

ISEE: INTERNET-BASED SIMULATION FOR EARTHQUAKE ENGINEERING PART II: THE APPLICATION PROTOCOL APPROACH

Kung-Juin Wang, Shiang-Jung Wang, Wei-Choung Cheng,
Yuan-Sen Yang, and Keh-Chyuan Tsai

National Center for Research on Earthquake Engineering, Taipei, Taiwan

*kjwang@ncree.gov.tw, sjwang@ncree.gov.tw, wccheng@ncree.gov.tw, yuansen@ncree.gov.tw,
kctsai@ncree.gov.tw*

Abstract

A project, Internet-based Simulation for Earthquake Engineering (ISEE), had been launched in National Center for Research on Earthquake Engineering (NCREE) to develop techniques of networked collaborative structural experiments. This paper gives description to a Platform for Networked Structural Experiments (PNSE) as one of the two approaches under the ISEE project. Via the Internet, PNSE links a numerical simulation program and a number of facility control programs geographically scattered around the world by point-to-point TCP connections to collaboratively conduct a single experiment. An application protocol, the Networked Structural Experiment Protocol (NSEP) defining relevant events and activities in real laboratories, as well as communication rules between the client and server, was proposed to work with the PNSE. Results of transnational numerical simulation and real experiments are presented to prove the validness and the efficiency of PNSE.

INTRODUCTION

For structural laboratories to meet the increasing demands on testing large and realistic specimen, a more practical solution is to link a number of laboratories by the Internet to collaboratively perform a single experiment. Researchers in Japan and Korea have conducted some tests jointly to investigate the practicability of pseudo dynamic tests between these two countries (Sugiura et al., 1998; Yun et al., 2000; Watanabe et al., 2001). The NEES (Network for Earthquake Engineering Simulation) project, envisioned by the National Science Foundation of USA, also aims at exploring the tremendous benefits of sharing and integrating laboratory resources via network (National Science Foundation 2000). A project, the Internet-based Simulation for Earthquake Engineering (ISEE), have been launched in National Center for Research on Earthquake Engineering (NCREE) in Taiwan to construct a platform that links a numerical simulation program and a number of facility control programs around the world by the Internet for purposes of networked collaborative structural experiments. Within the framework of ISEE project two approaches, the Database Approach (Yang et al. 2003) and the Application Protocol Approach, provide different platforms to achieve this goal. This paper briefly describes the architectures and practical implementation of the Application Protocol Approach.

The Transmission Control Protocol / Internet Protocol (TCP/IP) suite was designed as an open standard to meet the demand of data transmission on rigorous network conditions (Postel 1981a, 1981b). TCP guarantees reliable data transmission by providing services such as acknowledged delivery, error detection, retransmission if necessary, data sequence preservation, and flow control. IP provides addressing, routing, fragmentation and reassembly for data packets. TCP/IP stack thus handles all those tedious works for data transport between hosts on heterogeneous networks. The characteristic of open standard and the fact of support from almost all currently available operating systems make TCP/IP the foundation of the today's Internet. With TCP/IP, any two hosts on line can communicate to each other without difficulty provided they have the same application protocol, a set of predefined rules describing the information, as well as sending and receiving behaviors of the application. The main objective of the Application Protocol Approach under the ISEE project is to construct a platform which allows geographically distributed structural laboratories and a command generation program to be linked together by employing the TCP/IP suite for the purpose of networked collaborative experiments. A preliminary application protocol which defines information relevant to the structural experiments is proposed in this study for communication on PNSE.

ARCHITECTURE OF PNSE

The architecture of the PNSE is illustrated in Fig. 1. The PNSE concentrates on core works directly related to the progression of collaborative experiments (command generation and execution; signals transmission) and excludes accessory functionalities such as data storage, management, monitoring, and video display, which can be implemented by utilizing commercially available software such as the Microsoft SQL server and the Windows Media Player. Three types of modules on the platform, the PNSE server, Command Generation Module (CGM), and the Facility Control Modules (FCM) are connected by employing socket techniques utilizing the TCP/IP suite to ensure high interoperability between heterogeneous network and working environments.

The PNSE is a multi-client system with two kinds of clients, the CGM and the FCMs, connecting to the server with a TCP connection. For any client, all messages and data must be sent to and received from the server. The reason that clients do not connect directly to each other is to simplify the network topology and hence the communication flow. This star topology makes each client communicate with the server only, instead of the many other clients. This may introduce more time needed to transfer data to its destination, but this design can largely reduce the complexity of the communication flow and hence significantly save programming efforts. Communication between the server and client is full-duplex transmission meaning that either side of a connection can actively send and passively receive information at arbitrary timing and thus to realize the goal of an event reflective platform. The communication follows the rules depicted in the proposed application protocol NSEP detailed later.

The PNSE Server

The PNSE server is essentially the information center of the platform. It provides services of message dispatch and data delivery for clients. Although it is conceptually the center of the star-like network topology, it is not the "processing center" which administers the contents and sequence of works that each client should perform. The server maintains a simple login process for any connection attempt for security concern. It can also store experimental data onto a database program on the Internet for instant test result display and monitoring.

The CGM

The CGM calculates or prepares the command to be imposed on the specimen. It can be a numerical integration algorithm in pseudo dynamic testing, an input module that queues predefined command profile in quasi-static testing, or simply an remote control application with an user interface that allows its user to enter command promptly. The CGM prepares commands for all FCMs and integrates them into a single packet (composite command) to send to the server for parsing and dispatching. It then waits for a specific data packet (critical response, generated and sent by all FCMs, and finally integrated and dispatched by the server) as an indication of the completion of the command execution.

The FCM

The FCM receives the parsed command (individual command) from the PNSE server and controls the actuators to impose the target on the specimen. It then measures or calculates the critical response and sends it back to the server as an indication of the completion of command execution. When all FCMs complete command execution, the server notifies the CGM of this event and the CGM can then calculate and send the command for the next step. The FCM on PNSE is quite the same as the traditional facility control program in a structural laboratory. It still controls the actuator motion, performs the data acquisition, alters the test running state if necessary, and displays the real time test results to its operator. One difference is that the FCM on PNSE executes the command received from the PNSE server, instead of command generated by itself. Another difference is that it is obligated to send the notification to the server when the command execution is done.

In a traditional real laboratory, a test can be suspended, resumed, or even stopped prematurely for various reasons such as safety concern or the need of specimen adjustment and investigation. On PNSE, FCM still has full privilege to change its running state if necessary. However, to the end of an event reflective platform, the FCM is obligated to send a notification to the server to flag any change of its state.

Signals and Command Cycle

Signals on the PNSE are referred to those values that change continuously with respect to time. Any variable values in both CGM and FCMs, such as the structural responses calculated in the CGM, or sensor values measured by all FCMs in the laboratories, are considered signals on the PNSE. On PNSE, signals are divided into two categories: the "critical signals" and the "open signals." Critical signals refer to those signals that are critical to the progress of the collaborative experiment such as the command and restoring force responses in a pseudo dynamic test. They are prerequisites to complete a cycle of command execution and actually guide the progress of the test. The open signals refer to those selected signals that are to be publicized on the Internet. PNSE client publicizes the open signals by sending them to the server, which can in turns store them in a database on the Internet for purposes of elegant storage and real time display. A client can also decide not to publicize any of its signals on the Internet since they do not play critical roles in the progression of the collaborative experiment.

A command cycle is the smallest unit of command execution. It encompasses those actions to be executed before the next command is generated. The whole collaborative experiment is actually done by repeatedly executing a number of command cycles. For example, Table 1 lists actions to be included in a command cycle in a collaborative pseudo dynamic test.

Human Communication

On PNSE, human communication is still necessary but not as easy to implement since all the staffs and operators of PNSE client programs are in different locations around the world. To address this issue, a feature of instant discussion is included in PNSE. After successfully logins onto the server, all clients can actively send human readable texts as a mean of communication. This feature provides convenient conversation channel for human communication and is a necessity since it is almost impossible to define all events that can possibly occur in the course of the test in the application protocol.

NETWORKED STRUCTURAL EXPERIMENT PROTOCOL

The TCP/IP suite provides reliable data transmission without knowing the actual content of the data. It is only responsible for reliable delivery of a stream of bytes over the networks. In stead, the content of the transmitted data should be understood by both sides of a connection and hence applications must have an application protocol defined at the application so that they can communicate with each other. An application protocol, the Networked Structural Experiment Protocol (NSEP), is proposed to work with PNSE. NSEP defines the communication rules and data packets to hold signification information for PNSE. PNSE server and clients communicate to each other by sending and receiving those predefined data packets over the Internet.

The Data Packet

The NSEP data packets contain useful information relevant to the collaborative experiment and are composed of parameters of some defined primitive data types. The composition of NSEP data packets can be expressed as follows.

$$\text{LENGTH} + \text{TYPE} + \text{PARAMLIST} \quad (1)$$

Parameter LENGTH denotes the total length of the whole packet. PNSE employs the TCP connection which preserves data arriving sequence but not the packet boundary. Actually what is transmitted over the network is a stream of bytes and a data packet is simply only a concept. Hence, each data packet should contain explicit information on the total length of itself to indicate its boundary so that the receiving end of a connection can correctly receive it.

Parameter TYPE holds the category or classification of the data or information contained in this packet. Currently defined TYPEs are described at Table 2.

PARAMLIST is an optional list of parameters to make this data packet complete and meaningful. The number and the actual content of those parameters depend on each NSEP TYPE. Detailed description of those defined packets can be found in Tsai et al. 2003.

NSEP is designed intended to make PNSE an event reflective platform. Significant events that can occur in clients and information that is relevant to the progress of the experiment are defined in NSEP. Currently significant event only includes the running state of each client (the SD_CLNSTATE packet) and the consequent running state of the whole project (the SD_PRJSTATE packet). Relevant information currently defined in NSEP includes:

1. General information of this collaborative experiment (the SD_PRJINFO packet).
2. Commands (the SD_CPSCMD and the SD_IDVCMD packets).
3. Critical response (the SD_IDVRSP and the SD_CPSRSP packets).
4. Information and values of open signals (the SD_SIGNALINFO and SD_SIGNAL packets).
5. Human conversation (the SD_DISCUSS packet).

Active Notification

NSEP stipulates that all PNSE modules should take the responsibility of making active notification if it owns the information relevant to the progress of the experiment. For example, if an FCM needs to hold the experiment for any reason, it should notify the PNSE server of this holding action before or immediately after it does it. When the server receives this notification of change of an FCM's running state, it has the responsibility to notify all other clients. NSEP stipulates this active notification to simplify the communication over the networks and to enhance the overall communication efficiency since otherwise an information query mechanism may be needed to make the platform event-reflective. Although it is still feasible, an information query mechanism generally decreases the overall efficiency since typically the query operations need to be performed frequently enough so that events can be reflected promptly enough. However, frequent query waists a lot of time and network resources, especially when the information queried has not been updated at the moment of query.

The concept of "event-driven" is the modern philosophy of programming and is powerful when the sequence of computational works can't be determined by the programmer in the coding stage of the program. This is exactly the case of collaborative experiment and hence the NSEP is developed under the concept of event-driven. Consequently, there exists no "processing center/module" which administers the contents and sequence of works. The PNSE server does not mandate any client to do particular tasks to proceed with the test. The progression of the test is led by continual actions responding to events triggered by all clients. Summarily, for PNSE clients NSEP conceptually stipulates for:

1. Data packets that hold information. Sending a data packet is actually triggering an event.
2. Timing to send a particular data packet.
3. Behaviors that need to be done before sending a data packet. (Or equivalently, behaviors that need to be performed after receiving a data packet.)

The concept of event-driven actually constructs the PNSE a true cooperating platform since the server and all the clients take part of the responsibility of the test progression. In addition, combined with the concept of object-oriented programming, the characteristic of event-driven has great values especially when events and interactions in the system get more and more complex as time goes by.

EXPERIMENTAL VALIDATION

Test Specimen

To verify the feasibility of PNSE, series of networked pseudo dynamic tests were conducted on a bridge system. The system is composed of a rigid superstructure pin-supported by two double-skinned concrete filled steel tube (DSCFT) piers. The schematic drawing and photo of the DSCFT pier is shown in Fig. 2, Fig. 3 and Fig. 4. Detailed description of the design of the DSCFT column can be

found in Tsai et al. 2003. The four-DOF (X and Y displacement at the top of each pier) system is illustrated in Fig. 5. The two piers were constructed and tested in laboratories of National Taiwan University (NTU Lab) and NCREE (NCREE Lab) respectively. In each laboratory, two actuators mounted from the two sides of the L-shape reaction wall imposed the target displacement on the specimen.

Test Program

Three pseudo dynamic tests (Test A, B, and C) were conducted on the specimen. All testing parameters were exactly the same among the three tests except for the locations of the server and the CGM. In Test A, both the server and the CGM programs ran in NCREE Lab on a same computer. In Test B, the server program was resided in NCREE Lab and the CGM was at Stanford University in US. In Test C, the server program was placed in Stanford and the CGM in NCREE Lab. The assigned mass is 507.2 ton for DOF 1 and 3, and 235.6 ton for DOF 2 and 4. The damping ratio is 2% for DOF 1 and 2 and 5% for DOF 3 and 4. The stiffness used in numerical simulation for comparison purpose was obtained by linear regression from a pretest which cyclically drove the specimen within elastic range. Newmark explicit method was employed in the process of numerical integration with integration time step 0.02 seconds and duration 20.0 seconds. In this study, the structural seismic behavior is not the primary concern and hence the ground motion was carefully selected to remain the specimen elastic throughout the tests. Excitation records from 1999 Chi-Chi earthquake (TCU082NS and TCU082EW for X and Y directions, respectively) were used as the ground excitation input with the peak ground acceleration (PGA) scaled to 3.77 gal and 9.13 gal for X and Y directions respectively. The control systems employed in NCREE Lab and NTU Lab were MTS FlexTest IIm and MTS 407 controllers respectively. The FCMs in NCREE Lab and NTU Lab were built by C++ programming language and LabVIEW running under Microsoft Windows 2000 operating system.

Test Results

Comparisons between the displacement time histories of Test A, B, C, and the numerical simulation shown in Fig. 6 indicate that all the experimental responses are in good agreement with the numerical simulation results. This confirms that all signals including commands and critical signals were correctly transmitted over the Internet by PNSE.

Fig. 7 demonstrates the some key actions in a command cycle. The notations are detailed in Table 3. Table 4 lists the average time spent on respective tasks for Test A, B, and C. The Microsoft Ping program was used as a benchmark to investigate the time needed for data transmission between hosts and the results are also included in Table 4. Comparison between the performance of Microsoft Ping and PNSE, it suggests that PNSE has reached a reasonable performance level on the rate of data transferring. A round trip for a data packet between Stanford and Taiwan costs less than 0.1701 seconds on PNSE, which is pretty satisfactory since the time needed for similar task done by the operating system under the same network condition is 0.1608 seconds. Time distribution is illustrated in Fig. 8. As expected, most time was consumed on the actuator motion control. Elapsed time spent on communication on Internet was 1%, 21%, and 32% for Test A, B, and C, respectively. The test results conclude that the efficiency of data transmission on PNSE is satisfactory.

CONCLUSIONS

Transnational test results confirmed that data can be correctly transmitted over heterogeneous computer systems and environments, and significant events can be promptly reflected on PNSE. Efficiency of data transferring is also satisfactory, typically cost less than 0.1701 seconds to make a round trip for a data packet between Stanford and Taiwan. It can be concluded that the proposed PNSE can successfully achieve the goal of networked collaborative structural experiments.

ACKNOWLEDGEMENT

Financial support by National Science Council (NSC91-2711-3-319-200) is acknowledged. The authors would also gratefully acknowledge the laboratory support provided by NCREE and NTU.

REFERENCES

- National Science Foundation (2000), "Network for Earthquake Engineering Simulation (NEES): System Integration, Program Solicitation," Report NSC 00-7, National Science Foundation, US.
- Postel Jon. (1981a), "Transmission Control Protocol – DARPA Internet Program Protocol Specification", RFC-793, DARPA.
- Postel Jon. (1981b), "Internet Protocol – DARPA Internet Program Protocol Specification", RFC-791, DARPA.
- Sugiura, K., Nagata, N., Suzuka, Y., and Watanabe, E. (1998), "Internet Related Structural Testing," Proceedings of The Eighth KKNN Seminar on Civil Engineering, Singapore, 219-224.
- Tsai, K. C., Hsieh, S. H., Yang, Y. S., Wang, K. J., Wang, S. J., Yeh, C. C., Cheng, W. C., Hsu, C. W., and Huang, S. K. (2003), "Network Platform for Structural Experiment and Analysis (I)", Report NCREE-03-21, National Center for Research on Earthquake Engineering.
- Tsai, K. C. and Yeh, C. C. (2003), "Networked Substructure Pseudo Dynamic Tests of Double-Skinned CFT Bridge Piers under Bi-Directional Earthquakes (1)", Technical report NCREE-03-021, National Center for Research on Earthquake Engineering, Taipei, Taiwan. (In Chinese)
- Watanabe, E., Yun, C. B., Sugiura, K., Park, D. U., and Nagata, K. (2001), "On-Line Interactive Testing between KAIST and Kyoto University," Proceedings of The Fourteenth KKNN Symposium on Civil Engineering, Kyoto, Japan, 369-374.
- Yun, C. B., Lee, I. W., Part, D. U., and Watanabe, E. (2000), "Remote Parallel Pseudo-Dynamic Testing on Base-Isolated Bridge Using Internet," Proceedings of The Thirteenth KKNN Symposium on Civil Engineering, Taipei, Taiwan, 87-9.

Table 1: Actions encompassed in a command cycle in a pseudo dynamic test

No.	Action
1	The CGM calculates the displacement command for all FCMs, integrates them into a single packet (SD_CPSCMD) and sends it to the server.
2	The server parses the SD_CPSCMD packet to a number of SD_IDVCMD packets before dispatching them to all FCMs.
3	Each FCM controls its actuators to impose the command on its specimen.
4	Each FCM measures the restoring force response as the (individual) critical signals (SD_IDVRSP) and sends it back to the server.
5	Each FCM measures the open signal values (SD_SIGNAL, if any) and sends them to the server.
6	The server integrates all SD_IDVRSP packets into one composite critical response packet (SD_CPSRSP) and dispatches it to the CGM.
7	The CGM sends the values of open signals (SD_SIGNAL, if any) to the server.
8	The server stores the open signal values on the Internet and dispatches them to all clients.

Table 2: Currently defined NSEP data types

TYPE	Description
SD_ERROR	SD_ERROR is used to make a notification to clients when some level of errors occurred in the previous communication.
SD_LOGIN	SD_LOGIN contains client name and password for login procedure.
SD_PRJINFO	SD_PRJINFO contains information of the project, such as the number of the clients, names of clients, total duration or number of steps of this project.
SD_SIGNALINFO	SD_SIGNALINFO contains information (names and units) of open signals.
SD_PRJSTATE	SD_PRJSTATE contains the current running state of the whole project as well as the running states of all clients.
SD_CLNSTATE	SD_CLNSTATE contains the client's current running state.
SD_CPSCMD	SD_CPSCMD contains composite command generated by CGM.
SD_IDVCMD	SD_IDVCMD contains individual command parsed from SD_CPSCMD for each FCM.
SD_IDVRSP	SD_IDVRSP contains individual critical responses after the FCM finishes executing the SD_IDVCMD previous received from the server.
SD_CPSRSP	SD_CPSRSP contains composite critical responses integrated by all SD_IDVRSP received from all FCMs.
SD_SIGNAL	SD_SIGNAL contains the current values of open signals.
SD_DISCUSS	SD_DISCUSS contains human readable texts for human conversation.

Table 3: Description of notations in Fig. 7

Notation	Description
T ₁	Time consumed in transferring the SD_CPSCMD packet from CGM to the server.
T ₂	Time consumed in transferring the SD_IDVCMD packet from the server to NTU Lab.
T ₃	Time consumed in NTU Lab's work (mainly the actuator control).
T ₄	Time consumed in transferring the SD_IDVRSP (the restoring force) packet from NTU Lab to the server.
T ₅	Time consumed in transferring the SD_IDVCMD packet from the server to NCREE Lab.
T ₆	Time consumed in NCREE Lab's work (mainly the actuator control).
T ₇	Time consumed in transferring the SD_IDVRSP (the restoring force) packet from NCREE Lab to the server.
T ₈	Time consumed in transferring the SD_CPSRSP packet from the server to the CGM.
T _a	Time consumed in CGM's work (command generation).

T_b	The sum of T_1 and T_8 , representing the time consumed for data transferring between the server and the CGM.
T_c	Time consumed in the server's work.
T_d	The sum of T_2 and T_4 , representing the time consumed for data transferring between the server and the NTU Lab.
T_e	The sum of T_5 and T_7 , representing the time consumed for data transferring between the server and the NCREE Lab.
T_f	The maximum of $(T_d + T_3)$ and $(T_e + T_6)$.

Table 4: Time statistics on PNSE

Test		A	B	C
CGM work time (sec/step)		0.0180	0.0060	0.0153
Data transmission (round trip):	PNSE	0.0080	0.1661	0.1693
CGM ↔ server (sec/step)	Microsoft Ping	<0.001	0.1628	0.1653
Server work time (sec/step)		0.0437	0.0096	0.0459
Data transmission (round trip):	PNSE	0.0037	0.0056	0.1651
NTU Lab ↔ server (sec/step)	Microsoft Ping	<0.001	<0.001	0.1625
NTU Lab work time (sec/step)		0.5512	0.5486	0.6071
Data transmission (round trip):	PNSE	0.0046	0.0039	0.1701
NCREE Lab ↔ server (sec/step)	Microsoft Ping	<0.001	0.0070	0.1608
NCREE Lab work time (sec/step)		0.6356	0.6106	0.6540
Total experiment time per step (sec/step)		0.6634	0.7963	1.0545
Total experiment time (sec)		663.394	796.290	1054.537

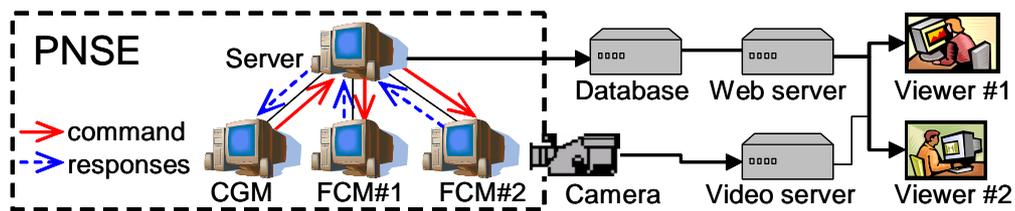


Fig. 1 Architecture of PNSE

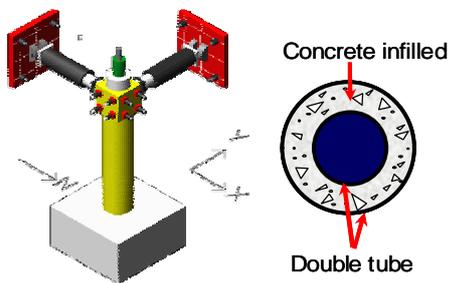


Fig.2 Schematic drawing of DSCFT Fig.3 Test Setup in NTU Lab Fig.4 Test Setup in NCREE Lab

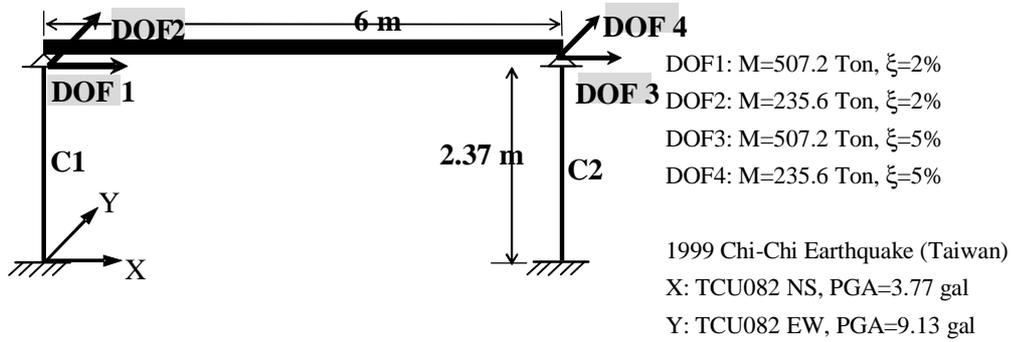


Fig. 5 The simulated bridge system model

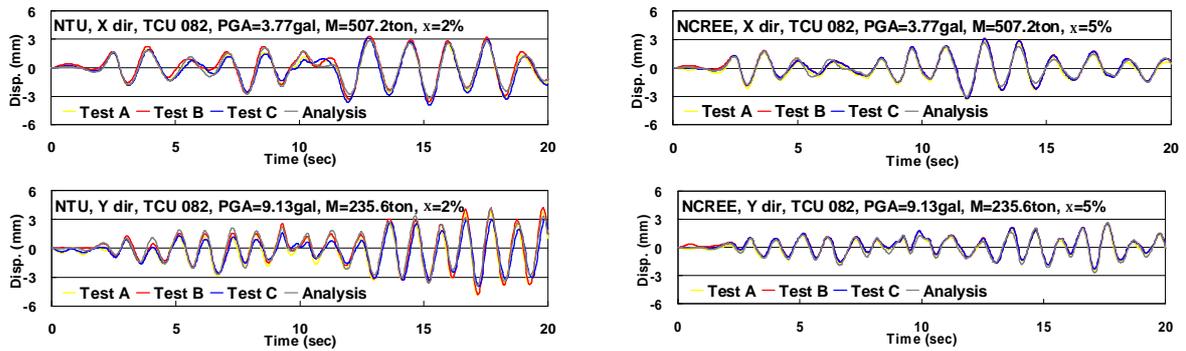


Fig. 6 Experimental and analytical displacement time history

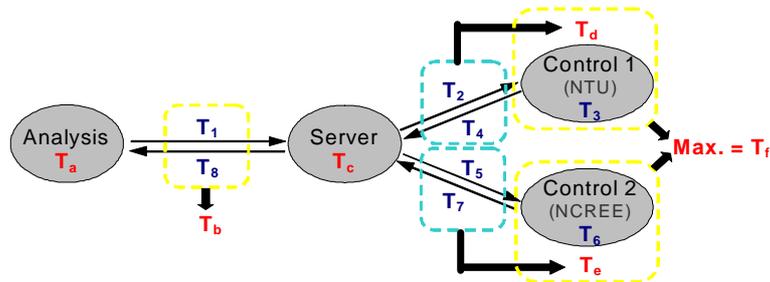


Fig. 7 Time distribution diagram

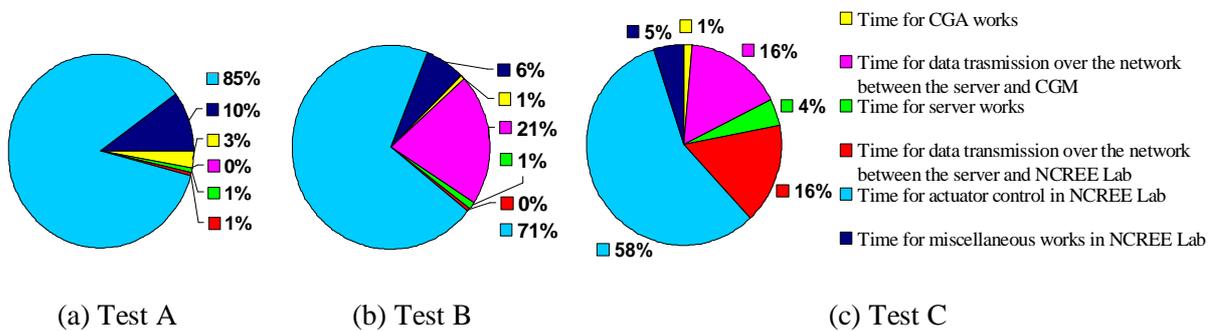


Fig. 8 Proportion of time distribution (measured in NCREE Lab)