Proceedings of APPT '95

International Workshop on
Advanced Parallel Processing Technologies
September 26–27, 1995, NJU, Beijing, China

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Proceedings of APPT'95

International Workshop on

Advanced Parallel Processing Technologies

September 26—27, 1995, NJU, Beijing, China

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

The Proceedings contains the written versions of 72 papers selected from the contributed papers, covering a wide spectrum of interests, including innovative computer architecture, parallel and distributed algorithms, automatic parallelization of software, high - speed networks, modelling and tools of parallel systems, performance measurement and analysis, neural networks and their applications, and artificial intelligence.

现代并行处理技术'95国际会议论文集

Proceedings of APPT '95

International Workshop on Advanced Parallel Processing Technologies

责任编辑：王庆育(特约)　史明生

电子工业出版社出版
北京市海淀区万寿路173信箱(100036)
电子工业出版社发行　各地新华书店经销
*
开本：787×1092毫米 1/16 印张：28.25 字数：700千字
1995年9月第1版　1995年9月北京第1次印刷
定价：95.00元
ISBN 7-5053-3304-6/TP·1243

Proceedings of APPT'95

International Workshop on

Advanced Parallel Processing Technologies

September 26—27, 1995, NJU, Beijing, China

Organizers:

- Department of Computer Science and Technology, Northern Jiaotong University
- Architecture Professional Committee of China Computer Federation (APC-CCF)
- Institute of Computer Science, University of Koblenz-Landau
- GMD (the German National Research Center for Computer Science)

In Cooperation with:

- Institute No. 706, Ministry of AeroSpace (PRC)
- Special Interest Group for Parallel Algorithms, Computing Architectures, and System Software (PARS) of GI/ITG
- Special Interest Group for Workstation Computing Systems (APS) of GI/ITG

APPT'95/Beijing

Workshop Committee

Advisors:

Baozong Yuan	Northern Jiaotong Uni. (PRC)
D. Schütt	Siemens AG (Germany)

Program Committee:

General Chairman:

D. Z. Su	APC-CCF (PRC)

Co-Chairmen:

J. Z. Ding	Northern Jiaotong Uni. (PRC)
K. E. Großpietsch	GMD (Germany)
K. Lautenbach	Uni. Koblenz-Landau (Germany)
X. Zhang	Institute of Computer Technology, Academia Sinica (PRC)

Members:

A. Bode	TU München (Germany)
J. L Cheng	Institute No. 706, Ministry of AeroSpace (PRC)
L. Du	Northern Jiaotong Uni. (PRC)
G. R. Gao	McGill University (Canada)
Z. Han	Northern Jiaotong Uni. (PRC)
R. Hofestädt	Uni. Leipzing (Germany)
H. K. Huang	Northern Jiaotong Uni. (PRC)
X. D. Huang	Uni. Koblenz—Landau (Germany)
O. Krämer-Fuhrmann	GMD (Germany)
J. F. Liu	STONE Group Corporation (PRC)
S. W. Luo	Northern Jiaotong Uni. (PRC)
E. Nett	GMD (Germany)
B. M. Qin	SUN H. K. Computer Co. LTD. (PRC)
K. D. Reinartz	Uni. Erlangen (Germany)
M. Sowa	Uni. of Electro—Communications (Japan)
Ch. Steigner	Uni. Koblenz—Landau (Germany)
D. Tavangarian	Fern Uni. Hagen (Germany)
W. M Zheng	Tsinghua Uni. (PRC)

APPT'95/Beijing

SUPPORTERS

. National Natural Science Foundation of China (NSFC)
. Deutsche Forschungsgemeinschaft (DFG)
. Daimler—Benz AG, Germany

EDITED BY EDITORIAL COMMITTEE OF APPT'95

C. X. Yu (Institute No. 706, Ministry of AeroSpace)
K. E. Groppiebuh (GMD)
J. L. Cheng (Institute No. 706, Ministry of AeroSpace)
J. Z. Ding (Northern Jiaotong Uni.)
R. Hofestadt (Univ. Leipzig)
H. K. Huang (Northern Jiaotong Uni.)
X. D. Huang (Univ. Koblenz—Landau)
S. W. Luo (Northern Jiaotong Uni.)
W. M. Zheng (Tsinghua Uni.)

Chairman's Address

It is my honour to avail this opportunity to extend my hearty welcome to you—professors, scientists and specialists from China, Germany, HongKong, Japan, Kuwait, Singapore and the United States of America. September—the best season of Beijing always sees beautiful happenings of great importance, one of which is the International Workshop on Advanced Parallel Processing Technologies — APPT'95 Beijing, China. Nodoubt, this workshop is bound to be an effective and fruitful success.

A total of 72 papers have been selected on the basis of originality, relevance and quality considerations, out of more than 100 papers. This is done as a team work by a panel of referees from various universities and institutes. The Program Committee has approved this final selection and here at the same time highly appraises those contributed but not accepted papers.

The Proceedings contains the written versions of 72 papers selected from the contributed papers, covering a wide spectrum of interests, including innovative computer architecture, parallel and distributed algorithms, automatic parallelization of software, high- speed networks, modelling and tools of parallel systems, performance measurement and analysis, neural networks and their applications, and artificial intelligence.

This is the first workshop and it is hoped to be held in future alternately in China and Germany. It will provide a forum for scientists and engineers to exchange up-to-date research results and experiences about parallel computer systems as well as to establish long-term cooperation in developing innovative products towards future market situations.

The Program Committee feels very much obliged to acknowledge the referee-panel, the session chairmen, the invited speakers as well as the contributors; without whose contributions this Workshop might become impossible.

We hope the present publication of general interest to those working on the computer architecture in their various disciplines.

On behalf of the Program Committee, I'd show my best wishes to you all, may this confernece be an informative conference, may your stay be an enjoyable stay. I'm looking forward to seeing my dear coleagues again in not far future at conference as this.

Prof. Dongzhuang Su

Dongzhuang Su

Program Committee Chairman
of APPT'95
1995. 9

The Operational Mechanism of A Bus Bridge Network[1]

Li Wei and Jin Li-Jie
Computer Science Department,
Beijing University of Aeronautics and Astronautics
Beijing, China, 100083
Email: {liwei, jlj}@cs.sebuaa.ac.cn

Abstract

The Bus Bridge Protocol (BBP) is a multi-computer interconnected protocol that is used to construct parallel computer clusters. It was put forward in 1992. The original intention is to design a kind of the relay protocol to satisfy the connection of devices and processors which use different bus standards. In recent years, researches and experiments demonstrate that the BBP has advantages with constructing Scaleable Parallel Computer Systems. This paper introduces its principle and the operation mechanism of the bus bridge multi-computer interconnected network based on the BBP. And then we discuss some key feature parameters such as blocking probability, setting-up delay, effective utility ratio and their relations in order to provide a theory to prove the implementation of a scaleable parallel computer system based on the BBP.

Keywords: Bus Bridge Protocol, Bus Bridge Network, BBP link, Scalable Parallel Computer

1. Mechanism of Bus Bridge Network

The multicomputer interconnected protocol, Bus Bridge Protocol (BBP), was presented in 1992. The original intention is to design a kind of relay protocol to satisfy connection of devices and processors which use different bus standards. In recent years, the research and experiment demonstrate that: BBP has advantages with constructing Scaleable Parallel Computer System.

In [1], the concepts, targets and implementation methods of the Bus Bridge were presented, a hierarchical model of Bus Bridge interconnected protocol was discussed , and the setting of function and the implementation method in each hierarchy are studied. Based on them, this paper studies the operation mechanism and feature parameters of the multicomputer interconnected networks based on the BBP.

Exchanging networks adopted in multicomputer interconnection have two kinds of basic ways of work: circuit switching mode and packet switching mode. As pointed out in [2], the Bus Bridge is able to support the transformation and transmission of bus signals among processor nodes. So the

[1] This work is supported by the Chinese High -Tech Programme (863-306-01-03-02) and the Chinese Aeronautics Foundations(95F)

circuit switching mode is adopted to implement relays. The reasonability and necessarity of the selection are discussed in the following sections of this paper.

1.1 A comparison between circuit switching mode and package switching mode

In data communication networks based on package switching, the data blocks are named as 'packets'. The packets are sent to an object node from a source node. In circuit switching based networks, a special transmission link is set up when communication between a couple of nodes or among a group of them is needed. This link will be kept on until the transmission is terminated. A synthetical network that combines the two technologies described is being developed.

The circuit switching mode is the first practical switching technology. It has been used in telephone systems for more than one hundred years. Its characteristices are relatively simple, clear and definite. When there is a connection request among nodes, it is necessary to wait for Tw, the set up time of physical link, before transmission. After finishing the task of transmission, network still needs to withdraw this link so as to free the network resource. Once physical link is set up, the overhead of transmission along this link is not very large. This is because there is no routing selection and blocking treatment. During data communication, physical links are maintained no matter whether the data flow is identical or not. The physical link is canceled under the request of the node which set it up .

The packet switching mode is a kind of store and forward mode. During transmission, messages are divided into packets, attached addresses, control, check sum and other information. Those packets are transmitted to an object node from the source node through some midway nodes. In each midway node, packet is temporarily stored and additional controls and check information are processed. The packet switching mode can be further divided into two operation methods namely the virtual circuit and the data diagram. The virtual circuit has some features that simulate to the circuit switching mode. It requires to start end-to-end virtual circuit establishing regulations. These regulations stipulate that the first packet of a message must carry an object address. Once data transmission process begins, the following packets are transmitted along the path that has been set up and needn't carry an address segment anymore. Packets are transmitted in sequence. If errors are found in a packet, it is necessary to transmit this packet again. After data transmission is finished, the end-to-end path is shutdown. When running in the data packet mode, network regards packets as independent units, when packets cross each midway node, they are checked one by one, but we still can't guarantee that the transferring sequence of packets is consistent with the sending sequence. Therefore, each packet must contain its own object address.

Packets coming from different source nodes can share channels and packets are transferred continuously. Thus efficiency of the networks can be improved through this way. Correspondingly, in the circuit switching mode, if source nodes transfer information in a burst way, network resource may be wasted during connection process.

A disadvantage of the packet switching mode is that packet transferring delay can be larger than that of the circuit switching mode. In the circuit switching mode, once physical link is set up, delay of data transmission in network is only caused by propagation. On the other hand, during packet switching, packets flow from one node to another. They are stored and then forwarded at each node. The produce and reorganization of pockets are another reason of delay at a continuous and even data flow among nodes.

When emphasizing quick response to interact among nodes, the circuit switching mode is more suitable. For example, responding to information coming from a network at the bus signal level, the circuit switching mode can get quicker reaction than the packet switching.

Through modeling these two switching modes, [3] compares their reaction under different network utility ratios and different data scales by using real examples. We can see that connection delay of the packet switching mode is shorter than that of the circuit switching when data scale is

about 40 to 300 bytes, but the connecting delay of the circuit switching mode is comparatively steady for a wide range.

This conclusion is also suitable to the Bus Bridge network based on parallel data and control links because serial link and parallel link are almost equivalent from the point of view of the information transmission. There are differences in the implementation method and presented performance. When abstracting the models of the circuit switching mode and the packet switching mode, we did not emphasize their corresponding implementation method in precious discussion. There is a new question following the research of the network features, i.e. using the Bus Bridge network in message passing mode that is identical with a common network or in a distributed shared memory system. This question is based on the results of a network feature study. Actually, the requirement that the Bus Bridge network adopts the circuit switching mode comes from the unfixed data flow length caused by memory access and the response speed that needed at bus transaction level.

If some high speed packet switching networks, such as FDDI, are used to transfer data and encode bus transactions, we still can obtain reasonable speed under the condition of satisfying the consistency of accessing on local memory and remote memory. Therefore the packet switching mode is not unfeasible.

1.2 Operational Principles of Switching Node in Bus Bridge Network

The switching node (SN) is an important component of the Bus Bridge network. Basic structure of an SN is a fully connected multiport network with a small dimension. It is controlled by an internal processor. External ports of the SN meet the request of the BBP. They can connect not only with a terminal node (TN) but also with other SNs. The number of the SN's ports is 2^n (n=2,3,4, ...) in order to decrease volume and cost of the SN.

1.2.1 Organization of Switching Node

Figure 1 gives out a frame diagram of the SN's main function blocks. We can see that the SN with 2^n ports contains a fully connected network consisting of 2^n inputs and 2^n outputs. The port i $\left(0 \le i \le 2^{n-1}\right)$ is combined by the ith input and the ith output. When the connection between the port i and the port j is established, the input of the port i and output of the port j are set up to connect, and vice versa. Thus the connection between the port i and the port j is a fully duplex. The current flow direction of this connection (that is the direction of data flow) is controlled by an R/W* signal of the port i. The LSIi and LSOi are an input and output bypass monitor of the part

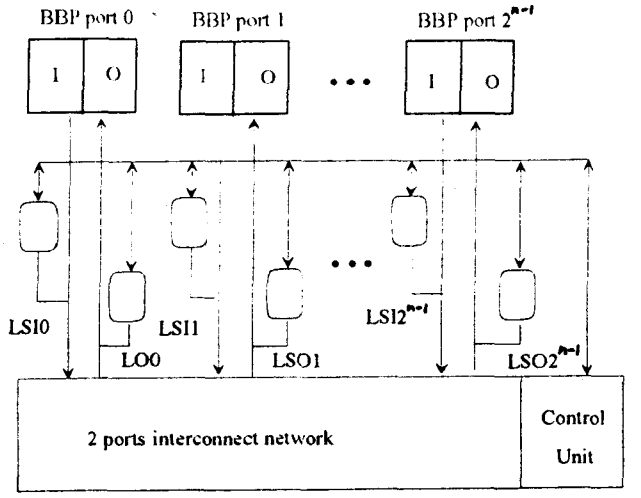


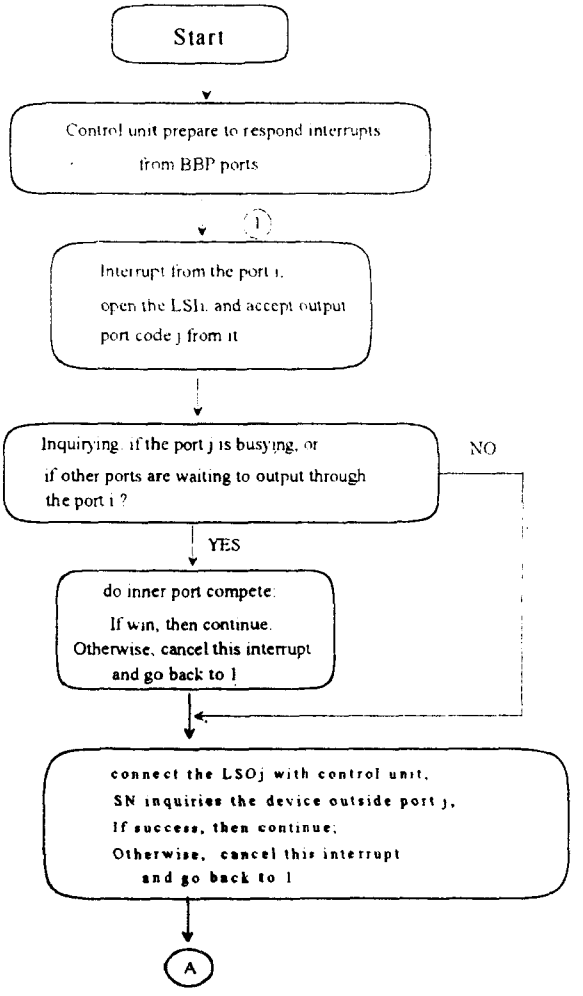
Fig. 1 Functional block diagram of an SN

monitor is an interacting path between the monitor unit of the SN and other nodes (TN or SN) that is connected by the BBP ports of the SN. It is also used to monitor running statuses of an active link. The monitor unit is a control center of the SN. It is in charge of setting up requirement from external link and judging the competition of internal pathes. It also supports the setting and removing of the SN's internal port connections. The SN can offer duplex connection of 2^{n-1} pairs of ports. It can also connect with all devices equipped with the BBP port.

1.2.2 Working Routine of Switching Node

After the power is up, an SN is in a passively waiting state. What the SN waits for is a link setting signal coming from any BBP port. When receiving the requirement coming from a port, the monitor unit will accept link set up instructions, and make corresponding responses according to the arbitration result of internal conditions. Once the SN finishes the current connection request, it monitors this internal connection, and is ready to respond to the request of changing the work state of this link. At the same time, it keeps an eye on other ports and (or) the internal connection.

Figure 2 gives the working procedure of the SN's monitor unit.



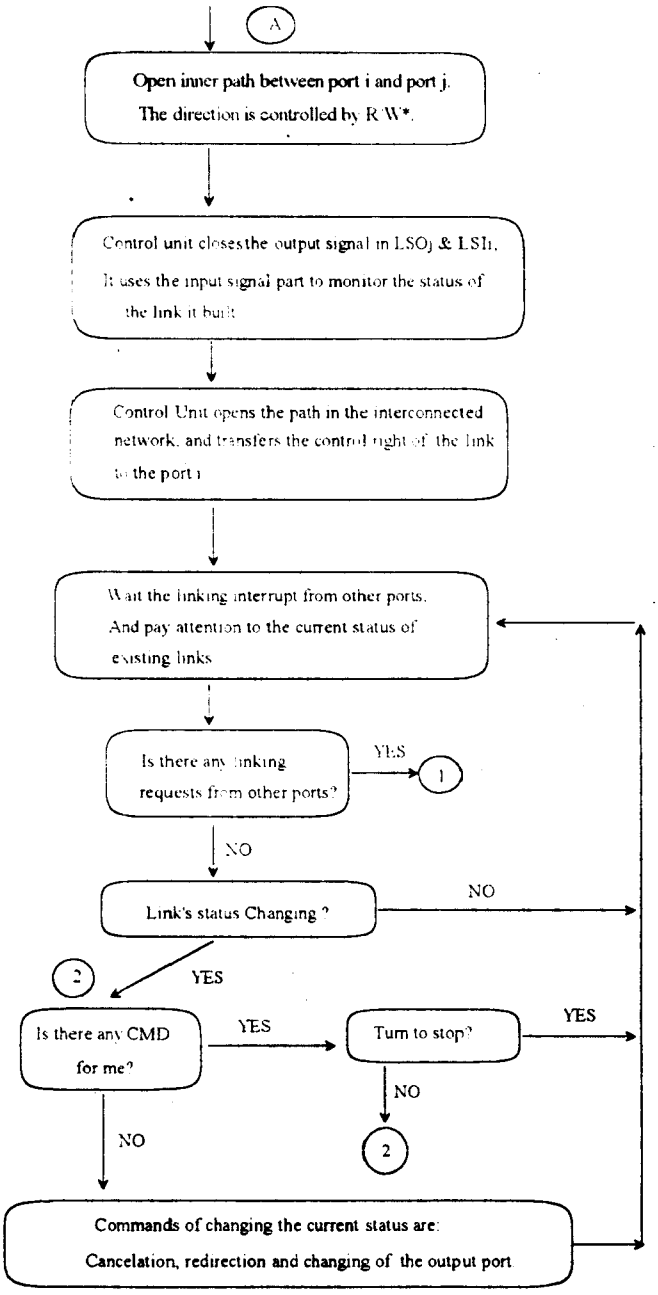


Fig. 2 The working processing graph of the SN

In Figure 2, the judgement of the internal priority can adopt many kinds of methods such as FIFO, polling and so on. In order to support the network layer of the BBP, the SN uses a method that firstly compares the priority of links that have conflict, and then compares the code value of the SN's ports from which the requests enter.

1.3 Operational Mechanism of the Bus Bridge Network

When two terminal nodes (TNs) are interconnected by a link that follows the BBP, the connection between them is named as *a Bus Bridge connection*. When more than two TNs are interconnected by one or more switching nodes, the connections among those TNs are called *a Bus Bridge Network (BBnet)*. Introducing the bus bridge network aims at embodying much more TNs into a multicomputer system. The BBnet is more scalable than a bus. The cost of the scalablity is a certain delay of the link building.

The working of the BBnet is controlled by the net layer of the BBP. Distributed runtime management is used to enhance the scalability of the BBnet. Those TNs in the BBnet are in the equal position. The TN which wants to send its subtasks to other TNs becomes a temporary client in the BBnet.

After it receives those subtasks, a TN becomes a server group. SNs in a BBnet are almost the same except their individual names. Each SN can cooperate with its neighborhood. Any SN is unnecessary to know the topology and working states of the whole BBnet.

At the connecting stage, conflicts of link are automatically resolved by SNs that along the path. Once the TN which inquires the connection has gotten the path, all things it should do are to wait for a reply message from the BBnet if the path OK or connecting inquire is failed.

At the maintaining stage, all SNs along the path monitor the running status of the path and do not join the communication process between TN pairs.

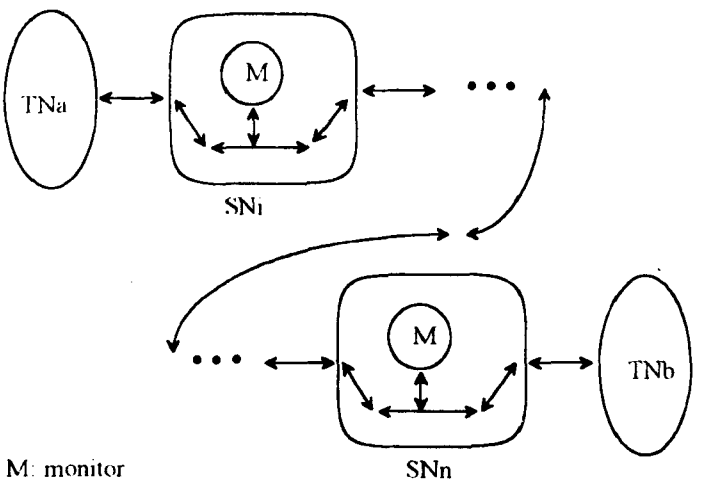


Fig. 3 A schematic diagram of the maintaining stage of a multilevel bus bridge link.

In this Fig. 3, the TNa controls the current link. The connection between the TNa and the TNb is an immediate path. It includes all signals in the terminal layer of Bus Bridge Protocol except the

arbitration signal bus. The arbitration bus is valid only between two directly connected BBP compatible ports. When the TNa changes the value of DIR*, one of the control signal in BBP compatible port, SNs alone the path will automatically change their data transfer direction. When TNa sets C/D* to 1, SNs will open data input channel so that they can receive control commands for following works from the TNa. There are three alteratives for the TNa to cancel the Bus Bridge connection with the TNb:

(1) The TNa cancels the path according to the opposite order of path building;

(2) The TNa maintains a long enough 'cancel ready' command so that the TNb and each SN alone the path can accept this cancel information, then the TNa gives a cancelation;

(3) According to the same order of setting path, the TNa notices the neighbor nodes to remove path with it and send a cancel-command to the next node on the path until it reaches the TNb.

In scheme 2. It is difficulty to decide the delay time for the cancel-command by the TNa. This is because that SNs on the path should monitor different amount of internal links and work in different states. The distinction of scheme 1 and 3 is that scheme 3 need that SNs join the canceling progress, but scheme 1 need only that SNs respond to the 'link close' command.

The setting up progress of long paths is protected by a priority strategy of the network-layer of the BBP. When path setting up progress is failed at some stage, the BBnet asks the TN which fails in obtaining its path to free all resources it has occupied. It will retry the path getting progress after a while or use a new path if it is possible.

2. Analysis of the feature parameters of the BBnet

From the point of view of a TN, the average link setting delay and data transmission delay are two important parameters for the BBnet. From the point of view of parallel system designers and users, those parameters about the aggregate bandwidth of the BBnet, the blocking probability of a link inquiry and the overhead of maintenance a link can give much more features of the BBnet.

2.1 Discussion for blocking probability of the BBnet

There are two types of blocking in the BBnet. One is caused by requirement conflicts on network resources. Another is caused by conflicting on an objective TN (OTN). In this section, we take a BBnet as a random server system, TNs in the BBnet as a group of clients. The server abilities and blocking situation of BBnet are consided.

Suppose that the number of different BBP links offered by the BBnet between the TNa and the TNb is n. Different BBP links may have some overlap parts. Each BBP link has no cycles. According to the demand of the BBP's network layer, if the TNa is failed in getting any of n possible BBP links to talk with the TNb, then its link inquiry is withdraw from the BBnet. The TNa will retry after a certain delay.

When a BBP link requirement is produced by the TNa, the BBnet can be in the following server statues:

S0: All possible BBP links are idle;

S1: One BBP link (or its part) is occupied, but other n-1 BBP links are idle;

...

Sm: m BBP links (or their parts) are busy, but other n-m BBP links are idle;

Sn: All n BBP links are busy.

[Definition 1]

Let $N = \{N_1, N_2, ...N_n\}$ be a set of all BBP links which a BBnet can offer between the TNa and the TNb. Suppose that $i \in N, i = i_1 \cup i_2 \cup i_3 \cup ... \cup i_m$, and $i_j (1 \le j \le m)$ is one of the segments of the BBP link i. $TNG = \{TN_x | x \in \{1, 2, ..., l\}\}$ contaions all those TNs whose number is l in the BBnet. When there is at least one element of i serving for the BBP connection in TNG, and if $TN_a \in TN$, then $TN_b \notin TNG$, or if $TN_b \in TNG$, then $TN_a \notin TNG$, we call it the BBP link i ***busying***.

The blocking probability of a BBP link set up inquiry between the TNa and the TNb is the probability of the random serve system, BBnet, working in status Sn. We call it P_n.

The different application programs running on the TN will product different distribution type of BBP link inquiries. Approximately, the distribution type of the BBP link inquiries produced by the TN satisfies the Poisson distribution. Then the link serving system of the BBnet can be seen as an M/M/n queuing system. Its state transfer chart is as follow:

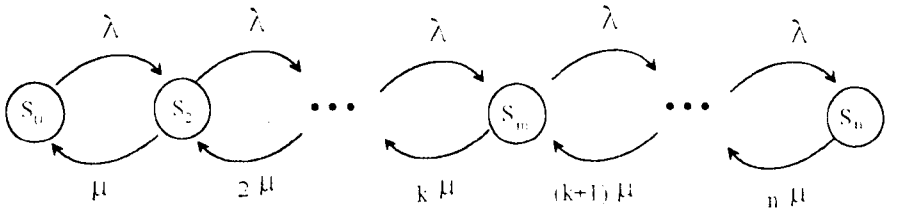


Fig. 4 The serve status transfer chart of the BBnet.

Suppose λ is the productive ratio of the inquiry stream, (also termed as strength), μ is the serve strength of the single BBP link, which represents that the inverse of the average time of the BBP link serving an inquiry.

Based on the balance principle of the system status:

for the status S_0, we have $\lambda P_0 = \mu P_1$, $P_1 = \rho P_0$. (1)

for the status S_1, we have $\lambda P_1 = 2\mu P_2$, $P_2 = \frac{\rho^2 P_0}{2!}$. (2)

Where $\rho = \frac{\lambda}{\mu}$.

Similarly, we have

$$P_m = \frac{\rho^m}{m!} P_0; ...; P_n = \frac{\rho^n}{n!} P_0 \quad (3)$$

According to the canonical condition: $\sum P_i =$, we have

$$P_0 + \frac{\rho}{1!} P_0 + \frac{\rho^2}{2!} P_0 + ... + \frac{\rho^m}{m!} P_0 = 1, \quad (4)$$

then

$$P_0 = \frac{1}{\sum_{m=0}^{n} \frac{\rho^m}{m!}} \tag{5}$$

Substituting (3) by (5), we have

$$P_n = \frac{\rho^n}{n! \cdot \left(\sum_{m=0}^{n} \frac{\rho^m}{m!} \right)} \tag{6}$$

The equation (6) points out the probability that all n BBP links between the TNa and the TNb are occupied, i.e. the probability that the inquiry from the TNa is refused. It also shows the relationship among the blocking probability, inquiry strength of TNs, average serving time of the BBP links and the total number of the possible BBP links.

2.2 Average Waiting Time of BBP link inquiry.

In the real system, the refused BBP link inquires will not be discarded directly. They will be reproduced after a while under some control strategies. During this time, the TN can either block the process which inquiries the BBP link and does context switch, or wait at the inquiry fail point before retrying. Because the context switch costs a lot of system time, the interval between two successive retry inquiries is an important parameter for the TN to determine whether do context switch or not.

We add a new state into the chart of Fig. 4 to show the situation that when n BBP links are all busy the BBP link inquiry will queue.

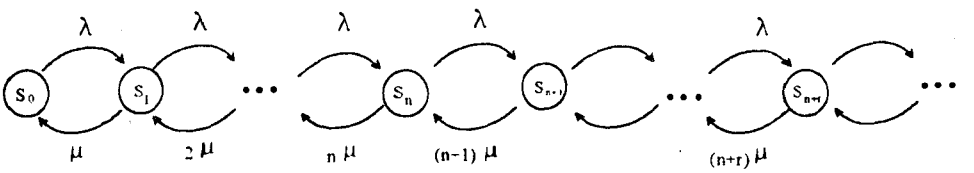


Fig. 5 The status transfer chart with inquiry queuing.

In Fig. 5: S_{n+1} represents that all n BBP links between the TNa and the TNb are busy, and there is an inquiry waiting in the TNa ; S_{n+r} represents that all n BBP links are busy, and there are r inquiries queueing in the TNa.

Suppose that $\frac{\rho}{n} < 1$, then we have

$$P_m = \begin{cases} \frac{\rho^m}{m!} P_0, & \left(0 \le m \le n-1\right) \\ \\ \frac{\rho^m}{n^{(m-n)} \cdot n!} P_0, & (n \le m \le n+r) \end{cases} \tag{7}$$

According to the canonical condition, we have:

$$P_0 + P_1 + \ldots + P_n + \ldots + P_{n+r} + \ldots = 1 \tag{8}$$

Substituting (8) by (7), we get

$$P_0 = \left[\sum_{m=0}^{n} \frac{\rho^m}{m!} + \frac{\rho^{n+1}}{n \cdot n!} \left(1 + \frac{\rho}{n} + \left(\frac{\rho}{n}\right)^2 + \ldots + \left(\frac{\rho}{n}\right)^r + \ldots\right) \right]^{-1}$$

$$= \left[\sum_{m=0}^{n} \frac{\rho^m}{m!} + \frac{\rho^{n+1}}{n \cdot n!} \cdot \frac{1}{\left(1 - \frac{\rho}{n}\right)} \right]^{-1} \tag{9}$$

The average number of BBP link inquiries waiting in the TNa. L, is given as follow:

$$L = P_{n+1} + 2P_{n+2} + \ldots + rP_{n+r} + \ldots = 1 \tag{10}$$

Substituting (10) by (7), we have:

$$L = \frac{\rho^{n+1}}{n \cdot n!} P_0 \left(1 + 2\left(\frac{\rho}{n}\right) + \ldots + r\left(\frac{\rho}{n}\right)^{r-1} + \ldots\right)$$

$$= \frac{\rho^{n+1} P_0}{n \cdot n! \left(1 - \frac{\rho}{n}\right)^2} \tag{11}$$

According to the Little formula:

$$L = \lambda T_w \tag{12}$$

Where, λ is the inquiry arriving strength and T_w is the average waiting time of the BBP link inquiry. Hence.

$$T_w = L/\lambda = \frac{\rho^n P_0}{n\mu \cdot n! \left(1 - \frac{\rho}{n}\right)^2} \tag{13}$$

The equation (13) shows the relationship among λ, μ and n and the average waiting time of the BBP link inquiry.

2.3 Usage Efficiency of BBP Links

From the discussion above, we see that if the number of the possible BBP link n between the TNa and the TNb increase, then the probability of building a successful BBP link is increase. But it needs much more SNs to enlarge the number n. The cost of the BBnet increases at the same time. Hence we want that each BBP link in a BBnet has suitable usage efficiency.

According to the status transfer model in Fig. 5, we know that the number, C, of average occupied BBP links between the TNa and the TNb can be shown as follow: