability is distributed as the normal distribution. The message length is distributed as Exponential distribution. The received messages are stored in leaving message queue in order to calculate the average waiting time of message finally. All events of the LAN are stored in event table.

The event scheduling strategy is used in the simulation system. There are three kinds of events such as message arriving event, sending message event and receiving message event. Each event has its time of occurrence. When it is carried out, it can give rise to changes in system state or trigger other events. As the simulation clock forwards, the system should find the earliest event in event table and call the corresponding event handler. According to the sequence of the events, the simulations finish when all events in event table have been handled one by one.

### 2.2 Simulation system modeling

The simulation model is built based on object-oriented technology. On one hand the object-oriented technology can help us build a model that is very similar to the entity; on the other hand its class and its inheritance technology can improve the modularity, reusability and flexibility of the simulation software.

In the simulation system the message class, the workstation class, the simulation manage class, the Bus LAN class, the event class, the class of message arriving event, the class of send message event and the class of receiving message event are defined through object-oriented analysis. The class of message arriving event, the class of send message event and the class of receiving message event are subclasses of the event class that is an abstract base class and inherit its attributes and methods.

The object model is shown in Figure 1.

### 2.3 The Realization of Simulation Algorithm

The Simulation Algorithm using VC++ is as follows.

```
SetSimulateParameter(); //simulation initialization
myBusLAN.BusTag=true;//set bus free
SimulationTime=0;//set initial simulation time
CMyEvent* p=new CCreatMessage(0); // create the initial event
myBusLAN.myEventList.AddTail(p); //append the initial event to the event table
p->handle(); // execute event handler
delete myBusLAN.myEventList.RemoveHead();//delete the initial event from the event table
while (!myBusLAN.myEventList.IsEmpty()) //while the event table is not empty
{
    p=FindEarliestEvent();//find the earliest event
    pos=Position(p);//the position of the earliest event in the event table
    SimulationTime= p->HappenTime;//the simulation clock forwards
    myBusLAN.myEventList.RemoveAt(pos);//delete the earliest event from the event table
    p->handle(); // execute event handler
}
CalculatePerformanceParameter();//calculate simulation result
```

### 3. Simulation Result Analysis

In order to analyze the effect of the send probability for the collisions and the average waiting time, in 10 seconds, the simulations at different load have been carried out, such as reaching 5, 10, 20, 30, 40, 50 and 60 messages per second.

From figure 2 we can see the effect of the send probability for the collisions:

- (1) When the send probability is constant, the number of conflicts increases as the load increases.
- (2) When the load is light, the effect of the send probability for the number of conflicts is small.
- (3) When the load is heavy, the number of conflicts increases rapidly if the send probability is big.

From figure 3 we can see the effect of the send probability for the average waiting time:

- (1) When the send probability is constant and within a certain range, the average waiting time increases as the load increases.
- (2) When the load is light, the average waiting time increases rapidly if the send probability is small.

According to the simulation results, we can draw a conclusion: The effect of the send probability for both the collisions and the average waiting time must be considered in order to improve the network performance; when the load is certain, the send probability should be correctly chosen in order to make collisions as few as possible and make the average waiting time not out of range.

### 4. Conclusion

The simulation software is not only comprehensive in functionality but also easy to operate. It can display the simulation process and results and it has better robustness and reusability.

### References

- Chen, Jiahai. (1995). *Writing Visual C++ Programming (Third) - Class Library Using Reference*. Beijing: Beijing Science Press. P.76-148.
- Gu, Qitai. (1999). *Modeling and Simulation of Discrete Event System*. Beijing: Tsinghua University Press. P.46-87.
- Li, Layuan & Li, Chunlin. (2001). *Computer Network Technology*. Beijing: National Defense Industry Press. P.113-152.
- Lu, Zhaoyi & Wang, Siming. (1997). *Theory of Amount of Information of Computer Communication Network*.

Beijing: Electronics Industry Press. P.64-93.

Yang, Debao. (1994). *Engineering Probability and Statistics*. Beijing: Beijing Institute of Technology Press. P.83-121.

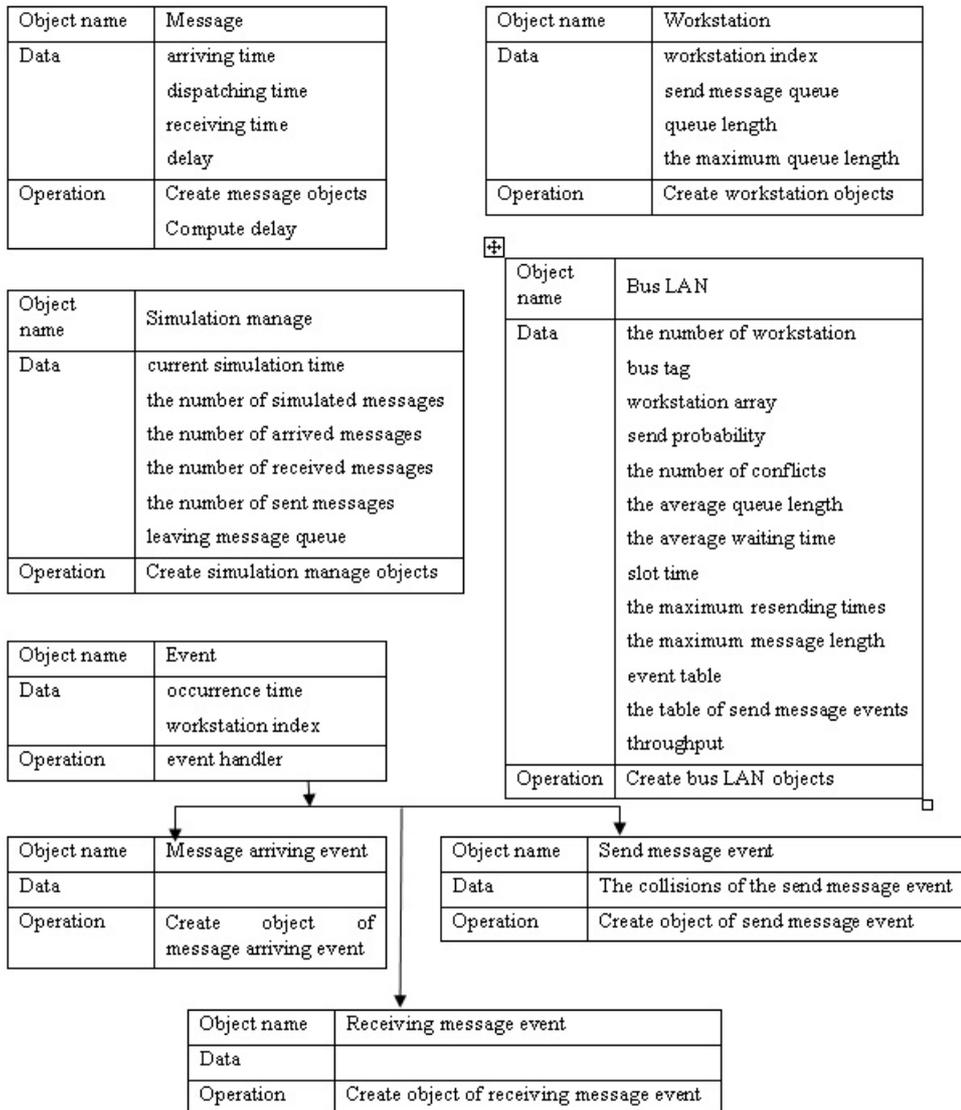


Figure 1. The object model

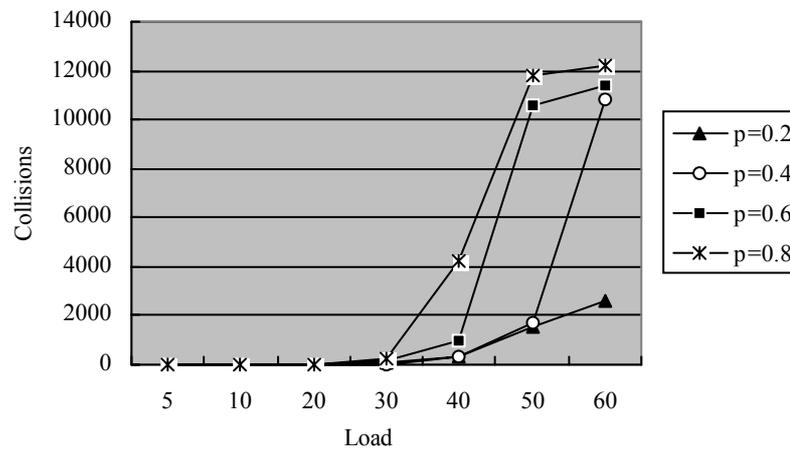


Figure 2. The effect of the send probability for the collisions

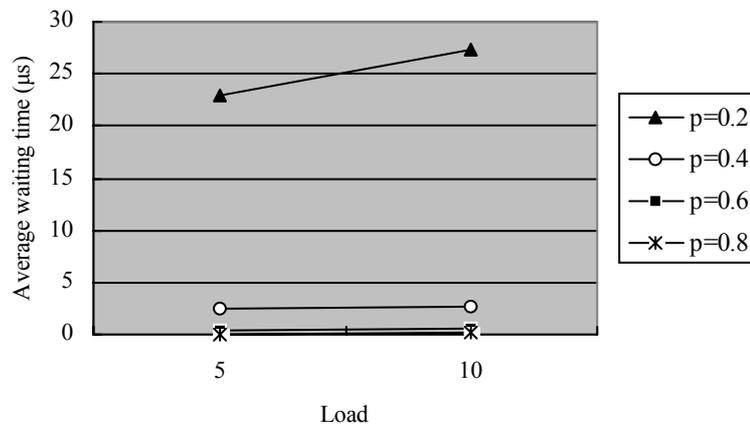
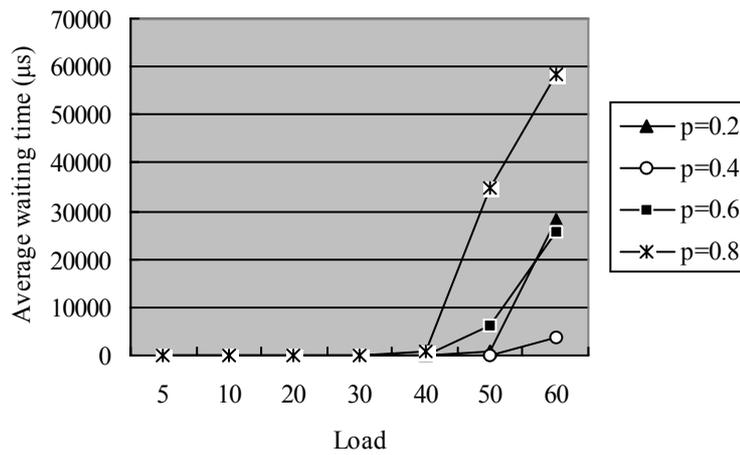


Figure 3. The effect of the send probability for the average waiting time