

A formal description method for P2P network models

Shen Jun Huang Yuan Song Jinjing

(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: To meet the requirements of modeling the new modality of peer-to-peer (P2P) network applications which have been rapidly developing in the Internet recently, a formal description method for modeling multiparty concurrent network interactions is studied. The main characteristics and the classifications of P2P systems are discussed. Considering the requirements of P2P application modeling and referring to the component-based modeling thought, a description method based on communicating sequential processes (CSP) is proposed for the P2P network models. By using a CSP process group, this method can describe the dynamic interactive relationship which focuses on multiparty concurrent interaction of P2P systems more advantageously and accurately. The application of nondeterministic semantics of CSP in describing the interactive relationship of P2P networks is discussed. The advantages and description abilities of the proposed method are demonstrated through the modeling of a new P2P media-on-demand system.

Key words: peer-to-peer; communicating sequential processes (CSP); component-based thought; interaction

As the computer network develops rapidly, the limitations and flaws of traditional B/S and C/S architecture for network application have been increasingly exposed. The servers often demand too many resources and it has become bottlenecks in the systems. And their drawbacks lead to the emergence of a new peer-to-peer (P2P) network application mode^[1].

Due to progress in many areas, the P2P technology originated from music sharing tools^[2] is now quickly broadening its range of application. The most remarkable characteristic of P2P is that it can convert the traditional storage mode of centralized content storage into the distributed content storage mode. Most of the peer entities have totally equal status. Every peer is an initiative participant and can both request and provide service while interacting with other peers. Thus they can share resources and cooperate together. The distinction between the architecture of C/S and P2P is shown in Fig. 1.

Although P2P has already become a hotspot of network research and application, up to now, the research of modeling P2P, especially the modeling method of the whole system is still quite weak. Moreover there is hardly any study about a formal description method for P2P.

1 Classification of P2P and Necessity of Modeling P2P

P2P network is built on the basis of the existing underly-

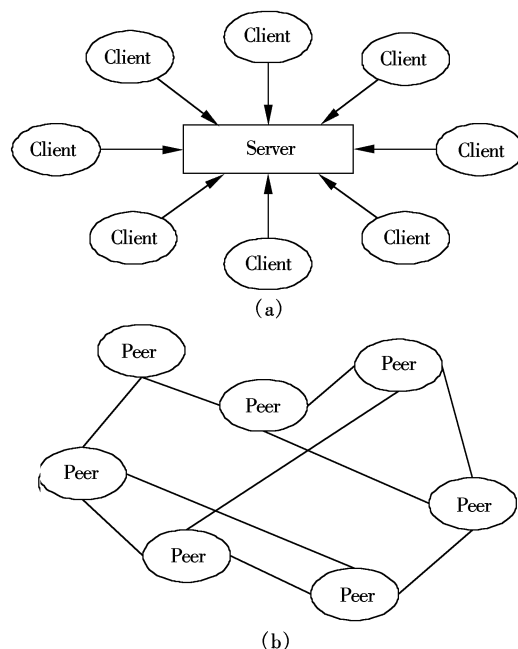


Fig. 1 The distinction between P2P mode and traditional C/S mode. (a) C/S mode; (b) P2P mode.

ing physical network, so it is also called a P2P overlay network. All the peer entities usually simultaneously play both the roles of clients and servers in the traditional network applications. However, there are still some differences among the detailed realizations of different P2P systems. According to the method of service integration, the P2P network models can be roughly divided into the following three categories^[3]:

1) Centralized P2P network: There is a central directory server in the network system. The directory server is responsible for the management and maintenance of the directory information of all nodes in the network. Of course, it is completely different from the traditional application modes. The central server is only responsible for the storage and publication of the indexing information for all of the peer nodes. Every peer node stores all its specific information in its own storage and provides service for other peers.

2) Totally distributed P2P network: There is no central directory server in this kind of system. All of the nodes are completely equivalent. They play both the roles of clients and servers. The whole network is constituted by the connections among adjacent nodes. The coordination of the central server is no longer needed.

3) Hybrid P2P network: This kind of network system integrates the characteristics of the above two modes. Some nodes which have more resources (bandwidth, memory, processing ability etc.) are selected as proxy nodes. The proxy nodes are used to store the information of data and service in some other nodes around it. A proxy node can be regarded as the central directory server of the nodes in its neighbor-

Received 2008-07-30.

Biography: Shen Jun (1963—), male, professor, junshen@seu.edu.cn.

Foundation item: The National Basic Research Program of China (973 Program) (No. 2003CB314801, 2009CB320501).

Citation: Shen Jun, Huang Yuan, Song Jinjing. A formal description method for P2P network models[J]. Journal of Southeast University (English Edition), 2009, 25(1): 36–40.

hood. Thus the nodes work as a centralized P2P network system. And all the proxy nodes are organized as a totally distributed P2P network system. So these systems are called hybrid P2P systems. The selection of proxy nodes is dynamic. The proxy nodes can leave the network systems just like normal nodes. When the network finds that a proxy node is no longer working, it will start a specific process to select a new high-performance node as the new proxy node.

Obviously, the modeling of no matter which kind of P2P network system will undoubtedly involve the description of complex interactive relationships among multiple entities. However, the traditional researches of network modeling almost exclusively use the one-to-one direct interaction model to describe the interactions among the network entities. If we continue using the conventional direct interaction model, the description of P2P applications' architecture will be very difficult or even impracticable. Therefore, it is of great necessity to study new method for the description of network architecture, especially for the modeling of P2P application modes.

2 Formal Description Method for P2P Application Mode

As mentioned above, a P2P network is a kind of typical multiparty application system. Usually a single peer node in a P2P system will directly communicate with a non-fixed number of other peers. However, the traditional network modeling method lacks of the ability to depict multiparty interrelations and to ensure a model's correctness. Especially, the description and verification of the P2P architecture's logical completeness is of great difficulty. According to these, we should borrow some ideas from related research domains.

Recently, the researchers in the software domain have made great progress in the abstraction and analysis work of large-scale software systems by combining the component-based thought with formal theories and techniques. The research of network modeling can follow the same route. For example, in Ref. [4], by taking the research achievements and thoughts of component-based software as references, a formal modeling method for a new generation network service architecture was established. Unfortunately this method still has quite a few limitations when considering the modeling requirements of multiparty network applications^[4-5].

Nevertheless, by introducing interactive viewpoints in architecture and formal means, the above mentioned component-based modeling method has indeed brought some convenience to the description of a network multiparty applications. Accordingly, we intend to do some research in the formal representation of network systems involving multiparty and nondeterministic interactions. Compared with traditional network applications, the concurrent interaction feature of a network is exposed more significantly in P2P systems. Meanwhile among many formal theories and tools, the process algebraic methods have some particular advantages which make them especially appropriate for modeling complex concurrent systems. Along with the explicit concurrency model, process algebraic methods also have good calculative features. They have been successfully applied in software component assembly, network protocol verification and some

other areas. Therefore, in this section, we will use a representative process algebraic method, communicating sequential processes, to discuss the modeling of P2P network systems (especially the description of the interactive relations in the system).

2.1 Communicating sequential processes

The communicating sequential processes (CSP) was created by Hoare^[6]. It is a language for describing patterns of interaction. It is supported by an elegant mathematical theory, a set of proof tools, and an extensive literature. CSP distinguishes processes by their failure sets, and it expresses the equivalence of processes by the failure semantics. CSP has defined many operators to describe the behavior of process as well as the relationships between different processes.

CSP has the same ability of expression as another process algebra method called CCS. An interleaving semantics model is used by both of them to describe parallelism, and this model brings good algebraic properties for the modeling of concurrent systems. However compared with CCS, CSP places more emphasis on the research of concurrent computing from the angle of program design and computer applications. It attaches great importance to the concept of interaction, and one of its fundamental features is to help the design and implementation of those computer systems which continue working and interacting with the environment. A very remarkable advantage of CSP is that the pattern it uses to describe, design and verify complex concurrency systems can be easily understood by most computer practitioners. CSP is also the theoretical foundation of Ada language's concurrent mechanism, distributive programming language Occam, and architecture description language Wright^[6-7].

CSP mainly focuses on the description of concurrent systems' communication through message transmission. The most important fundamental concepts of CSP are as follows:

Event: Event is the elemental unit of a system model. Each event can be an atomic module or a collection of atomic modules.

Process: A process is a part of a system. It is used to characterize the course of activities in the system. All the possible events in the specified process are named by the symbol α followed by the process name. A process can involve 0 or more events. For example, the Stop process corresponds to the inactive state. It does not include any events. The common process operators include: prefix (then) " \rightarrow "; choice " $|$ "; sequential combination (successfully follow) " $;$ "; parallel " $||$ "; nondeterministic or " \square "; interleave " $|||$ ", etc. In addition, both the input operator " $?$ " and the output operator " $!$ " are introduced with the "channel" concept to simplify the expression of communication between processes. They are frequently used in the description of network protocols.

Trace: For process P , trace (P) represents a possible course of its running activities. It is constituted by a group of events. It can be seen as a kind of ordered arrangement of events. The operations such as connection and restrictions can be applied to traces. The traces (P) correspond to the trace set which include all the potential traces of process P .

Specification: Specifications are the rules that traces must obey. If a process P satisfies a specification S then all the traces of process P satisfy S , i. e. $P \text{ sat } S \Rightarrow (\forall \text{ tr, tr} \in \text{traces})$

$(P) \Rightarrow \text{tr sat } S$.

CSP uses failure semantics to explain the equivalence of processes. The processes can be identified by their failure set. It is worth noting that CSP has a quite powerful expression ability and a relatively completed theory for nondeterministic process behaviors.

2.2 Description of the P2P network interaction

As mentioned above, the component-based thought originated from the software field can be used as a reference for the formal modeling of network systems. By decomposing the system into various components and the interactions among components, studying the intrinsic link among different parts of the system from a higher abstraction level as well as the research and analysis of a whole P2P system's behavior can be supported. We can adopt a method similar to the one used in Ref. [7] to describe P2P network systems by using the elemental modeling units named network entity components and interaction connectors, respectively. A network entity component represents a basic network functional element which can provide specific service and is involved in the interactions ruled by some protocols, while interaction connectors are used to model the interactive interrelationships among different entity components. Moreover, the detailed dynamic characteristics of both entity components and interaction connectors can be depicted by a group of CSP process expressions severally^[8].

It is not difficult to find out that in many P2P network systems there are multiple peer entities with the same or similar interactive behaviors inside the peer entities and their access points to the environment. Traditionally, we have to define a number of independent processes separately and repeatedly for every one of them. For this situation, we can introduce the symbol transformation method of CSP^[6] to simplify the definition of similar processes.

Suppose that f is an injective function which maps the alphabet of process P to a set of symbols (events) named A .

Then $f(P)$ is used to represent such a process: When process P is executing event C , $f(P)$ is executing the event $f(C)$. Moreover, the process complies to the following rules:

$$\alpha f(P) = f(\alpha P) \quad (1)$$

$$\text{traces}(f(P)) = (f^*(s) \mid s \in \text{traces}(P)) \quad (2)$$

The form $f(C)$ represents the image set of C which consists of the mapping result of every element in symbol set C .

By defining many different injective functions, a process can be mapped to a group of processes with similar actions. Also a single injective function can be applied to a number of processes, for example, the processes P_1, P_2, \dots, P_n being mapped will result in $f(P_1), f(P_2), \dots, f(P_n)$, respectively. This is just the kind of circumstance which we often encounter during our description procedure of P2P networks.

It is not enough to have only the processes describing independent components' interactions for the property analysis of the entire network system. We often need to consider the combination and cooperation of multiple components. Fortunately, the researchers in some related research domains of

computer science (especially software architecture) have already thoroughly concluded different forms of component composition. Several important composition forms and their corresponding CSP semantics are shown below^[9]:

Parallel component composition (corresponding to the “||” operation, means the parallel execution of different component functions); selective component composition (corresponding to the “□” operation), replicating component composition (corresponding to the “@” transformation that implements the multiple instances of a component class in the way of parallel); sequential component composition (corresponding to the “;” operation) and interruptive component composition (corresponding to the “^” operation). In the P2P network system, the interactive events corresponding to the selection and sequential operations generally only emerge inside a single entity, and the interactions between different entities (including the distributed concurrent interactions among peer entities) can mainly be modeled by parallel composition and replication composition. Moreover, in the P2P systems, as the many-to-many interactions are usually much more complicated than the traditional server-client relationship, the events of different processes or even entities often cannot be divided into disjointed alphabets clearly. Sometimes the occurrences of different operations are also not so regular. In these cases, we can utilize the nondeterministic semantics of CSP to cope with the irregular process behaviors. For example, we have to use the interleaving operation “|||” instead of the normal parallel operation “||” when different processes include the same events. It usually corresponds to the situation that all multiple peers have access to some certain resources and there are no explicit priorities defined. For lack of space, the detailed composition reasoning will not be discussed in this paper.

3 Application Example

In this section, we present the application of our formal modeling method through a simple example. The exemplified system shown in Fig. 2 is abstracted from the interactive relationship of a P2P media-on-demand system MOD of SEU^[10], which is based on the Pastry^[4] overlay network. And the interaction is simplified and slightly adjusted for convenience. The protocol interaction involves the following participants: a non-fixed number of peers which send and receive media flows and publish various controlling information.

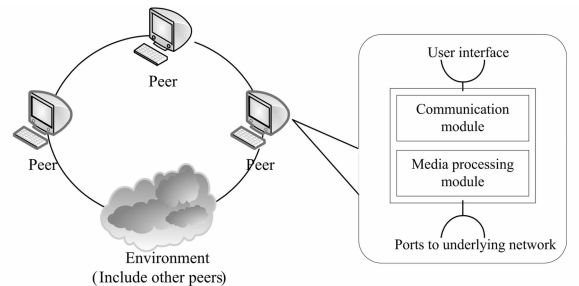


Fig. 2 The general architecture of the MOD P2P system

For every P2P node entity, actually it only interacts with its upper user interface and underlying network carrier. These two kinds of interactions run concurrently with its in-

ternal processing. So the interactional behavior of the corresponding entity component for a single peer can be described briefly as follows:

/* The identifiers in the process expressions are defined as follows:

Spin/Spout: The input/output channel to the upper user interface;

Npin/Npout: The input/output channel to the underlying network peripherals;

ProcessIns(): Process user instructions;

ProcessMsg(): Process messages come from the underlying network and acquire the data of media stream;

GetList(): Return the list of nodes which the messages should be sent to;

USEROUT/NETOUT: An event to output data to the user interface/underlying network;

FORWARD: An event to forward a message to other peer nodes;

GET: An event to receive a message. */

Pnode. csp = (Spin? ins → ProcessIns (ins) → Pnode)

/* ins is the input instruction from the user */

|| (USEROUT → Spout! < Info, Media > → Pnode)

|| (Npin? nmsg → med = ProcessMsg (nmsg) → (FORWARD → { for every nodeaddr in GetList () } Npout! < nodeaddr, nmsg >)

□ Get → Spout! < Info, med >) → Pnode)

|| (NETOUT → { for every nodeaddr in GetList () } Npout! < nodeaddr, nmsg > → Pnode)

When we aim at the cooperation of multiple peers, some internal interactions of the peer nodes no longer need to be taken into account. Only the events related to the interaction between peer nodes have to be abstracted. So the interactions of the multiple above-mentioned peers can be depicted as

/* NP_i corresponds to the process of the i -th peer. */

NP_i . csp = (NP_i . Npin? nmsg → ProcessMsg (nmsg)

→ (FORWARD → { for every nodeid in GetList () }

NP_i . Npout! < nodeid, nmsg >)

□ GET → GetProcessmsg () → Pnode)

/* GetProcessmsg () is used to process all the received messages */

|| (NETOUT → nmsg = GetProcessmsg () → { for every nodeid in GetList () } NP_i . Npout! < nodeid, nmsg > → ✓
| NETOUT → Pnode)

Obviously, different peer entities all play equivalent roles and have identical dynamic characteristics. So the process labeling transformation can be used to define multiple peers in batch; meanwhile, their combination can be considered as replicating composition. Then the inter-entity interactions can be represented by

$$|||_{i=1,2,\dots,n} (NP_i. csp)$$

As illustrated through the above simple example, by introducing the CSP method and component-based thought, we can describe the behavioral properties of network functional units (especially the peer entities) in P2P systems much more efficiently than before. Furthermore, the nondeterministic

expression ability of CSP is also very useful for both the modeling of interactions among different peer entities and the inner behavior of a single entity.

4 Conclusion

This paper analyzes the P2P network applications' modeling requirements and divides the application modes of P2P systems into three categories due to the methods of service integration. By referencing the theory of process algebra, a formal description method aimed at the modeling of P2P networks is proposed. The forms of component composition for a P2P network using CSP are discussed. The new description method can essentially improve the efficiency of P2P system modeling and establish a stable foundation for the logic-based analysis of multiparty interactions. Future works include the in-depth research of compositional reasoning and the verification method for the models of P2P interactions.

References

- [1] Fox G. Peer-to-peer networks [J]. *Web Computing*, 2001, 3 (3): 75 – 77.
- [2] Napster LLC. Napster [EB/OL]. (2007-12-11) [2008-04-30]. http://www.napster.com/about_napster.html.
- [3] P2P Group. Peer 2 peer—file-sharing. [EB/OL]. (2008-03-01) [2008-04-30]. <http://www.peertopeer.org>.
- [4] Yang Peng, Wu Jiagao. An interaction based service oriented model of network architecture [J]. *Acta Electronica Sinica*, 2005, 33(5): 804 – 809. (in Chinese)
- [5] Nenad M, Richard N T. A classification and comparison framework for software architecture description languages [J]. *IEEE Transactions on Software Engineering*, 2000, 26 (1): 70 – 93.
- [6] Hoare C. *Communicating sequential processes* [M]. Englewood Cliffs: Prentice Hall International, 1985: 21 – 87.
- [7] Yang Peng, Liu Ye. Interaction-based network service architecture description language [J]. *Journal of Southeast University: Natural Science Edition*, 2006, 36 (5): 685 – 689. (in Chinese)
- [8] Huang Yuan, Shen Jun, Song Jinjing. The research of multiparty-application modeling supported network architecture description method [C]//*Proceedings of International Symposium on Advances in Computer and Sensor Networks and Systems*. Salt Lake City: Aardvark Global Publishing, 2008: 124 – 130.
- [9] Mark Gaynor, Scott Bradner. A real options framework to value network, protocol and service architecture [J]. *Computer Communications Review*, 2004, 34(5): 31 – 38.
- [10] Fang Qun, Wang Min, Ji Yi. Design and implementation of media on demand server based on RTSP/RTP [J]. *Computer Engineering and Design*, 2006, 27(1): 4 – 6. (in Chinese)
- [11] Castro Miguel, Druschel P, Hu Y Charlie, et al. Topology-aware routing in structured peer-to-peer overlay networks [EB/OL]. (2003-08-15) [2008-04-30]. <http://www.research.microsoft.com>.
- [12] Rowstron Antony, Druschel Peter. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems [C]//*18th IFIP/ACM Intl Conf on Distributed Systems Platforms*. Springer-Verlag, 2001: 329 – 350.
- [13] Clark D, Tennenhouse D. Architectural considerations for a new generation of protocol [C]//*Proc ACMSIG-COMM*. New York, NY, USA: ACM. 1990: 200 – 208.

[14] Mahajan R, Castro M, Rowston A. Controlling the cost of reliability in peer-to-peer overlays[C]//*Proc of the 2nd Intl Workshop on Peer-to-Peer Systems(IPTPS 2003)*. Springer-Verlag, 2003: 368 – 374.

[15] Zegura E, Calvert K L, Bhattacharjee S. How to model an internetwork[C]//*Proc of the IEEE Infocom'96*. San Francisco: IEEE Computer Society Press, 1996: 594 – 602.

P2P 网络模型的形式化描述方法

沈 军 黄 元 宋金晶

(东南大学计算机科学与工程学院,南京 210096)

摘要:为了更好地适应对 Internet 中迅速发展的新形态 P2P 网络应用进行建模的需求,深入研究了适于描述多方并发网络交互的形式化建模方法. 首先,对 P2P 网络系统模型的各项特征及其分类进行了探讨. 然后,参照构件化建模的研究思路,基于对 P2P 应用建模需求的分析,提出了一种运用通信顺序进程(CSP)进行描述的 P2P 网络应用的形式化描述方法. 借助 CSP 进程组表示,该方法能够更为方便精确地描述 P2P 系统中以多方交互为主的动态交互关系,探讨了 CSP 非确定性语义在 P2P 交互建模中的应用. 最后通过一个新型 P2P 媒体点播系统的实例展示了以上描述方法的应用,验证了该方法在描述并行交互关系等方面的能力.

关键词:P2P;通信顺序进程;构件化思想;交互

中图分类号:TP393