ilmedia

TECHNISCHE UNIVERSITÄT ILMENAU

*Miloucheva, Ilka; Simeonov, Plamen L.; Rebensberg, Klaus; Turner, Kenneth J.; Donaldson. A. John M. :*

# *Prototype Performance Evaluation of Multimedia Service Components*

*Beitrag zur:*
3rd International Conference on Computer Communications and Networks, San Francisco, CA, September 11-14, 1994, 8 Seiten

*Ilka Miloucheva, Plamen L. Simeonov, Klaus Rebensberg, Kenneth J. Turner, and A. John M. Donaldson. Prototype performance evaluation of multimedia service components. Contribution to the 3rd International Conference on Computer Communications and Networks, pages 1-8, San Francisco, USA, September 11-14, 1994.*

# Prototype Performance Evaluation of Multimedia Service Components @

*I. Miloucheva[τ], P. L. Simeonov[τ], K. Rebensburg[τ], K. J. Turner[£], A. J. M. Donaldson[£]*

Research Center for Process Control Applications[τ]
Technical University of Berlin
Straße des 17. Juni 136, Sekr. MA 073
D-10623 Berlin, Germany
{ilka,plamen,klaus}@prz.tu-berlin.de

Department of Computing Science[£]
University of Stirling
Stirling, FK9 4LA
Scotland, UK
{kjt,ajd}@compsci.stirling.ac.uk

**Abstract**

This paper deals with a formal approach for decomposition and description of multimedia service components and their performance analysis. Our approach is based on the Temporal Logic of Actions (TLA) specifications. A TLA based specification of multimedia components is transformed into process prototypes described with the SPIMS (SICS Protocol Implementation Measurement System) application language. The multimedia component prototype derived in this way is then evaluated with the SPIMS tool for different QoS parameters.

The proposed approach using TLA based specifications, transformations in SPIMS application prototypes, and performance analysis provides the background for an computer based system for test specification and performance analysis which is currently under development.

We present and discuss practical test scenarios derived from the proposed method for performance analysis of the Audio-Visual Communication (AVC) component of the Joint-Viewing and Tele-Operation Service (JVTOS).

The multimedia test scenarios shown use the TCP/IP and XTP protocols on top of FORE ATM networks.

**Keywords:** Performance Evaluation, Multimedia Services, Temporal Logics, XTP, ATM, SPIMS.

## 1 Introduction

Test and performance evaluation of complex multimedia applications requires a methodology for decomposition of the application into relatively small and deadlock-free parts which have to be investigated with regard to their performance characteristics in order to analyse the QoS provision. Our work focuses on the application of TLA specifications in a multimedia context as proposed by [1] for specification of QoS based test scenarios and their performance analysis, [2], in the framework of the RACE 2088 project TOPIC. This project is concerned with the development of a TOolset for Protocol and advanced service verification in Integrated broadband Communication environment [3]. Related works on decomposition and description of communication service components and their performance analysis are done within projects of the RACE program and refered to as the Brick Wall Model line of broadband services [4], [5], [6].

TLA was introduced by L. Lamport [7] to support the expression of concurrent algorithms, and for reasoning about their computation. Algorithms are modeled on non-deterministic interleaving of events consisting of state changes resulting from single atomic actions. Process-algebraic approaches and conventional FDTs like LOTOS [8] and SDL [9] do not explicitly express liveness properties; and while extensions to both Petri Nets and process algebras to incorporate certain timing and statistical features have been proposed, these are available using TLA.

One of the major premises to apply the TLA formal description technique for test specifications and performance analysis was to structurize the service and investigate for which basic operations of the multimedia application component the use of which protocol options gives the required QoS parameter provision.

In our approach, the TLA actions of the audio-visual service components are transformed into SPIMS application prototypes. Our examples for specification of

@ Proc. of the Third International Conference on Computer Communications and Networks, San Francisco, CA, September 11-14, 1994.

multimedia components are related to the Joint Viewing and Tele-Operation Service (JVTOS), an advanced teleservice developed within the RACE 2060 project CIO, Coordination, Implementation and Operation of Multimedia Teleservices, [10], to support cooperative work over distance by allowing distributed users to work in a collaborative fashion.

The JVTOS CIO service uses the broadband transport protocol over OSI layers 3 and 4, XTPX ([11], [12], [13]), derived from the XTP 3.6. specification [14] with enhancements for provision of the QoS transport requirements in an internetwork environment, controlled access to network resources based on QoS specifications as well as resource and admission control.

The CIO transport system is developed to support QoS provision of multimedia operations in a multinetwork (ATM, FDDI, Ethernet ) and multivendor (Sun, PC, Mac, Silicon Graphics) environment. Our measurements are based on the use of XTPX and TCP/IP on Sun workstations connected with ATM and FDDI networks.

This paper is organized as follows. Section 2 presents a brief overview of TLA specifications and the concept of their transformations in SPIMS prototypes of the Audio-Video Communication (AVC) service. Section 3 describes a sample AVC service scenario and section 4 the TLA description of the video transfer module. Section 5 discusses SPIMS prototype scenarios derived from TLA specifications. Section 6 focuses on performance analysis following these prototypes. Finally, section 7 presents the conclusions and the future plans of this work.

## 2 Transformation of TLA Descriptions into SPIMS Prototypes

A system behaviour in TLA is described in terms of *states* and *state transitions*. System states are represented by *predicates* whereas transitions are expressed by *actions*. Predicates are boolean-valued state functions.

Actions represent the relationship between old (*s*) and new (*s´*) states and as such map pairs of states to booleans.

- $[A]_f$ means : $A \vee (f' = f)$ for any action $A$ and state function $f$.
- $\langle A \rangle_f$ means : $A \wedge (f' \neq f)$ for any action $A$ and state function $f$.

One of the outstanding qualities of TLA is its ability to assert liveness to a behaviour and this is achieved through application of temporal operators to actions.

Liveness properties are expressed in TLA using the concept of *fairness*. We can apply two kinds of fairness to a behaviour.

Weak Fairness: asserts that either $A$ is infinitely often disabled, or infinitely many actions $A$ occur.

$$WF_v(A) \triangleq \Box\Diamond\neg\, Enabled\langle A\rangle_v \vee \Box\Diamond\langle A\rangle_v$$

Strong Fairness: asserts that action $A$ is forever disabled, or infinitely many actions $A$ occur.

$$SF_v(A) \triangleq \Diamond\Box\neg\, Enabled(A)_v \vee \Box\Diamond(A)_v$$

In *temporal* declarations it may be seen that to specify a system, TLA defines an initial stage, predicate *Init* (a boolean-valued state function), followed by some action *Next* (a boolean-valued formula about the state change) and which has the state function (*v*) quoted as a subscript. This action describes all possible legal steps (successive pairs of states, i.e. state changes) to follow in combination with the temporal operator “box” (always). The resulting formula asserts that every step in a behaviour is a legal step, or that the states associated with state function *v* remain invariant.

When this formula is conjoined with the predicate *Init* (ensures that the system starts in the correct state) then the assertion *Safe* can be made so that:

$$\begin{aligned} Safe \triangleq\ & \wedge Init \\ & \wedge \Box[Next]_v \end{aligned}$$

In practice, the formula *Safe* describes what may not happen - the behaviour may not start in an incorrect state, and it may not take an incorrect step; it may however *stutter* (effectively do nothing) forever.

The complete temporal formula (*SpecifyFunction*) is obtained by the addition of a liveness property SF (Strong Fairness) which asserts that something will eventually happen if the correct conditions prevail.

The TLA actions may then be transformed into SPIMS application prototypes[15].

SPIMS prototypes are built based on the following kind of primitive processes [16]:

- *bulk_get*: bulk data transfer read;
- *bulk_put:* bulk data transfer write;
- *req-res*: transactions (request-response)
- *conn_disc*: connect or disconnect.

The following table describes the relation between TLA specifications and SPIMS application prototypes:

| TLA | SPIMS |
|---|---|
| Actions | Primitive Processes |
| Modules | Specification of Process Interaction |
| Predicates | Assertion for call of process structures |
| Parameters | Process Options |
| Temporal Formulas | Temporal Assertions |
| Theorems | None |

Table 1: **Comparison between TLA and SPIMS**

Currently, a special graphical user interface (GUI ) is designed to specify process options and process interaction structures for automatic generation of test prototypes [15]. Figure 1 shows the design of the SPIMS Generator Parameter Help Assistent.

An enhancement of this test generator will allow the generation of SPIMS elements and test suites based on TLA specifications as described in table 1.

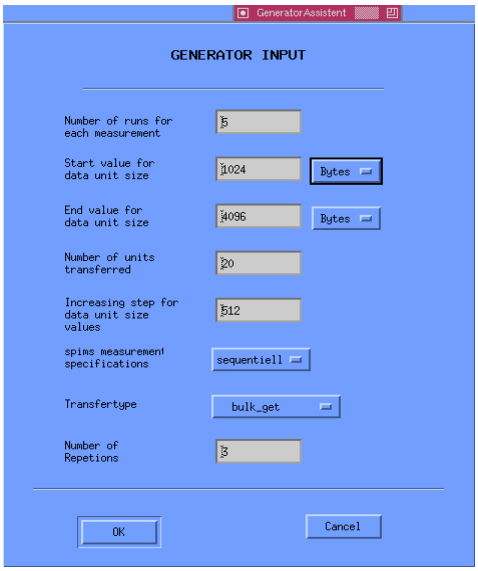



Figure 1: **SPIMS Test Generator Interface**

## 3 An AVC Service Component Scenario

To demonstrate the practical use of the TLA, we selected an example of QoS specification of the AVC component of the multimedia service JVTOS [1], [10]. The goal of this specification is to define QoS measurement points for performance evaluation.

We consider several service elements, such as the service participants, the signal types, the system flags and variables, the QoS values.

Actions are described by predicates reflecting the signal types coming from the action initiation state and leading to the action target state along with their parameters such as transmission path (tp), set of senders (snd), set of receivers (rcv), QoS parameters (qos) and protocol data type.

We assume that the major service participants are :

- *Session Manager* (SM) who has the principal supervisory role and may also function as a user.
- *Resource Allocator* (RA) which is a system module for resource reservation.
- *User(s)* (U) who communicate via the AVC service.

We have to identify the following system *states* : Idle Session, Active Session, Join Active Session, Leave Active Session, Closed Session.

All legal steps can be classified in the following phases and service component functions:

- *Call Set-up* phase (connection establishment at the transport layer) which consists of two functions : Start a Session and Invite Participants/Transmission.
- *Information Transfer* including : User Requests to Join, User Requests to Leave, Association/Removal of participants by the RA and Audio/Visual Data Transfer.
- *Call Clearing* phase (connection release at Transport Layer) has the single function Terminate Session.

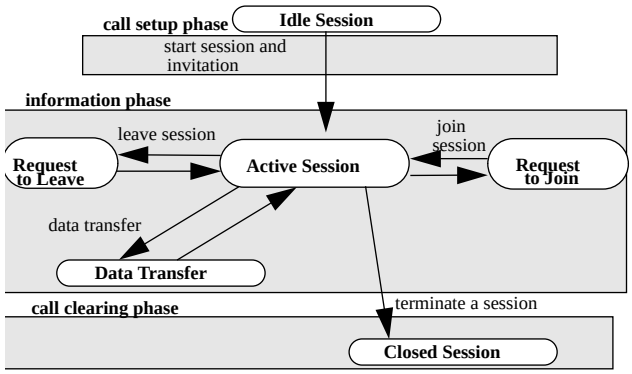


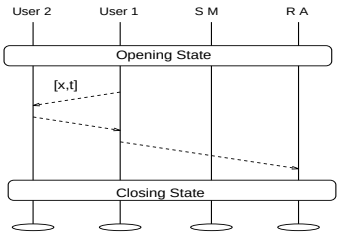

Figure 2: **AVC Service Component Architecture**

Our specification is created using a series of mutually referencing modules, [24], which allows us to import service component functions as shown in figure 2 and thus keep the service structure concise and comprehensive.

## 4 Example: TLA Description of the Video Data Transfer Module

The message sequence chart in figure 3 depicts a Video Data Transfer scenario between 2 users supervised by a System Manager (SM) using a Resource Allocator (RA). The bars indicate the starting and end states of the service component functions which are also referent points for QoS measurement.

We assume a polling policy allowing up to **x** attempts at sending the data, each with a timeout interval of **t**. Here User 1 sends some video data to User 2, and upon receipt of the appropriate confirmation, the originator informs the System Manager and the Resource Allocator about the termination of the video data transfer (the dotted line messages in figure 3).




**SPIMS**
**primitive processes**
req (U2->U1);
bulk_put (U1-> U2);
req (U1-> RA).

Figure 3 : **A Message Sequence Chart for Video Data Transfer (VDT)**

The corresponding module is specified in TLA as shown in figure 4a. Predicates are used to specify necessary attributes associated with each type of signal involved.

The polling signal for video transfer $S_{vt}$ allows a transmission path tp between any two of User 1, User 2 and SM. This is expressed with conjoining tuples of communicating participants. The sender (snd) and receiver (rcv) are declared to be one of these three parties and the polling policy in force (pp) is fixed to x retries with timeout time t. Earlier specification of QoS properties (qos) has been imported via the module SignalTypes.

The expression $Q_b \hat{} \langle Q_v \rangle$ unifies the relevant QoS parameters, where $Q_b$ refers to the "basic" set of QoS parameters at the transport and network layers such as *throughput, delay, delay jitter, error rate, error burstiness,* and $Q_v$ refers to specific video data transfer parameters (at the session and application layers). $Q_v$ is a tuple of the following QoS parameters [13], [23]:

QoSTP - Scalable *Throughput* which is defined as vector of throughput values describing different scales for throughput requirements (scalable video).

QoVDL - The *Quality of Video Delivery Loss* which is defined as the arithmetical mean value of the differences between the sent and received video TPDUs along the measured time interval.

QoVRF - The *Quality of Receiver Video Frequency* which is the rate of displayed video TPDUs on the receiver side.

QoVCE - The *Quality of Video Connection Establishment*, a measure for the video connection establishment time. .

Depending on the type of protocol in use, we may then go on to specify the type of PDU (pdu) that is being sent (call, clear, data, cntl). A similar treatment is given to the response from the other participant with the differences that for the signal $R_{vt}$ we only specify the basic QoS parameters $Q_b$ (this only happens at the transport layer) and make no mention of retries.

This module goes on to specify the signal $I_x$ used to inform the Resource Allocator of this event. This signal is an informative one; it has no retry requirements - only the information QoS parameter in has been specified.

The module ends with a statement that the video data transfer function has a PPUniDirectionalStream (imported via module SessionStatus) and requires the three signals to be sent.

In this specification, modules are imported into other modules higher up in the scheme where we are able to assert that the state transitions taking place are legal within the required context. In TLA the individual modules such as VideoDataTransfer in this example are imported through other modules with other session details and the state function ($v$) defining the system states associated with each function.

The TLA specification is completed with the module SessionFunction shown in figure 4b where the temporal liveness conditions are asserted.

**module** *VideoDataTransfer*

**import** : *SignalTypes, SessionStatus*

**predicates**

$$User \triangleq U \in \{U1, U2, U3\}$$

$$\begin{aligned} S_{vt}(Vid1) \triangleq\ & \wedge tp = \langle U, U\rangle \vee \langle SM, U\rangle \vee \langle U, SM\rangle \quad \% \ TransPath \\ & \wedge snd \in \{U \cup \{SM\}\} \quad \% \ Sender \\ & \wedge rcv \in \{U \cup \{SM\}\} \quad \% \ Receiver \\ & \wedge pp = [x, t] \quad \% \ PolingPolicy \\ & \wedge qos = Q_b \hat{} \langle Q_v\rangle \quad \% \ QoS \ Parameters \\ & \wedge pdu = \text{"data"} \quad \% \ PDU \ Type \end{aligned}$$

$$\begin{aligned} R_{vt}(Vid2) \triangleq\ & \wedge tp = \langle U, U\rangle \vee \langle SM, U\rangle \vee \langle U, SM\rangle \\ & \wedge snd \in \{U \cup \{SM\}\} \\ & \wedge rcv \in \{U \cup \{SM\}\} \\ & \wedge qos = Q_b \\ & \wedge pdu = \text{"data"} \end{aligned}$$

$$\begin{aligned} I_x(Vid3) \triangleq\ & \wedge tp = \langle U, RA\rangle \vee \langle SM, RA\rangle \\ & \wedge snd \in \{U \cup \{SM\}\} \\ & \wedge rcv \in \{U \cup \{SM\}\} \\ & \wedge qos = Q_b \hat{} \langle in\rangle \\ & \wedge pdu = \text{"cntl"} \end{aligned}$$

$$\begin{aligned} fn(VDatTfr) \triangleq\ & \wedge TfrType = \text{"PPMultiDirectionalStream"} \\ & \wedge \langle S_{vt}(Vid1),\ R_{vt}(Vid2),\ I_x(Vid3)\rangle \end{aligned}$$

Figure 4a: **TLA Video Data Transfer**

**module** *SessionFunction*

**import** *FunctionList, Signalling*

**predicate**

$$\begin{aligned} Init \triangleq\ & \wedge systate = \text{"stable"} \\ & \wedge qm = \langle \text{"off"}, \text{"off"}, \text{"off"}, \text{"off"}\rangle \\ & \wedge SessionActive \\ & \wedge \neg FunctionActive \end{aligned}$$

**action**

$$\begin{aligned} Next \triangleq\ & \wedge Transmit \\ & \wedge qm = \langle \text{"on"}, \text{"on"}, \text{"on"}, \text{"on"}\rangle \end{aligned}$$

**temporal**

$$\begin{aligned} SpecifyFunction \triangleq\ & \wedge Init \\ & \wedge \Box[Next]_v \\ & \wedge SF_{\langle v\rangle}(Next) \end{aligned}$$

Figure 4b: **TLA Video Session Function**

Having satisfied ourselves that the state transitions are correct and that we have determined exactly what is required to achieve them in the individual modules, we may then use those modules to make the conversion to the use of SPIMS and so on to our practical evaluation.

# 5 SPIMS Test Scenarios Derived from TLA Specifications

The foregoing section provided a simple example of the nature of the TLA specification and how it may be translated into SPIMS. Here we present further examples of SPIMS application prototypes derived from TLA test scenarios. We may assume that the Session Manager and Resource Allocator are situated at different places/ workstations than the service users.

In the message sequence charts (MSC) below we represent the signalling sequence of several service component functions. All parties are represented as rectangle nodes and the signalling/transfer actions as arrows corresponding to SPIMS primitive processes.

## 5.1 Starting a Session / Inviting Participants

A session is opened by the Session Manager (SM) or a "master" user who sends messages to each user to confirm its willingness to participate in the session and to the Resource Allocator (RA) to reserve a fixed bandwidth (initial state)

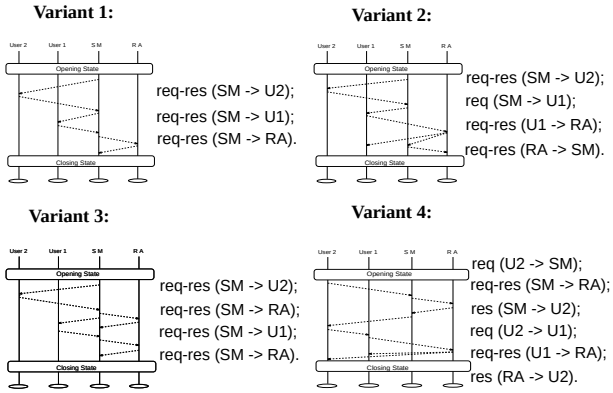



Figure 5: **Session Establishment Test Scenarios**

**Variant 1:** The SM invites users to participate. After a distinct number of users have positively acknowledged the request, the SM asks the RA to reserve bandwidth using the users' response information. The RA responds to the SM either with a confirmation or with a refuse of the session.

**Variant 2:** SM requests a number of users to participate. A "master" user (the last requested) asks the RA to reserve bandwidth for all of the users. If this is possible, the RA informs the SM for taking the users into account and SM responds for beginning the session. Otherwise the session is refused.

**Variant 3:** SM invites a number of users to participate. After each confirmation, SM asks the RA to reserve bandwidth. If this is possible, the corresponding user is involved into the session.

**Variant 4:** The "master" user permitted to open the session requests the SM for session initiation. SM asks the RA about how many users can participate in the session. Afterwards, SM informs User 1 who possibly involves User 2 to request ressources at the RA. Finally, User 1 is informed about the association of the new participant.

## 5.2 Data Transfer (Audio/Video)

Data transfer is specified between two or multiple users.

The "||" sign expresses simultaneous transmission on multiple connections with temporal relationship whereas the semicolon (";") describes simultaneous transmission on connections having no temporal relationship.

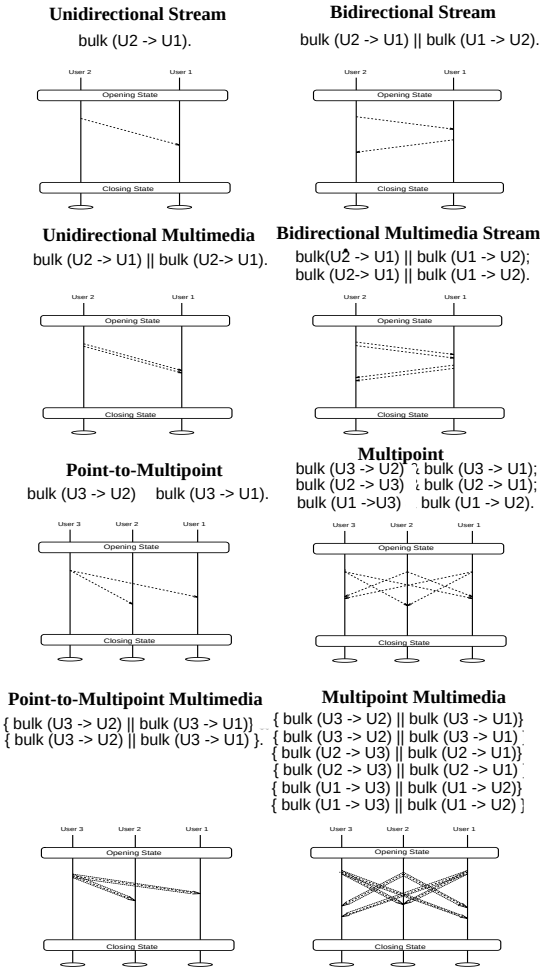



Figure 6: **Video Transfer Scenarios**

### 5.3 User Request to Join or Leave a Session

**Join:** The user requests the SM for participation. After the SM has asked the RA for resource allocation, he acknowledges his participation in the session to the user.

**Leave:** The user makes a request to the SM to leave the session. The SM requests the RA to release the resources for the leaving user.

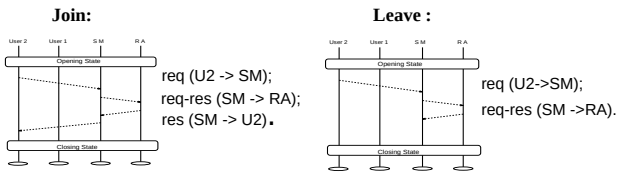



Figure 7: **Join/Leave Scenarios**

### 5.4 Session Termination

**Variant 1:** The master user requests the SM to terminate the session. The SM responds to the other users that the session will be terminated. The SM requests the RA to release the resources for the participants.

**Variant** 2: The master user requests the other users to terminate the session. After their acknowledgments, he/she requests the SM to terminate the session. The SM requests the RA to release the resources.

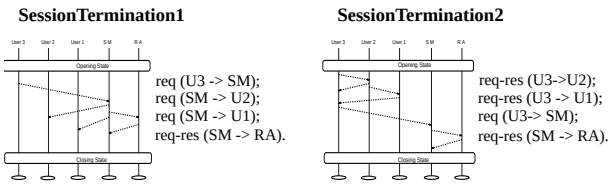



Figure 8: **Session Termination Scenarios**

## 6 Performance Analysis

Within the following performance analysis with the SPIMS tool based on experimental test scenarios, which were described in the previous section, we considered the use of different transport protocol options and parameters for QoS support.

We made measurements between Sun workstations connected with FORE ATM [17] using TCP/IP and XTPX Version1.3 [18] in the testbed environment of the PRZ at the Technical University of Berlin.

We will focus on test scenarios for throughput QoS provision of multimedia applications.

Test scenarios for other QoS parameters supported by the XTPX architecture [13] such as delay, delay jitter, selectable error control are focus of further research. Specific for the throughput QoS parameter of multimedia streams like real time video are such requirements as:

- Threshold values which specify minimum or maximum throughput values which are object of monitoring (if violated the user is informed).
- Scalable throughput specifying values for throughput QoS renegotiation. It allows dynamic change of the throughput QoS parameter during the connection life-time.

### 6.1 Threshold Throughput

We analysed the throughput QoS parameter for different TSDU sizes dependent on the protocol parameters buffer and widow size for point-to-point streams using the TCP/ IP and XTPX protocols.

We observed significant throughput variation when the TSDU size is not well suited to the buffers or the window and rate size. This leads to TSDU peak minimum and maximum throughput values seen in figure 9 which can cause violation of the required threshold values.

Measurement Environment
Sender : Sun Sparc Station1 + (25 MHz, 15.8 MIPS)
Receiver: Sun Sparc Station 2 (40 MHz, 28.5 MIPS)
Network Interfaces : ATM FORE 2.1 Version, AAL 4
Measurement scenario

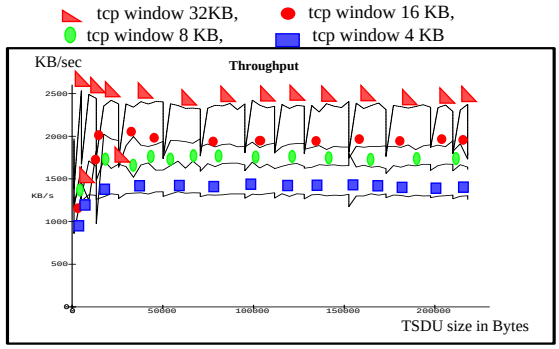



Figure 9: **TCP/IP Throughput Dependence on the TSDU and Window Size**

Because of flexible selection of acknowledgment schemes in XTPX, the throughput variation due to different window and buffer sizes is not so significant (figure 10).

Measurement Environment
Sender : Sun Sparc Station1 + (25 MHz, 15.8 MIPS)
Receiver: Sun Sparc Station 2 (40 MHz, 28.5 MIPS)
Network Interfaces : ATM FORE 2.1 Version, AAL 4
Measurement scenario

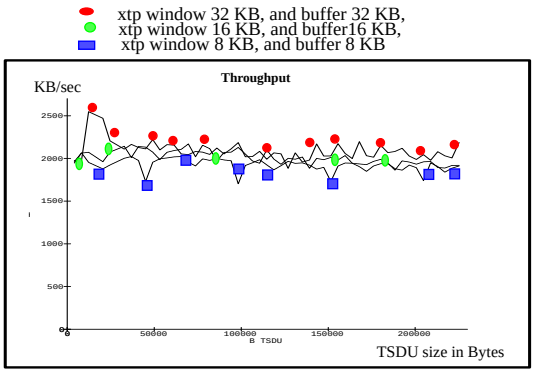



Figure 10: **XTPX Throughput Dependence on the TSDU and Window Size**

In addition, the XTPX rate control mechanisms can provide for better throughput threshold support.

## 6.2 Simultaneous Streams

The next analysis is based on point-to-point multimedia streams. Our focus is to show performance issues when more than one streams are used for simultaneous transfer of video data. The dependence of the throughput of a connection on the throughput of other simultaneous connections assuming the same traffic characteristics using TCP protocol is shown in next figure:

Measurement Environment
Sender : Sun Sparc Station1 + (25 MHz, 15.8 MIPS)
Receiver: Sun Sparc Station 2 (40 MHz, 28.5 MIPS)
Network Interfaces : ATM FORE 2.1 Version, AAL 4
Measurement Scenario
snd_buf = 16 KB, rcv_buf = 16 KB,
TSDU length= 20KB

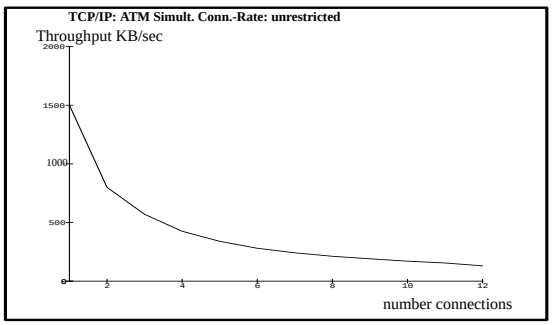



Figure 11: **Dependence of Throughput on the Number of Simultaneous Connections using TCP**

Due to the same flow control strategy used by the TCP connections, the individual connections have the same performance degradation, when more connections are used simultaneously for data transfer. It means that in TCP due to the increasing load the throughput QoS parameter can not be renegotiated flexibly.

Figure 12 shows scenarios for point-to-point multimedia streams based on the XTPX protocol using different requirements for scalable throughput QoS. We measured the throughput of up to 12 simultaneous connections while varying the throughput QoS requirement - unrestricted rate, 500 KB/s, 200 KB/s and 20 KB/s.

Measurement Environment
Sender : Sun Sparc Station1 + (25 MHz, 15.8 MIPS)
Receiver: Sun Sparc Station 2 (40 MHz, 28.5 MIPS)
Network Interfaces : ATM FORE 2.1 Version, AAL 4
Measurement scenario
window = 29120 Bytes, recv buffer = 50 KB,
send buffer = 50 KB, TSDUlength = 20 KB

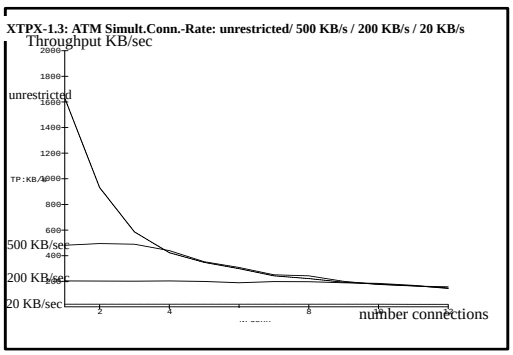



Figure 12: **Dependence of Throughput on the Number of Simultaneous Connections using XTPX**

Using XTPX, the throughput QoS requirement can be scaled dependent on the load, i.e. number of simultaneous multimedia streams. For example when the maximum number of simultaneous connections is 12 , the throughput of each stream is restricted to 200 Kb/s.

## 7 Summary and Further Research

We presented a method to describe measurements of multimedia service components based on TLA specifications translated into SPIMS application prototypes.

We selected TLA for the following reasons:

- It is able to provide broad descriptions of behaviours by allowing us to reason about the actions and state transitions that take place in an easily comprehensible, concise and yet straightforward manner.
- TLA has been shown [19] to be well suited to handle real time issues.

- Specifications such as this usually involve long and complicated formulas. However, by a combination of ordinary mathematics and some conventional programming notation, TLA is able to provide a clean presentation.
- TLA handles safety and liveness aspects in a manner not found in other more conventional formal description techniques (FDTs).
- TLA is suitable for the automatic verification of specifications of distributed systems [20], [21]

Further work considers the integration of TLA tools ([20], [21]) within SPIMS$^+$ [15]. Our goal is development of a computer based system for testing and measurement based on TLA specifications and their subsequent translation into SPIMS prototypes. The system will allow:
- A formal specification of SPIMS test prototypes.
- Use of TLA theorem proving tools (based on the Larch Prover [22]) to verify the system relationships specified in TLA and to probe areas of uncertainty.
- Systematization of test prototype scenarios based on formal descriptions.
- Composition of complex multimedia applications in components, and specification of test scenarios for integrated test of particular components.

## Acknowledgements

We would like to acknowledge the discussions with our colleagues involved in CIO and TOPIC transport protocol activities.

## References

[1] A. J. M. Donaldson, "Specification of Quality of Service Measurement Points in JVTOS", *Master Thesis*, University of Stirling, September, 1993.

[2] I. Miloucheva, P. L. Simeonov, "Performance Evaluation of Multimedia Service Components", Proc. of the *First Workshop on Architecture and Implementation of High-Speed Performance Communication Systems*, Karlsruhe, Jan., 1994.

[3] P. L. Simeonov, J. Burmeister, "Installation Guidelines for the Service and Protocol Verification Demonstrator", *TOPIC, RACE Project 2088, R2088/TUB/PRZ/DS/P/009/b1*, 1993.

[4] RACE Project QOSMIC, "QoS and Performance Relationships", *Deliverable RACE 82/KT/LM/DS/B/010/b1*, June, 1991.

[5] RACE, "QoS Parameters for Videoconferencing", *CFS, Specification D511*, Brussels, 1994.

[6] RACE, "QoS Parameters for File Transfer", Brussels, *CFS, Specification D513*, May, 1994.

[7] L. Lamport, "A Temporal Logic of Actions". *Research Report 57, Digital Equipment Corporation, Systems Research Centre*, 1990.

[8] T. Bolognesi, E. Brinksma, "Introduction to the ISO specification language LOTOS", *Computer Networking and ISDN Systems*, 14(1987), 25-59.

[9] F. Belma, D. Hogrefe, "The CCITT Specification and Description Language SDL", *Computer Networking and ISDN Systems*, 16(1989), 311-341.

[10] B. Plattner et. al, "JVTOS: Requirements on a Joint Viewing and Tele-Operation Service", Rep. *CIO RACE 2060, 60/ETHZ/CIO/DS/P/002/b1*, 1992.

[11] B. Metzler, I. Miloucheva, "Multimedia Communication Platform: Specification of the Broadband Transport Protocol XTPX", Rep. *CIO RACE 2060, 60/TUB/CIO/DS/A/002/b2*, 1993

[12] I. Miloucheva, "Specification of Enhanced Protocol Facilities for Multicast and Broadcast", Rep. *CIO R2060/TUB/CIO/DS/P/003/b1*, June,1994.

[13] I. Miloucheva, K. Rebensburg, "QoS based Architecture using XTP", *4th IEEE Conference on Future Trends of Distributed Systems*, Lisboa, September, 1993.

[14] Protocol Engines Inc., "XTP Protocol Definition", *XTP FORUM Revision 3.6*, January, 1992.

[15] A. Weidt, A. Guther, "SPIMS+ A Protocol Engineering and QoS Test Tool", *Workshop on New Protocols for Multimedia Systems*, Berlin, 1994.

[16] SICS, "SICS Protocol Implementation Measurement System (SPIMS)", *Swedish Institute of Computer Science*, User Manual, Feb., 1991.

[17] SPANS, "Simple Protocol for ATM Network Signalling", *Fore Systems Inc*., 1993.

[18] I. Miloucheva, "XTPX Version 1.3", *Technical Documentation*, Technical University Berlin, January, 1994.

[19] M. Abadi and L. Lamport, "An Old-Fashioned Recipe for Real Time" *Res. Report 91, DEC, Systems Res. Centre*, 1992.

[20] P. Hermann, T. Kraatz, H. Krumm, M. Stange, "Automated Verification of Refinements of Concurrent and Distributed Systems", *Tech. Report*, TU Darmstadt, 1993.

[21] M. Stange, "Ein Werkzeug zur automatischen Verification Verteilter Systeme per On-the-Fly Model Checking in TLA", *MSc. Thesis*, TU Darmstadt, 1993

[22] U. Engberg, "TLP: The TLA+ Proof Checker, Release 2.5.A", *University of Aarhus*, Dept. of Computer Science, May, 1994.

[23] P. L. Simeonov, L. Ulrich, "A Method for QoS Verification by Measurement for the JVTOS Video Transfer", *Workshop on Multimedia Applications and Quality of Service Verification*, CRIM, Montreal, June, 1994.

[24] A. J. M. Donaldson, "Formal Specification of QoS Properties", *Workshop on Multimedia Applications and Quality of Service Verification*, CRIM, Montreal, June, 1994.