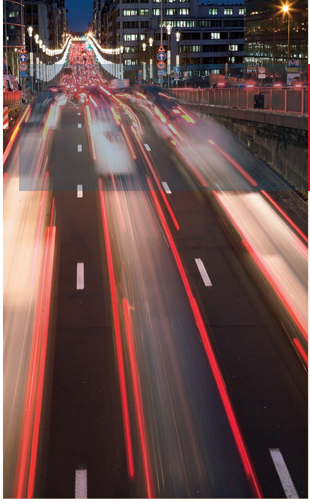

Editor: **Lefei Li,** Tsinghua University, lilefei@tsinghua.edu.cn

Cloud Computing for Agent-Based Urban Transportation Systems

ZhenJiang Li and Cheng Chen, *Chinese Academy of Sciences*
Kai Wang, *National University of Defense Technology*

Agent-based traffic management systems can use the autonomy, mobility, and adaptability of mobile agents to deal with dynamic traffic environments. Cloud computing can help such systems cope with the large amounts of storage and computing resources required to use traffic strategy agents and mass transport data effectively. This article reviews the history of the development of traffic control and management systems within the evolving computing paradigm and shows the state of traffic control and management systems based on mobile multiagent technology.

Intelligent transportation clouds could provide services such as decision support, a standard development environment for traffic management strategies, and so on. With mobile agent technology, an urban-traffic management system based on Agent-Based Distributed and Adaptive Platforms for Transportation Systems (Adapts) is both feasible and effective. However, the large-scale use of mobile agents will lead to the emergence of a complex, powerful organization layer that requires enormous computing and power resources. To deal with this problem, we propose a prototype urban-traffic management system using intelligent traffic clouds.

History of Traffic Control and Management Systems

When an IBM 650 computer was first introduced to an urban traffic-management system in 1959, the traffic control and management paradigm closely aligned with the computing paradigm in IT science.[1] As Figure 1 shows, this paradigm has five distinct phases that mirror the five stages in the deployment of the traffic control and management paradigm.

In the first phase, computers were huge and costly, so mainframes were usually shared by many terminals. In the 1960s, a whole traffic management system always shared the resources of one computer in a centralized model.

Thanks to large-scale integrated (LSI) circuits and the miniaturization of computer technology, the IT industry welcomed the second transformation in computing paradigm. At this point, a microcomputer was powerful enough to handle a single user's computing requirements. At that time, the same technology led to the appearance of the traffic signal controller (TSC). Each TSC had enough independent computing and storage capacity to control one intersection. During this period, researchers optimized the control modes and parameters of TSC offline to improve control. Traffic management systems in this phase, such as TRANSYT, consisted of numerous single control points.

In phase three, local area networks (LANs) appeared to enable resource sharing and handle the increasingly complex requirements. One such LAN, the Ethernet, was invented in 1973 and has been widely used since. During the same period, urban-traffic-management systems took advantage of LAN technology to develop into a hierarchical model. Network communication enabled the layers to handle their own duties while cooperating with one another.

In the following Internet era, users have been able to retrieve data from remote sites and process them locally, but this wasted a lot of precious network bandwidth. Agent-based computing and mobile

1541-1672/11/$26.00 © 2011 IEEE
Published by the IEEE Computer Society

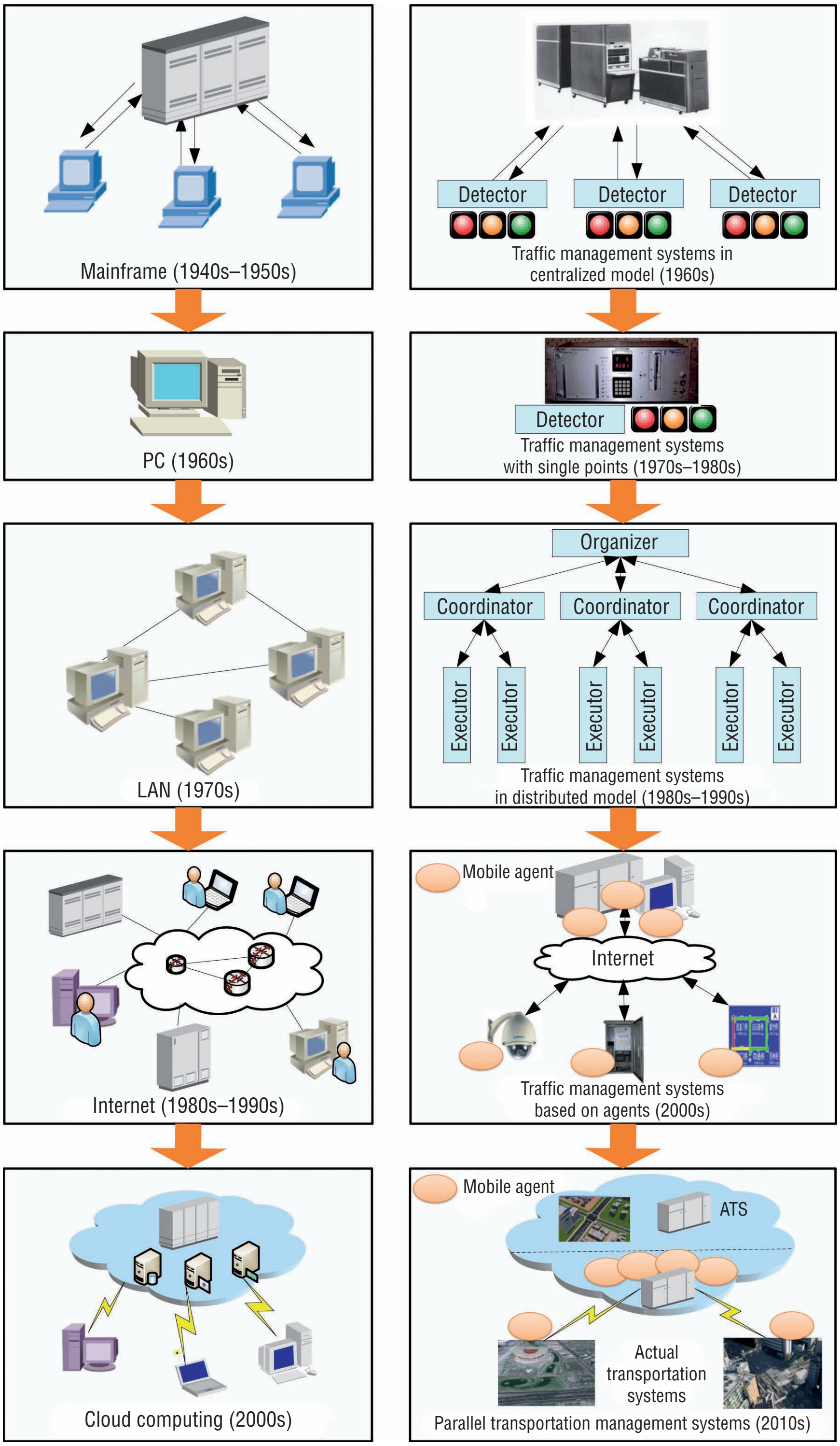

Figure 1. Relationship between the shifts in the computing and traffic management paradigms.

agents were proposed to handle this vexing problem. Only requiring a runtime environment, mobile agents can run computations near data to improve performance by reducing communication time and costs. This computing paradigm soon drew much attention in the transportation field. From multiagent systems and agent structure to ways of negotiating between agents to control agent strategies, all these fields have had varying degrees of success.

Now, the IT industry has ushered in the fifth computing paradigm: cloud computing. Based on the Internet, cloud computing provides on-demand computing capacity to individuals and businesses in the form of heterogeneous and autonomous services. With cloud computing, users do not need to understand the details of the infrastructure in the "clouds;" they need only know what resources they need and how to obtain appropriate services, which shields the computational complexity of providing the required services.

In recent years, the research and application of parallel transportation management systems (PtMS), which consists of artificial systems, computational experiments, and parallel execution, has become a hot spot in the traffic research field.[2,3] Here, the term *parallel* describes the parallel interaction between an actual transportation system and one or more of its corresponding artificial or virtual counterparts.[4]

Such complex systems make it difficult or even impossible to build accurate models and perform experiments, so PtMSs use artificial transportation systems (ATS) to compensate for this defect. Moreover, ATSs also help optimize and evaluate large amounts of traffic-control strategies. Cloud computing caters to the idea of "local simple, remote complex" in parallel traffic systems. Such systems can take advantage of cloud computing to organize computing experiments, test the performance of different traffic strategies, and so on. Thus, only the optimum traffic strategies will be used in urban-traffic control and management systems. This helps enhance urban-traffic management system performance and minimizes the system's hardware requirements to accelerate the popularization of parallel traffic systems.

Agent-Based Traffic Management Systems

Agent technology was used in traffic-management systems as early as 1992, while multiagent traffic-management systems were presented later.[5] However, all these systems focus on negotiation and collaboration between static agents for coordination and optimization.[6–8] In 2004, mobile agent technology began to attract the attention of the transportation field. The characteristics of mobile agents—autonomous, mobile, and adaptive—make them suitable to handling the uncertainties and inconstant states in a dynamic environment.[9] The mobile agent moves through the network to reach control devices and implements appropriate strategies in either autonomous or passive modes. In this way, traffic devices only need to provide an operating platform for mobile traffic agents working in dynamic environments, without having to contain every traffic strategies. This approach saves storage and computing capacity in physical control devices, which helps reduce their update and replacement rates. Moreover, when faced with the different requirements of dynamic traffic scenes, a multiagent system taking advantage of mobile agents will perform better than any static agent system.

In 2005, the Agent-Based Distributed and Adaptive Platforms for Transportation Systems (Adapts) was proposed as an hierarchical urban-traffic-management system.[10] The three layers in Adapts are organization, coordination, and execution, respectively. Mobile agents play a role as the carrier of the control strategies in the system.

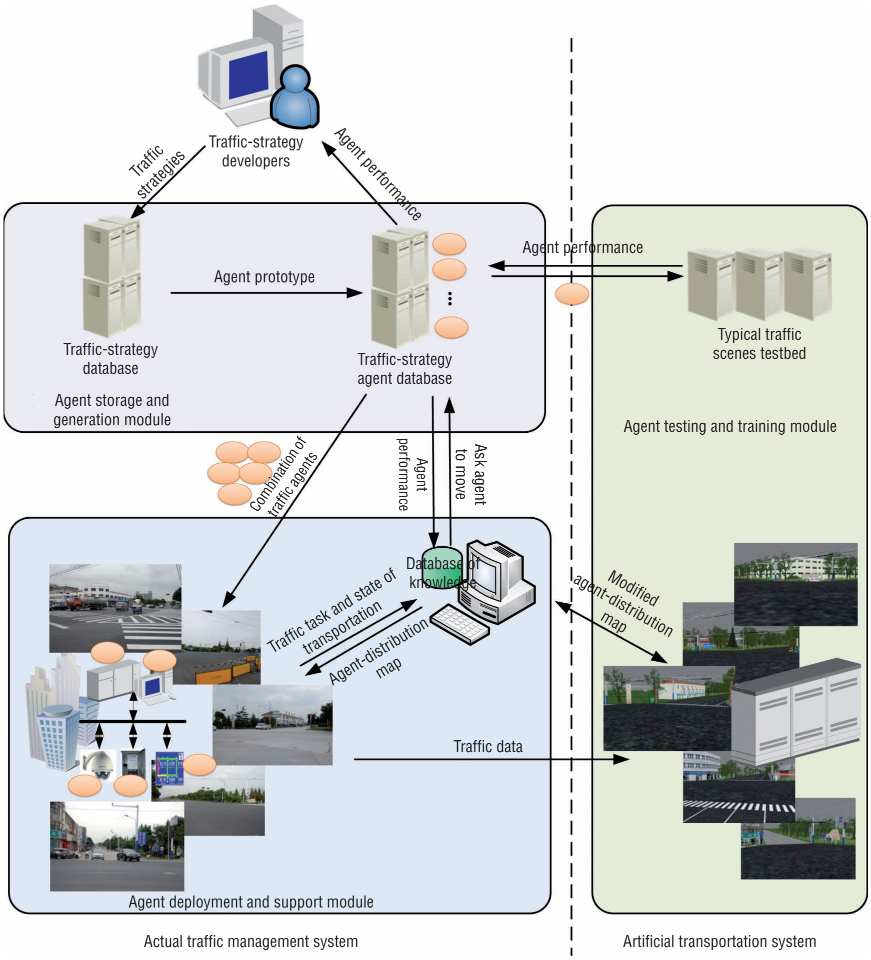

Figure 2. Overview of the main functional elements in the organization layer of Adapts.

In the follow-up articles, both the architecture and the function of mobile traffic control agents were defined clearly. The static agents in each layer were also depicted in detail. What's more, a new traffic signal controller was designed to provide the runtime environment for mobile agent.

Currently, Adapts is part of PtMS, which can take advantage of mobile traffic strategy agents to manage a road map. The organization layer, which is the core of our system, has four functions: agent-oriented task decomposition, agent scheduling, encapsulating traffic strategy, and agent management (see Figure 2). The organization layer consists of a management agent (MA), three databases (control strategy, typical traffic scenes, and traffic strategy agent), and an artificial transportation system. As one traffic strategy has been proposed, the strategy code is saved in the traffic strategy database. Then, according to the agent's prototype, the traffic strategy will be encapsulated into a traffic strategy agent that is saved in the traffic strategy agent database. Also, the traffic strategy agent will be tested by the typical traffic scenes to review its performance. Typical traffic scenes, which are stored in a typical intersections database, can determine the performance of various agents. With the support of the three databases, the MA embodies the organization layer's intelligence.

The function of the agents' scheduling and agent-oriented task decomposition is based on the MA's knowledge base, which consists of the performances of different agents in various traffic scenes. If the urban management system cannot deal with a transportation scene with its existing agents, it will send a traffic task to the organization layer for help. The traffic task contains the information about the state of urban transportation, so a traffic task can be decomposed into a combination of several typical traffic scenes. With knowledge about the most appropriate traffic strategy agent to deal with any typical traffic scene, when the organization layer receives the traffic task, the MA will return a combination of agents and a map about the distribution of agents to solve it. This way, this system takes advantage of the strategy agent to manage a road map.

Lastly, we set up an ATS to test performance of the urban-traffic management system based on the map showing the distribution of agents. ATS is modeled from the bottom up, and it mirrors the real urban transportation environment.[11] Because the speed of the computational experiments is faster than the real world, if the performance is unsatisfactory, the agent-distribution map in both systems will be modified.

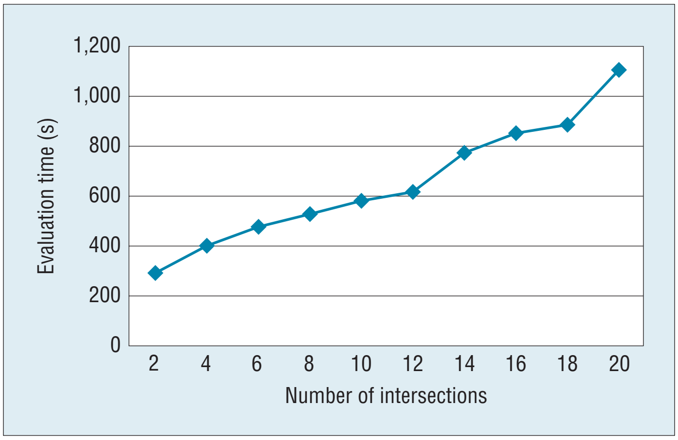

Figure 3. The time required to run ATS and Adapts experiments on one PC.

New Challenges

During the runtime of Adapts, we need to send the agent-distribution map and the relevant agents to ATS for experimental evaluation, so we tested the cost of this operation. In our test, traffic-control agents must communicate with ATS to get traffic-detection data and send back lamp-control data. Both running load and communication volumes increase with the number of intersections. If the time to complete the experimental evaluation exceeds a certain threshold, the experimental results become meaningless and useless. As a result, the carry capacity for experimental evaluation of one PC is limited.

In our test, we used a 2.66-GHz PC with a 1-Gbyte memory to run both ATS and Adapts. The experiment took 3,600 seconds in real time. The number of intersections we tested increased from two to 20, and Figure 3 shows the time cost of each experiment. When the number of traffic-control agents is 20, the experiment takes 1,130 seconds. If we set the time threshold to 600 seconds, the maximum number of intersections in one experiment is only 12. This is insufficient to handle model major urban areas such as Beijing, where the central area within the Second Ring Road intersection contains up to 119 intersections. We would need several PCs or a high-performance server to handle the experimental scale of several hundreds of intersections.

Furthermore, a complete urban-traffic management system also requires traffic control, detection, guidance, monitoring, and emergency subsystems. To handle the different states in a traffic environment, an urban-traffic management system must provide appropriate traffic strategy agents. And to handle performance improvements and the addition of new subsystems, new traffic strategies must be introduced continually. So future urban-traffic management systems must generate, store, manage, test, optimize, and effectively use a large number of mobile agents. Moreover, they need a decision-support system to communicate with traffic managers. A comprehensive, powerful decision-support system with a friendly human-computer interface is an inevitable trend in the development of urban-traffic management systems. Thus, future systems must have the following capabilities.

Computing Power

The more typical traffic scenes used to test a traffic-strategy agent, the more detailed the learning about the advantages and disadvantages of different traffic strategy agents will be. In this case, the initial agent-distribution map will be more accurate. To achieve this superior performance, however, testing a large amount of typical traffic scenes requires enormous computing resources.

Researchers have developed many traffic strategies based on AI. Some of them such as neural networks consume a lot of computing resources for training in order to achieve satisfactory performance. However, if a traffic strategy trains on actuator, the actuator's limited computing power and inconstant traffic scene will damage the performance of the traffic AI agent. As a result, the whole system's performance will deteriorate. If the traffic AI agent is trained before moving it to the actuator, however, it can better serve the traffic management system.

Rational traffic decisions and distributions of agents need the support of ATS, which primarily use agent-oriented programming technology. Agents themselves can be humans, vehicles, and so on. To ensure ATS mirrors real urban transportation, we need large computing resources to run many agents.

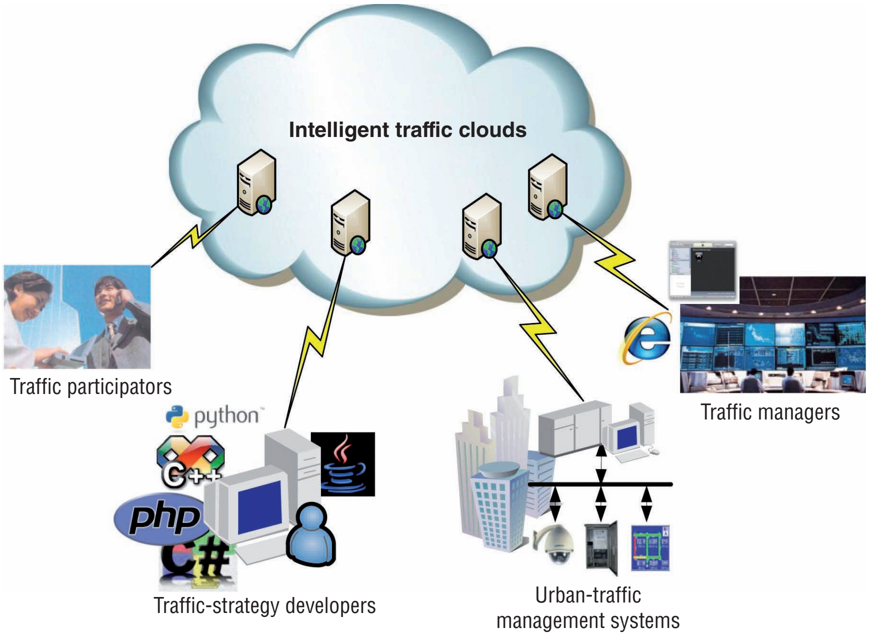

Figure 4. Overview of urban-traffic management systems based on cloud computing.

Storage

Vast amounts of traffic data such as the configuration of traffic scenes, regulations, and information of different types of agents in ATS need vast amounts of storage. Similarly, numerous traffic strategy agents and relative information such as control performances about agents under different traffic scenes also consume a lot of storage resources. Finally, the decision-support system requires vast amounts of data about the state of urban transportation.

Two solutions can help fulfill these requirements:

- Equip all centers of urban-traffic management systems with a supercomputer.
- Use cloud computing technologies—such as Google's Map-Reduce, IBM's Blue Cloud, and Amazon's EC2—to construct intelligent traffic clouds to serve urban transportation.

The former both wastes social resources and risks insufficient capacity in the future. On the contrary, the latter takes advantage of the infinite scalability of cloud computing to dynamically satisfy the needs of several urban-traffic systems at the time. This way we can make full use of existing cheap servers and minimize the upfront investment of an entire system.

Intelligent Traffic Clouds

We propose urban-traffic management systems using intelligent traffic clouds to overcome the issues we've described so far. With the support of cloud computing technologies, it will go far beyond other multiagent traffic management systems, addressing issues such as infinite system scalability, an appropriate agent management scheme, reducing the upfront investment and risk for users, and minimizing the total cost of ownership.

Prototype

Urban-traffic management systems based on cloud computing have two roles: service provider and customer. All the service providers such as the test bed of typical traffic scenes, ATS, traffic strategy database, and traffic strategy agent database are all veiled in the systems' core: intelligent traffic clouds. The clouds' customers such as the urban-traffic management systems and traffic participants exist outside the cloud.

Figure 4 gives an overview of urban-traffic management systems based on cloud computing. The intelligent traffic clouds could provide traffic-strategy agents and agent-distribution maps to the traffic management systems, traffic-strategy performance to the traffic-strategy developer, and the state of urban traffic transportation and the effect of traffic decisions to the traffic managers. It could also deal with different customers' requests for services such as storage service for traffic data and strategies, mobile traffic-strategy agents, and so on.

With the development of intelligent traffic clouds, numerous traffic management systems could connect and share the clouds' infinite capability, thus saving resources. Moreover, new traffic strategies can be transformed into mobile agents so such systems can continuously improve with the development of transportation science.

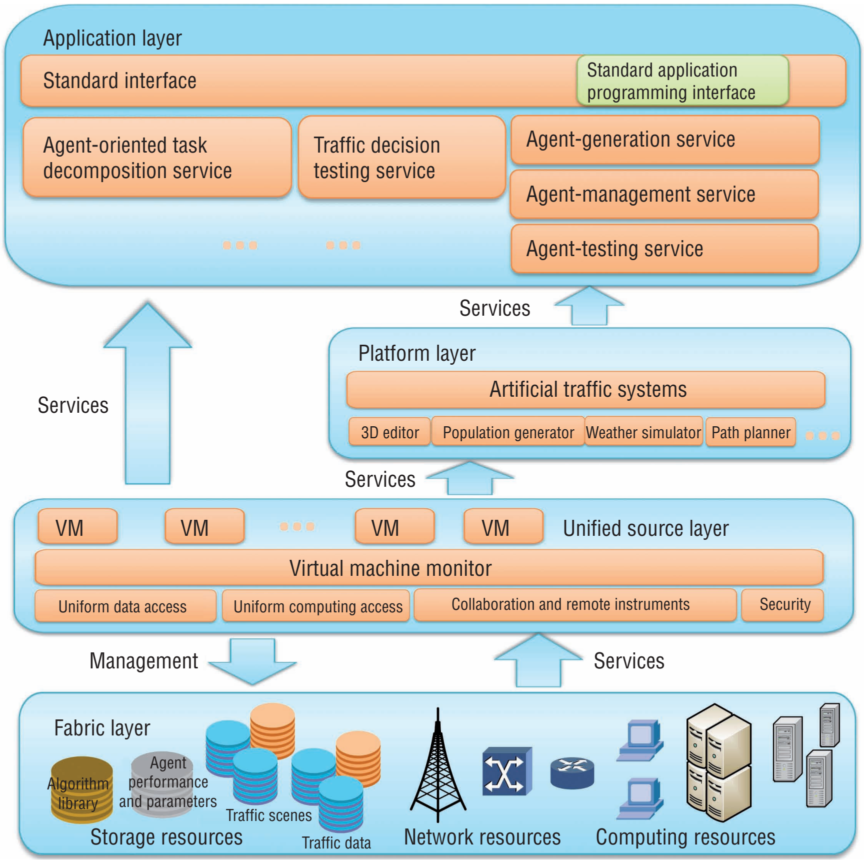

Figure 5. Intelligent traffic clouds structure has application, platform, unified source, and fabric layers.

Architecture

According to the basic structure of cloud computing,[12] an intelligent traffic clouds have four architecture layers: application, platform, unified source, and fabric. Figure 5 shows the relationship between the layers and the function of each layer.

The *application layer* contains all applications that run in the clouds. It supports applications such as agent generation, agent management, agent testing, agent optimization, agent-oriented task decomposition, and traffic decision support. The clouds provide all the services to customers through a standard interface.

The *platform layer* is made of ATS, provided platform as a service. This layer contains a population synthesizer, weather simulator, path planner, 3D game engine, and so on to provide services to upper traffic applications and agent development.

The *unified source* layer governs the raw hardware level resource in the fabric layer to provide infrastructure as a service. It uses virtualization technologies such as virtual machines to hide the physical characteristics of resources from users to ensure the safety of data and equipment. It also provides a unified access interface for the upper and reasonable distribute computing resources. All those will help solve information silo problems in urban traffic and help fully mine useful information in the traffic data.

Lastly, the *fabric layer* contains the raw hardware level resources such as computing, storage, and network resources. The intelligent traffic clouds use these distributed resources to cater the peak demand of urban-traffic management systems, support the running of agents and ATS test beds, and efficiently store traffic strategy agents and their performances.■

Acknowledgments

We would like express our appreciation to Fei-Yue Wang for his guidance and encouragement. This work was supported in part by NSFC 70890084, 60921061, 90924302 and 90920305.

References

1. D.C. Gazis, "Traffic Control: From Hand Signals to Computers," *Proc. IEEE*, vol. 59, no. 7, 1971, pp. 1090–1099.
2. F.-Y. Wang, "Parallel System Methods for Management and Control of Complex Systems," *Control and Decision*, vol. 19, no. 5, 2004, pp. 485–489.
3. F.-Y. Wang, "Toward a Revolution in Transportation Operations: AI for Complex Systems," *IEEE Intelligent Systems*, vol. 23, no. 6, 2008, pp. 8–13.
4. F.-Y. Wang, "Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 3, 2010, pp. 1–9.
5. B. Chen and H. H. Cheng, "A Review of the Applications of Agent Technology in Traffic and Transportation Systems," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 2, 2010, pp. 485–497.
6. M.C.Choy, D.Srinivasan and R.L.Cheu, "Cooperative, Hybrid Agent Architecture for Real-Time Traffic Signal Control," *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 33, no. 5, 2003, pp. 597–607.
7. M.C.Choy, D.Srinivasan, and R.L.Cheu, "Neural Networks for Continuous Online Learning and Control," *IEEE Trans. Neural Networks*, vo1. 7, no. 6, 2006, pp. 1511–1531.

8. B.P. Gokulan and D. Srinivasan, "Distributed Geometric Fuzzy Multiagent Urban Traffic Signal Control," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 3, 2010, pp. 714–727.
9. N. Suri, K.M. Ford, and A.J. Cafias, "An Architecture for Smart Internet Agents," *Proc. 11th Int'l FLAIRS Conf.*, AAAI Press, 1998, pp. 116–120.
10. F.-Y. Wang, "Agent-Based Control for Networked Traffic Management Systems," *IEEE Intelligent Systems*, vol. 20, no. 5, 2005, pp. 92–96.
11. F.-Y. Wang and S. Tang, "Artificial Societies for Integrated and Sustainable Development of Metropolitan Systems," *IEEE Intelligent Systems*, vol. 19, no. 4, 2004, pp. 82–87.
12. I. Foster et al., "Cloud Computing and Grid Computing 360-Degree Compared," *Proc. Grid Computing Environments Workshop*, IEEE Press, 2008, pp. 1–10.

ZhenJiang Li is an assistant professor at the Key Laboratory of Complex Systems and Intelligence Science, Chinese Academy of Sciences. Contact him at lzjwhb@gmail.com.

Cheng Chen is a PhD student at the Key Laboratory of Complex Systems and Intelligence Science, Chinese Academy of Sciences. Contact him at chengchen.cas@gmail.com.

Kai Wang is a PhD student at the College of Mechatronics Engineering and Automation, the National University of Defense Technology (NUDT), Changsha, Hunan, China. Contact him at yogikai@163.com.