Journal of Southeast University (English Edition) Vol. 23, No. 2, pp. 244 – 249 June 2007 ISSN 1003—7985

Network-based products detection system: congestion control and self diagnosis

Yi Qiaomu[1] Cheng Jinshu[1] Fang Hongchao[2] Chen Youping[2]

([1]School of Materials Science and Engineering, Wuhan University of Technology, Wuhan 430070, China)
([2]School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: The problems that arise while developing a real-time distributed information-processing software system are studied. And based on the TCP/IP protocols and socket, for its facility in client/server (C/S) model networking programming, a prototype is designed for data transmission between the server and clients and it is applied on an on-line products automatic detection system. The probability analysis on network congestion was also made. A proper mechanism based on the ARCC (adapted RTT congestion control) algorithm is employed for detecting and resolving congestion, the purpose of which is mainly to achieve congestion avoidance under the particular conditions in this network-based system and reach the desired performance. Furthermore, a method is proposed for a client to diagnose automatically the connection status between the server and the client and to re-connect to the server when the disconnection is detected.

Key words: distributed system; network communication; network congestion; self-diagnosis

The intercommunication of information in networks as a rapidly developing technology has far-ranging utilizations in modern industrial production. As to a large-scale on-line product detection system, it is essential and indispensable to achieve real-time communications between two different terminals under the networking conditions; i. e., the system should have capabilities to respond and to process the data quickly enough and then return the results to its destination in real time. So we adopt a distributed system implemented synchronized strategy and by this means we can guarantee its synchronization and rapidity when running the program[1-2].

Distributed systems have many applications, including telecommunications, distributed information processing, scientific computation and real-time process control. Two properties are essential for distributed systems[3]: ① Computation activity is represented as the concurrent execution of sequential processes; ② The processes communicate by passing messages.

The computation models generally considered to be distributed are process models, in which computational activity is represented as the concurrent execution of sequential processes. The process models that are most obviously distributed are ones in which the processes communicate by message passing: one process sends a message by adding it to a message queue, and another process receives the message, by removing it from the queue. These models vary in such details as the length of the message queues and how long a delay may last between the moment the message is sent and when it is received.

1 Distributed Network-Oriented System

There are two kinds of analyses used for the analysis of distributed systems: behavior analysis and performance analysis. All the possible trajectories are analyzed during the behavior analysis and this allows checking the correctness of a specification. Various validation and verification methods are used for correctness analysis. During performance analysis the developed specification of distributed systems is executed by a computer. Performance analysis is carried out by means of simulation.

Various formal methods are used for specification and analysis of distributed systems[4-5]. This paper focuses on the performance analysis which permits us to prove that the developed specification is correct and to evaluate the performance characteristics of the analyzed system. Fig. 1 presents the topological diagram of the network-based float glass on-line detection system. It basically consists of one server, eight clients and one workgroup exchanger. Besides, eight CCD cameras connect to eight clients and transmit image data acquired by a frame grabber to each client respectively; the server receives signals emitted by two sensors fixed

Received 2006-11-01.
Foundation item: The National Natural Science Foundation of China (No. 60474021).
Biographies: Yi Qiaomu (1962—), male, graduate; Cheng Jinshu (corresponding author), male, professor, chengjinshu@whut.edu.cn.

above the glass, one is for its crack detection, another is to manipulate labeling equipment (ink-jet marker) to achieve the glass classification.

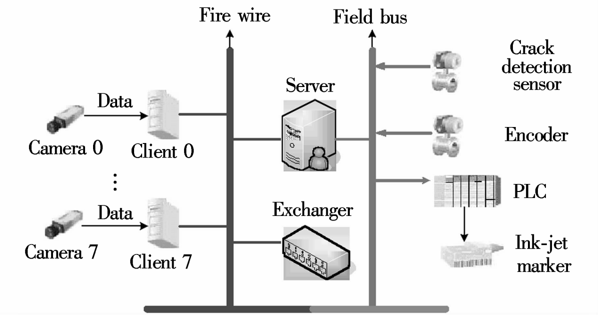

Fig. 1 Network topological diagram

In the distributed network-oriented system, basically, the clients mainly aim at analyzing and processing obtained image data in real time, locating the position of each fault accurately, and classifying the glass faults in the first round to conform to its industrial standards. As far as the server terminal is concerned, the main task is to reclassify the glass faults by a pattern recognition based on a fuzzy neutral network matching strategy which refers to the glass faults template database. Besides, the server indicator needs to demonstrate integrated diagrams as a faults distribution map and the purpose of the diopter tendency chart is to establish an applicable interactive man-machine interface. Furthermore, the server should be capable of receiving the detection signals sent from external sensors and encoders the purpose of which is to perform crack detection and subsequent classifications. And in Fig. 1 we can enumerate three relatively independent data transmission routines: ① Image data from each client to the server through a fire wire; ② The detection data from sensor or decoder to server through a field bus; ③ Controlling messages from the server to the programmable logical controller and the ink-jet marker as well.

In response to their respective tasks we define basic implementations of this network-oriented system, i. e., achievement of the transmission of glass fault contained image data and fault category information (including bubbles, stones, strias, scratches, optical distortions, etc.) from client terminals to the server terminal in real time; achievement of the transmission of the image acquisition parameter settings (dimensions of a image frame) and the manipulation signals (data archiving) from the server terminal to the client terminals.

In this network-oriented system we initially employ the C/S model in the socket to establish the communication between each client and the server. The design and the implementation of the glass faults on-line detection that we are developing depend on inputs from clients, and every single intercommunication should be launched by the client, that means the server is always in idle in order to ensure that the server can respond to the requests from clients in real-time.

2 Network Intercommunication

In the float glass fault on-line detection system, each client needs to send the connection requests including their names, IP addresses and port information while the server must be in the waiting status and ready to respond by a monitoring function. Then the server continues the identity validation in order to perform the permission to the connecting requests and simultaneously send back the acknowledgement message to the client, or eliminate the request if it is in invalidated.

Once the connection between the server and the client is established through socket protocol, data was transmitted in packet units for the concise structure which is composed of a packet header and parameters in which the header labels the packet while parameters contain the host station detailed commands to the slave stations.

The base protocol employed in this networking system is the connection-oriented TCP protocol (represented as a socket stream CSocket class in programming). That means there is no theoretical limitation on the packet length. The receiver can receive the data stream in proper order and classify packets by analyzing each header and then obtain the lengths and contents. No CRC check on these packets for TCP protocol guarantees a sequential and error-free transmission.

In the client process, the achievement of data transmission first requires a CCommSocket class derived from a socket base class to create a new Socket in the client terminal. The client is allowed to connect to the server once the program has started. After the connection is established, data communication can be established by calling the previously created Socket. Also the packaging format of the sent data is of great significance for the detected data including their types, amount of the detected faults and their dimensions and locations as well. Here respective proper data types must be employed by which these data can be labeled by characterized packet headers.

In the server process, initially we need to create a CListeningSocket class and a CClientSocket class. The former is used for monitoring the connection requests from the client, the latter is used for creating a new socket corresponding to each distinguished requests. The application of list structure carries out the scheduling management of multiple client connections so as to avoid, or, at some level, to reduce possible occurrences of transmission confusion. Furthermore, in order for each client to send data to the server in real time, the server must be able to validate sources (client label, IP address and port) of the received data and to identify its data type through parsing.

3 Network Congestion Control

Congestion control takes magnificent consideration to the networking environment between terminal nodes in order to avoid overloading beyond its potential capacity[6-7]. Mostly, a congestion control algorithm consists of two different strategies: congestion avoidance and congestion control. Congestion avoidance is a precautionary measure which aims at keeping the network operating in a high throughput and low delay condition, while congestion control is a recovery measure which can recover the network from congestion.

Congestion is caused by the disequilibrium of network resources and the flow distributions which cannot be eliminated with the upgrade of the network's processing capacity. In our network-based system, it is hypothesized that the network's maximum processing capacity remains constant and is based on the RTT (round-trip time) fluctuation analysis. We adopt a technique which is an adapted conventional TCP congestion control algorithm, namely, ARCC (adapted RTT congestion control algorithm). In ARCC, we induce the predictive RTT into Tahoe TCP[7] to predict the network congestion ranks so as to access the fitting window parameters in congestion control procedure.

The ARCC algorithm can be described as follows:

① After receiving an ACK for new data,
  If L (congestion ranks) $\geqslant 3$
    If (cwnd < threshold)
      cwnd = cwnd * 2;
    Else cwnd = cwnd * a;
  Else
    If (cwnd < threshold)
      cwnd = cwnd * 2;
    Else cwnd = cwnd + (1/[cwnd]).

② If receive the repeated ACK 3 times (before timeout)
    cwnd = cwnd * a;
    threshold = cwnd * a.

③ If no ACK before timeout
    threshold = cwnd * a * a;
    cwnd = 4.

In the description above a is a scale coefficient and in classical TCP algorithms (such as Reno TCP) it is a constant equal to 1/2, which means that when receiving repeated ACKs, cwnd may decrease to half of its current size.

In the ARCC algorithm, the factor a can be dynamically regulated according to the determination of the network congestion severity. The principles used for selecting a are shown in Tab. 1 and its optional value can be chosen to conform to certain conditions.

Tab. 1 Congestion ranks and a values

Congestion ranks L	1	2	3	4
Range A	[0, 0.25]	[0.25, 0.5]	[0.5, 0.75]	[0.75, 1]
Value a	1/2	1/4	1/8	1/16

Compared with Tahoe TCP, ARCC is the conservative congestion control algorithm which can forecast the probability of the congestion before its occurrence through RTT, then manipulate it to reduce the transmission speed in advance which may avoid congestion exacerbation. In addition, in Tahoe TCP the scale coefficient a is a constant while in this new adapted algorithm a dynamic policy is adopted in order to choose the a value with alternated congestion conditions. When operating in real network circumstances, the algorithm has not altered the TCP packet's data standard. It only needs the essential information as Tahoe TCP required and all the changes happen in its protocol software codes.

4 Application and Related Analysis

4.1 On-line diagnosis

As illustrated in section 2, the real-time system based on distributed network-oriented mechanisms has successfully been used on the production line[8] in Wuhan Changli Glass Co., Ltd. Regarding the on-line detection system, we still have to pay more attention to the case in which the server cannot receive the detection data in real time, under this situation the system cannot catch any exceptions and subsequent operations such as detections and quality classification may fail correlatively. There are dozens of reasons. A possible one is that other components can operate in a good condition regarding fault detection in a client; however, diagrams represented on the server screen can only indicate the previous detection results; i.e., there is no effective socket connection between client and server by which the detection data in a client cannot be transmitted to the server in real time. Even worse, after this delay, these data may transmit in a over-loaded cluster way. This includes the detection data, both previous and current. This may definitely result in misrepresentations and impose unpredictable contributions to successive operations. To detect whether there is a safe socket connection between the client and the server, the following are two possible approaches:

① Set a timer in the server process and send a testing message to each client in fixed intervals. If the client can send back the feedback, then we can say there is a safe connection; otherwise, the server may keep resending the testing message. And if the server can receive the acknowledgement within n intervals (set by user, here $n = 5$) we can still consider them to be in connection. If this range exceeded, the connection can be considered to be lost.

② Set different timers in each client process respectively and send a testing message to the server at fixed intervals. If the server can send back the acknowledgement message, then we can say there is a safe connection; otherwise, the client may keep resending the testing message. And if the client receives the acknowledgement from the server within n intervals we still believe that they are connected. If this range exceeded, the connection can be considered to be lost.

Comparing these two methods, it is proved that the second one is preferable. Because when a socket connection is established between the server and the client, these two terminals can only play the distinguished role in which the client sends out the connection request while the server is in a passive mode of listening and waiting. In addition, when disconnections are detected in the server, it cannot transmit the message to the corresponding client and thus this client cannot be triggered to send a reconnection request to the server. So in the network-based system we employ the second approach to achieve the automatic diagnosis of connection status in the client terminal, and the client can launch socket reconnection to the server once a disconnection is detected. The flow diagram of the self-diagnosis implementation is represented in Fig. 2.

4.2 Performance of ARCC algorithm in congestion control

Here we have to evaluate the performance of the data transmission within this real-time detection system, and to evaluate the ARCC algorithm. The running system processes a program to achieve the ARCC algorithm, and the results make a comparison between the Tahoe congestion control algorithm and the ARCC in two aspects: the congestion windows and their throughput.

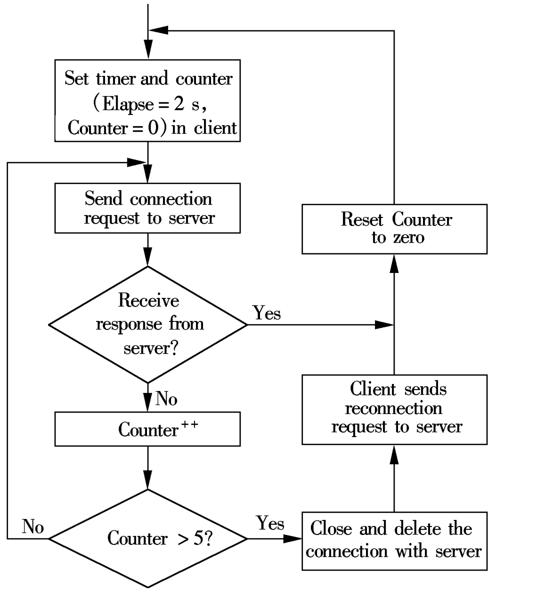

Fig. 2 Self-diagnosis implementation

The network topological diagram of the real-time system can be shortly described as shown in Fig. 3. In the diagram, S1, …, S8 represent 8 clients as nodes of the front sending source, R1 represents the exchanger as the node of the middle router, and D1 represents the server as the node of the receiving destination. The dimension of the receiver window (rwnd) is set to 50 (package amount). Data from S1 to S8 transmits to D1 through R1. We adopt the Ethernet here and there is delay between S1, …, S8 and R1 and between R1 and D1, though we assume that the previous delay is 5 ms, and the latter one is 100 ms. Though it is called the kilomega Ethernet, eventually the link bandwidth is the same, only approximately 10 MB. So it is presumed that there may be congestion between R1 and D1.

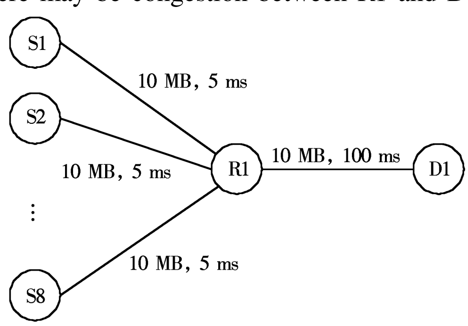

Fig. 3 Topological diagram

Fig. 4 shows that the comparison of the congestion window (cwnd) of the two algorithms. We can learn from this figure that the fluctuation of the cwnd based on the ARCC algorithm is rather mild; also there is a lower tendency of congestion occurrence and it will not lead to time-out retransmission or dropout retransmission. While under the effect of the Tahoe TCP congestion control (TTCC) algorithm, there is continuously occurrence of congestion with two kinds of retransmission mentioned-above.

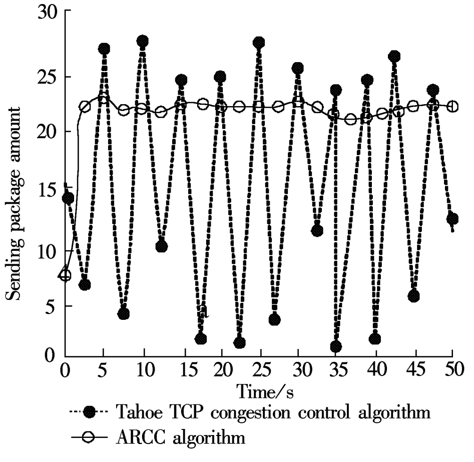

Fig. 4 Comparison on cwnd in TTCC algorithm (Running time is 50 s, rwnd is set to 50 (package amount))

The comparison on throughput, based on these two congestion control algorithms, is shown in Fig. 5. This figure indicates that the throughput based on the ARCC algorithm is somewhat lower than that of its counterpart. This particular state can be accredited to the ARCC algorithm reducing the sending rates in order to achieve avoidance of network congestion, but this does not bring out the negative impacts on the algorithm's applicable effectiveness. Furthermore, the ARCC algorithm reserves certain portion of the network resources, with the relative graduality of the cwnd fluctuation, it achieves a higher transmission efficiency.

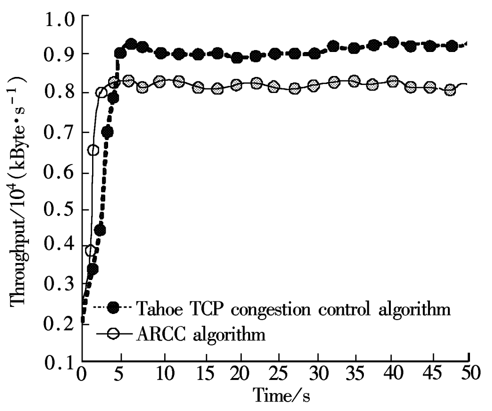

Fig. 5 Comparison of throughput in TTCC algorithm and ARCC algorithm

5 Conclusion

The machine-vision-based on-line products detection system leads to a new trend in automatic industrial production detections, and one of the key technologies to its full application is the implementation of data communication in stability, reliability and efficiency. The network-based system designed in this paper adopted the standard TCP/IP protocols, applied the Windows Socket programming techniques to the communication between the client and the server, and the socket programming procedure has also been given. All the components discussed above provide a scenario for the prototype design of a network-based distributed system. In the distributed network, the distributed processing technology allows the system to achieve satisfactory performance. The float glass fault on-line detection system based on the above-mentioned communication program has been demonstrated on float glass production line Ⅱ in Wuhan Changli Glass Co., Ltd with fairly good performance.

Since there are totally eight clients in this network-based system and each of them needs to achieve data communication through the Ethernet with server, respectively, discontinuity of the detection data may cause network congestion at some level in the transmission, and this has been solved properly and under control through the ARCC algorithm. Compared with Tahoe TCP, to a significant degree ARCC aims at the implementation of congestion avoidance. We also can learn that the network throughput in ARCC is lower than in Tahoe TCP since the ARCC algorithm slows down the transmission speed and achieves the distribution of network resources conservatively. That is the reason why the variation of the cwnd in ARCC is relatively smooth while the congestion occurs at a higher rate in Tahoe TCP and thus requires timeout retransmission and dropout retransmission.

Besides, in the system there is an innovative approach for a client to automatically diagnose the connection status between server and client and to re-connect the server when the disconnection is detected. This method may enable users to reestablish the connection between server and client without restarting the program.

References

[1] Mills D L. Internet time synchronization: the network time protocol [J]. *IEEE Trans on Communications*, 1991, **39** (10): 1482 - 1493.

[2] Ramanthan P, Kandlur D D, Shin K G. Hardware-assisted software clock synchronization for homogenous distributed systems [J]. *IEEE Trans on Computers*, 1990, **39**(4): 514 - 524.

[3] Henrikas Pranevicius. Formal specification and analysis of

distributed systems [J]. *Journal of Intelligent Manufacturing,* 1998, **9**(6): 559 − 569.

[4] Xie Anming, Shao Meizhen. The design of the network communication program with socket [J]. *Journal of Information Engineering University*, 2001, **2**(1): 68 − 70. (in Chinese)

[5] Dong Yu. Design of network communication program on winsock [J]. *Journal of Lanzhou Railway University: Natural Science Edition*, 2001, **20**(1): 54 − 57. (in Chinese)

[6] Giuseppe Di Fatta, Giuseppe Lo Re, Alfonso Urso. A fuzzy approach for the network congestion problem [C]//*International Conference of Computational Science*. Amsterdam, 2002: 286 − 295.

[7] Yong Ho Song, Timothy Mark Pinkston. Distributed resolution of network congestion and potential deadlock using reservation-based scheduling [J]. *IEEE Trans on Parallel and Distributed Systems*, 2005, **16**(8): 686 − 701.

[8] Fang Hongchao, Peng Xiangqian, Yu Wenyong, et al. Online products detection system based on network [J]. *Computer Application Research*, 2006, **23**(11): 160 − 163. (in Chinese)

基于网络的产品检测系统:拥塞控制和自诊断

易乔木[1]　程金树[1]　方宏超[2]　陈幼平[2]

([1] 武汉理工大学材料科学与工程学院,武汉 430070)

([2] 华中科技大学机械科学与工程学院,武汉 430074)

摘要:研究了一种实时分布式信息处理系统软件开发中的关键技术问题. 基于 TCP/IP 协议及 Socket 技术,利用 C/S 模型的网络编程技术,设计实现了用于在线产品自动检测系统的服务器与客服机之间的数据传输原型系统. 对网络拥塞进行了可能性分析,提出了一种基于 ARCC 算法的有效控制机制,并应用于基于网络的实时检测系统中,以解决在特定工况下出现拥塞时实现拥塞避免,达到了预期的效果. 而且,还提出了一种自动诊断客户机状态的方法,以方便地检测服务器与客户机的连接状态,当连接中断时可以自动再连接.

关键词:分布式系统;网络通讯;网络拥塞;自诊断

中图分类号:TP393