



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



**CC creative commons**  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

**BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# Generic Performance Management of Multiservice Networks

*Iain Phillips, Mark Sandford, David Parish, Omar Bashir*  
*Department of Electronic Engineering,*  
*Loughborough University,*  
*Leics, LE11 3TU, UK.*  
*{I.W.Phillips, J.M.Sandford, D.J.Parish}@lboro.ac.uk*

## Keywords

Networks, ATM, SMDS, Performance Management, Monitoring, Adaptive Testing.

## 1. Introduction

Performance Management includes the task of determining whether a system's behaviour is meeting a particular target. In applying this to networks we need to know if a network is under utilised or over burdened by traffic load; whether unusual effects are taking place that might affect user traffic; or if operation is satisfactory.

Work at Loughborough University on a BT funded University Research Initiative (URI) has concentrated on Performance Measurement and Management. The aims of the work have been to produce a generic Monitoring Test Architecture that will monitor a multiservice network irrespective of underlying technology.

It is important at this stage to differentiate between performance management and fault management. The latter concentrates totally on monitoring for finding faults in the network, for example damaged hardware. In performance management we are further concerned with the "non-broken" behaviour of the network. For example, delays in a packet network, which may unexpectedly change, yet the network remains operational. The rationale here is that some applications may require a better performance than the specification, or that network providers may need to know how close to a broken-network they are before committing resource in expansion.

## 2. Performance Management

Tasks in performance management can be grouped in to gathering, storing, processing and presenting network performance information, together with a control architecture combining these tasks together.

In order to measure performance, schemes are needed to gather low-level data about network characteristics. To this end, a GPS (Global Positioning System) based timing mechanism was developed, which can provide globally synchronised timings accurate to around 1 microsecond.

A mechanism to store the gathered data has been developed. This is based on the concept of Intermediate Information. This is a general information representation

that is able to service a wide range of user queries efficiently. This infers an ability both to generate multiple statistical results for similar time periods, and also to be able to generate similar summaries over different time periods, without the need for repeated reprocessing of the low-level gathered data.

An Object-Oriented command and control architecture provides for higher-level analysis techniques that have been developed to highlight significant events in the network behaviour. These higher-level performance indicators require the combination of information from several lower-level measures, for example average delay, packet loss ratio etc. Even these measures may be too low-level to provide useful information to network operators. For this reason, work has been undertaken to provide an efficient mechanism for the notification of unusual network behaviour. Note that this could indicate good and bad network characteristics.

AIR (Automatic Incident Reporting) is therefore concerned with identifying incidents or exceptions. The current system uses a rule-based definition to compare recent performance with historical and baseline trends. Any performance that is thought to be different, according to a given criteria, may be raised as an exception.

The control architecture can also be used to control the starting and stopping of tests in response to the results from previous tests. For example, if repeated weekly testing of a route shows that there is an event in the delay graphs at the same time each week then more intensive testing of the link or testing of related links could be started automatically at the appropriate time in the future.

### 3. Conclusions

This paper has discussed various approaches to the development of an integrated and automated network performance measurement tool. Adopting an object-orientated approach to the entire system design can assist this requirement for intelligence in a distributed manner. The authors have found the JAVA language to greatly assist in this task and this language has been used for all aspects of the system from traffic generation/reception, to the database and display systems.

Finally, examples of the systems have been developed and implemented at various levels, from experimental operations on ATM networks, to a prototype operational system on BT's commercial SMDS (Switched Multi-megabit Data Service) network.

### 4. References

- [1] R. Brookes, M. Blattner, Z. Pawlab, E. Barrett. Using Partitional Databases for Statistical Data Analysis. Proceedings of the AFIPS National Computer Conference, 1981, pp453-457.
- [2] O. Bashir, I. Phillips, D. Parish, J. Adams, T. Spencer, Management and Processing of Network Performance Information. BTTJ October 1998, pp 203-212.