Deploying QoS sensitive services in OSGi enabled home networks based on UPnP

Nico Goeminne, Kristof Cauwel,
Filip De Turck, Bart Dhoedt
Ghent University - IBBT - IMEC
Department of Information Technology
Gaston Crommenlaan 8, bus 201
9050 Gent, Belgium

Abstract – *Enabling end-to-end Quality of Service (QoS) in the home network will push emerging services such as iDTV and pay per view to an acceptable and guaranteed user experience. Currently used QoS solutions in the wide area network can not be used in the home network. This home network management becomes a more challenging task since it's changing drastically from a pure PC only to a multi device, multi user network. In view of the complexity and the error proneness to manage these networks, this challenging task should clearly be shifted from the residential user to the service provider and / or the connecting provider .This paper presents a management platform for enabling end to end quality of service implementing the UPnP QoS Architecture specification, based on a dynamic UPnP utility framework for the OSGi Service Platform.*

Keywords: OSGi, UPnP QoS, WiFi

1 Introduction

Broadband internet, WiFi, mobile devices and new exciting services such as interactive Digital Television (iDTV) or pay per view are among the new technologies that are key factors in pushing the intelligent home networks to a new level. A broad audience is becoming aware of the new possibilities and the time where domotica equaled controlling lights or room temperature is long over.

Managing all those technologies and devices can be a daunting task which should not be the responsibility of the end user, but of the devices themselves. Technologies should be kept secure; devices should be self configuring and services interoperable. Service discovery protocols such as Jini or UPnP are in place to facilitate interoperability and self configuring networks. Furthermore a residential or home gateway enables service providers to perform external and additional configurations. The gateway should be extensible so later additions, for example services or software updates, are easily deployed.

In this paper we present an innovative platform that brings a fully integrated end to end solution for deploying QoS in the wireless home network. This platform combines several leading standards and technologies, such as Web Services, the TR69 communication protocol, the UPnP QoS Architecture [1]

and the OSGi Service Platform [2]. The platform is unique in a way that it is automatically deployed on already established OSGi enabled home networks and consists out of a generic UPnP utility framework that supports dynamic deployment, configuration and rapid development by composition of UPnP devices and services for the OSGi Platform. Furthermore a unique Wireless QoS Protocol [3] that can be used in combination with 802.11e [4] has been developed.

The next section describes a use case that shows the need for QoS in the home network and presents the architecture of the end-to-end solution. The third section describes the chosen standards and protocols, while the fourth section describes the UPnP QoS Architecture mapping with our Wireless QoS Protocol. The fifth section presents the UPnP utility framework which facilitates rapid UPnP device development within the OSGi Service Platform.

2 Use case

2.1 Introduction

This use case addresses the autonomous delivery of multimedia content to a home network terminal. Clearly without QoS guarantees, multimedia service execution will be jeopardized. Figure 1 displays the use case setup, showing the main interacting parties.

Content Provider The content provider offers some sort of content in the form of multimedia streams. Important functions such as indexing and streaming requests are exposed to clients.

Remote Management Server The role of the remote management server is to control the residential gateway. It exposes an interface to the content provider for configuring the home network, for example for providing QoS for a certain multimedia stream.

Residential Gateway The residential gateway sits on the edge between the wide area network (WAN) and the local area network (LAN). It can be accessed from both the WAN and LAN side, therefore it is a good candidate to deploy new services, in addition to performing standard functions (Firewall, NAT, etc.).

Home Device Home devices reside in the home network and can be anything from desktop PC's over set-top boxes to PDA's and smart phones.
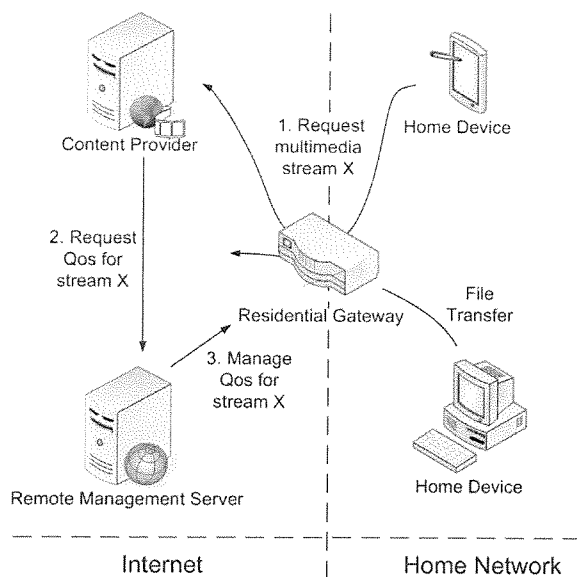
Fig. 1. use case setup

2.2 Operation

When the content provider receives a request to stream a movie, it knows the duration and the bandwidth it needs to provide a good user experience. It asks the remote management server of the residential gateway to reserve the necessary bandwidth. The remote management server analyses the request and forwards it to the residential gateway which is responsible for setting up the QoS within home network. Once the setup is completed and the necessary bandwidth is reserved, the media stream starts to play.

Since the bandwidth is guaranteed for the media stream, other traffic such as a file transfer will not interfere with the stream.
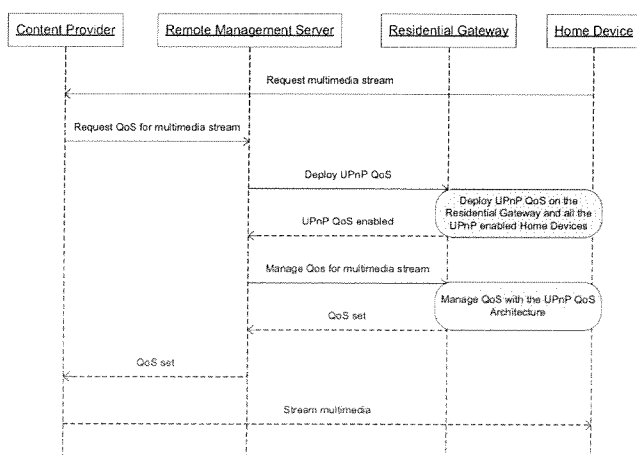
Fig. 2. use case sequence diagram

Figure 2 displays a simplified sequence diagram of the use case. The UPnP QoS architecture as well as the UPnP deployment is described in detail the fourth section.

3 Technical Analysis

It is assumed that each home device supports the UPnP QoS Architecture and that the user uses a standard web browser to select the multimedia stream from a website. Based on that selection the content provider negotiates with the remote management server to get a lease on the required bandwidth for the duration of the stream. The stream can be uniquely identified by the gateway's IP and port number. Furthermore it requests the opening of the gateway's firewall for that port. The negotiation should be standardized for example through web services or TR69 [5], which is a prime candidate to succeed SNMP [6] and is heavily supported by the DSL Forum [7]. TR69 foresees a yet to be defined north-bound interface, especially for those purposes.

Before the remote management server can set the requested bandwidth and firewall rules, it checks if the residential gateway supports UPnP QoS at its LAN side, using TR69. If that's not the case, the UPnP QoS bundles are automatically provivisioned by the management server. This way they can be easily pushed and deployed on the gateway.

QoS on the LAN side is only useful if QoS is also guaranteed at the WAN side too, but that's considered outside the scope of this paper since there are existing solutions.

4 UPnP QoS

4.1 UPnP QoS architecture

In order to standardize the way QoS streams are set up and managed inside a home network, the UPnP forum has released the UPnP QoS architecture specification. The UPnP QoS architecture consists of three elements (services) as shown in figure 3.
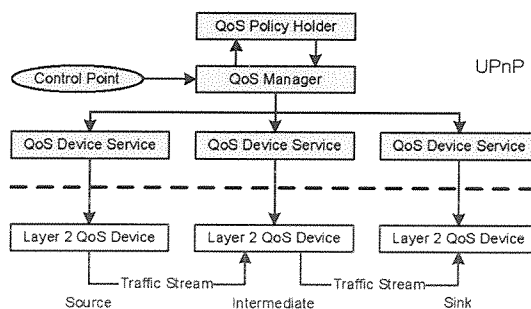

Fig. 3. UPnP QoS Architecture

The QoS Policy Holder is the repository of the QoS policy for a given LAN. It is configured by the user and allocates traffic priorities.

The QoS Manager is the central part of the UPnP QoS framework. All QoS Device and QoS Policy Holder services are discovered and controlled by this Manager. As shown in the figure, QoS requests/releases are sent to the QoS Manager through a Control Point. This Control Point offers users and applications an interface for the QoS Manager. All interaction with the QoS Manager should happen through the Control Point.

The QoS Device service is responsible for managing a network device's resources. So the service provides a link between a QoS capable network device and UPnP.

4.2 UPnP QoS deployment

Deployment of the UPnP involves two steps. First, all devices in the home network need to be discovered, and for each discovered device, the UPnP QoS Device Service should be added. This could be done using OSGi QoS Device bundles using the UPnP utility framework to enhance an existing UPnP Device with the new deployed service. Of course depending on the device type, wired or wireless, different implementations should be pushed. Furthermore special implementations are required for wireless access points and the residential gateway itself. Secondly the control point should be deployed somewhere in the home network.

4.3 UPnP QoS implementation

To test the UPnP QoS architecture in a real life environment, we implemented the model architecture in OSGi. On the residential gateway, an OSGi bundle is deployed which encapsulates a QoS Policy Holder, a QoS Manager and two QoS Devices. A second OSGi bundle implements a QoS Control Point. This Control Point exposes the QoS Manager functions as an OSGi service, so that it can be used by other bundles. Every client in the Residential Network has a QoS Device service.

The QoS Manager keeps track of all QoS streams currently active in the network. It is responsible for setting up new QoS streams and maintaining old ones. This can be updating new QoS streams, or removing them if their QoS lease time expires. Closely connected to the QoS Manager is the QoS Control Point. As stated before, this is an OSGi bundle that offers an interface to other bundles in the OSGi framework. The interface takes user-friendly parameters, such as destination IP address and source IP address of the stream, instead of the UPnP defined XML Traffic Descriptors.

The first Residential Gateway QoS Device sets the correct ToS value in the IP header on the packets received from the internet. This ToS field will be used by other devices in the network to differentiate traffic streams. The QoS Device also implements a Diffserv mechanism for upstream traffic. This, for example, allows feedback packets from the digital TV to skip packets from a big FTP upload.

All other QoS Device services (client QoS Device and second Residential Gateway QoS Device) are mapped on our own Wireless QoS Protocol. This protocol is based on a token

[1] Application request QoS stream
[2] Control point forwards request to QoS Manager
[3] QoS Manager contacts QoS Policy Holder
[4] QoS Policy Holder returns the traffic policy
[5] QoS Manager configures devices on the path
[6] QoS Device configures the 802.11 device
[7] Control Point returns status to application
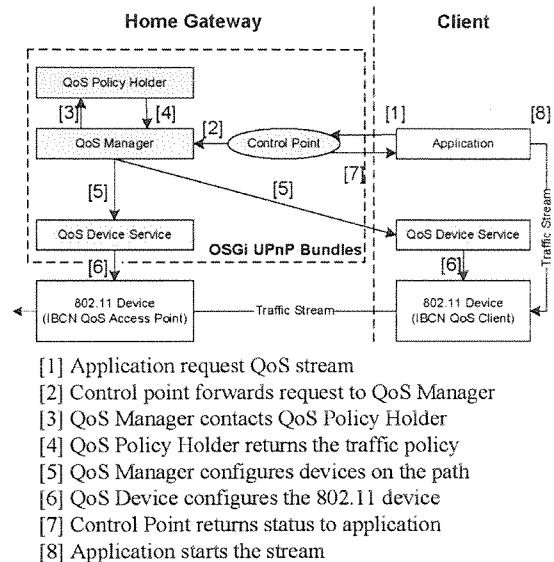[8] Application starts the stream

Fig. 4.   UPnP QoS control flow

bucket system. It enforces bandwidth constraints throughout the whole wireless network. This way bandwidth guarantees can be met and stations cannot saturate the network. The protocol runs on top of Layer 2 and can thus be used in combination with 802.11e.

5 Dynamic UPnP Device composition

5.1 OSGi based UPnP utility framework

The OSGi Service Platform specification offers a model for dynamically composing software applications or services and a recommended section called the UPnP Device Service Specification which defines the UPnP Base Driver [8].

The base driver allows OSGi services to be made available on networks with UPnP enabled devices. By implementing the proper OSGi interfaces one can easily create a UPnP compatible device, yet all implementations share the same structure. There is a root device, some embedded devices, some services, state variables and actions. Only the actions contain non standard methods to be invoked when the action is called.

When a UPnP device is implemented and deployed, it cannot be altered or dynamically composed anymore and thus losing the key success factor for the OSGi Service Platform. For example, if a device does not support UPnP QoS, it would have to be completely reengineered because there's no easy way to add the UPnP QoS Service to the already existing and deployed device.

Therefore, an architecture was developed to allow for dynamic UPnP device updates. The architecture will be discussed in the next section.

5.2 UPnP utility framework architecture

To optimally reuse and benefit from the OSGi Service Platform a UPnP utility framework was developed. The framework works in combination with the OSGi UPnP specification. The utility framework contains two packages.

The first one extends the OSGi UPnP specification interfaces by adding setter methods. These new interfaces allow more flexibility and post deployment configuration of the UPnP device or UPnP service. The defined interfaces are backward compatible so that implementations can be used transparently by the base driver.

Thanks to the new interfaces, UPnP devices and services become very dynamic. Devices can be updated with new services at run time. Or services can be extended by adding new actions. This allows a user to manage his UPnP devices in a flexible way. Adding or removing devices, services and actions requires no new UPnP device implementation.
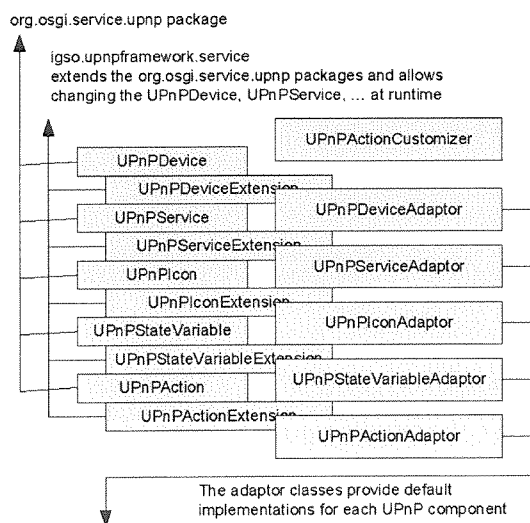
To create a UPnP Device :
  1. Instantiate or select an already in the OSGi framework available instance of an Adaptor class,
  2. Group statevariables and actions into services,
  3. Group services, icons and embedded devices into devices.

To change the default behavior of an action, either :
  - Extend the UPnPActionAdaptor class and overwrite the invoke method
  - Set the UPnPActionCustomizer object for the UPnPActionAdaptor

Fig. 5. UPnP utility framework

Secondly, since the structure of a UPnP device or service doesn't change, it makes good sense to provide adaptor classes that implement the defined interfaces. The adaptor classes can be used unmodified or certain methods can be overridden to specify special behavior. Especially the UPnPActionAdaptor

class which is responsible for the action invocation behavior should be extended. The default behavior returns the default value for the UPnP State Variables (if any). This means that a UPnP Device can be created without actually having to implement the device's actions.

A second way to implement actions is specifying a UPnPActionCustomizer which will only have the invoke method and is used in the same way as the serviceChanged method within the ServiceTracker and ServiceTrackerCustomizer model. Multiple actions can be supported by only one customizer, centralizing the specific device service behavior in one place.

Each UPnP Device, Service, Icon, State Variable or Action could be implemented and deployed in a loosely coupled way. Existing technologies, like ServiceTracker, ServiceBinder [9] or Declarative Services can be used to bind the different parts. Consider, for example, having a bundle that implements the UPnP QoS Service. By deploying and exporting that service in the OSGi framework registry, already deployed UPnP Devices can pick up and integrate the new QoS service, without having to be redesigned or reconfigured.

5.3 UPnP device generation

The architecture describe above allows for easy and fast UPnP device implementation. Because this implementation is straight forward, the process can be automated. We developed a UPnP device generator to make UPnP device development even faster.

The generator uses the UPnP XML device description to generate Java source code. This source code can be compiled and packaged directly into an OSGi bundle. The generated code is reduced to a minimum: only an Activator and one InvokeCustomizor per UPnP Service are generated.

If a user wishes to add functionality to his UPnP Device, he will have to alter the generated InvokeCustomizor code. This is very easy, since there is one block per UPnP Action. So to implement these actions, the user would just fill in the corresponding code blocks.

6 Conclusion

In this paper we presented an end-to-end solution for bringing quality of service to the wireless home network, based on the combination of state of the art technologies, including our unique Wireless QoS Protocol. These technologies were exploited to their fullest potential to obtain a truly dynamic and automated user friendly platform. This platform guarantees a zero touch configuration and a guaranteed user experience.

Furthermore, our UPnP utility framework can be used to rapidly develop dynamic extendable UPnP devices and services for the OSGi Service Platform.

Finally, an OSGi based UPnP utility framework was developed to assist in dynamic composition of UPnP Services and Devices.

7 Future work

The UPnP utility framework can be further equipped with a policy manager which manages how devices are composed and what UPnP Services should be included in which devices.

8 Acknowledgements

References

[1] Universal Plug and Play Forum: UPnP QoS Architecture 1.0, March 2005 - http://www.upnp.org.

[2] OSGi Alliance: OSGi Service Platform Release 4, March 27, 2003

[3] Haerick W., Goeminne N., Cauwel K., De Jans G., De Turck F., Dhoedt B., Demeester P., Bracke S., Acke W. , Bouchat C.: Success in Home Service Deployment: Zero-touch or chaos? *The Journal of The Communications Network* , (July-September 2005).

[4] IEEE 802.11e: MAC Enhancements for Quality of Service (draft version).

[5] DSLHome Technical Workgroup: TR-069 CPE WAN Management Protocol, May 2004.

[6] RFC 1098, A Simple Network Management Protocol (SNMP).

[7] DSLHome - DSLForum:www.dslforum.org

[8] Domoware Project : domoware.isti.cnr.it

[9] Cervantes H.,Hall R. S. : Automating Service Dependency Management in a Service-Oriented Component Model, *in 6th ICSE Workshop on Component-Based Software Engineering*, May 2003.

ICOMP'06

The 2006 International Conference on Internet Computing

Zhan Liu, Mi Lu

Session: WEB-SERVICES, INTERFACES AND LANGUAGES

Language Model Grammar Conversion
Wesley Holland, Julie Baca, Dhruva Duncan, Joseph Picone

Developing Interoperable Software For Differing XML Schemata
Thuy-Linh Nguyen

Techniques for Handling JSF Exceptions, Messages and Contexts
Dwight Deugo

Index and Search XML Documents by Combining Content and Structure
Faïza Abbaci, Jean-Baptiste Valsamis, Pascal Francq

A Distributed API for Searching Multimedia Databases
Dale Parson, Lisa Frye

Automatic Web Service Composition Based on Service Interface Description
Jing-zhou Zhang, Shou-jian Yu, Xiao-kun Ge, Guo-wen Wu

A Framework for Collaborative Applications using Web Services
Ricardo Coppo, Claudio Delrieux

Session: USABILITY ASPECTS AND VISUALIZATION

A Methodology for Structured Use-Centered Quantitative Full-Life-Cycle Usability Requirements Specification and Usability Evaluation of Web Sites
Guoqiang Hu, Kai H. Chang

The Early Stages of TA Spoken Shared Internet Browser for the Blind
Emad Eldin Mohamed, Hesham Kamel

A Study on the Screen Designs of E-Learning Material
Hiroshi Ichikawa, Kaoru Honda, Hiroo Hirose, Yoshito Yamamoto

Proposal of Web usability by comparison experiment of paper media and internet browser that considers individual characteristics
Hiroo Hirose, Hiroshi Ichikawa, Yoshito Yamamoto

Can We Improve Web Accessibility for Users with Below Average Cognitive Capacity?
Jenny Jerrams-Smith, David Heathcote

Session: SENSORS, NETWORKS AND COMMUNICATION

Hybrid WDM and Optical Spectral Amplitude Coding with Array Waveguide Gratings over Fiber-to-the-Home Network
Che-Chih Hsu, Jen-Fa Huang, Yao-Tang Chang

A Web-based Server Management System with IPMI and WMI Techniques
Yi-Hsuan Yeh, Yu-Chin Szu, Yuan-Cheng Lai

Distributed Sensor Networks based on Hybrid Communication Model
Mohammed Ketel

Session: 4TH WORKSHOP ON XML TECHNOLOGY AND APPLICATIONS – XML TECH'06

Efficient Filtering System to accelerate XML Access Control Enforcement

Changwoo Byun Byun, Seog Park

On-Line News Management System Based on ASP and XML

Chen Hua, Tang Chunyan, Yang Yuexin, Cai Yan, Fang Zhoui, Zhao Rujia, Joan Lu

Virtual Enterprise Oriented Enterprise Strategies and Cooperative Product Design System Development Based on J2EE MVC Mode

Cao Yan, Chen Hua, Yang Lina, Liu Ning, Zhao Rujia, Joan Lu

The Application of XML in the Denotation of Plant Diseases and Insect Pests Information

Qingming Fan, Hua Chen, Yan Cao, Baolong Liu, Yuexin Yang, Bo Sun, Rujia Zhao, Joan Lu

Semantic Enhanced Self-Configuring Grid Framework

Hao Li, Gehao Lu, Joan Lu, Shaowen Yao

Examining Analysis and Evaluation System Based on XML

Chen Hua, Lv Qingli, Cao Yan, Liu Baolong, Yuan Li, Zhao Rujia, Joan Lu

XQuery as a Spatial Query Language

Xia (Lisa) Li

The Application of XML Parsing Technology in E-Government

Gang Xue, Lili Zhang, Rujin Yang, Hao Li, Shaowen Yao

XML Security in Certificate Management

Joan Lu, Nathan Cripps, Hua Chen

Agent Mediated SOA with XML Framework for Grid Computing

Gehao Lu, Hao Li, Joan Lu, Shaowen Yao

XML-Based Security – A New Challenge to I.T. Protection

J. Lu, U. Rahman, Jim Yip

Session: MOBILE MULTIMEDIA + WIRELESS CONTENT DELIVERY ISSUES

Modelling Service Delivery System for Wireless Handsets using Instant Alert and Frequent Scanning Technologies: An Approach for Bus Service Application

Johnnes Arreymbi, Abdullah Al-Zakwani

Mobile Devices Evolution and Revolution: A Cause for Security Concern

Abdullah Al-Zakwani

Interactive Design and Delivery Challenges for Wireless Handheld Multimedia Systems

Johnnes Arreymbi, Esther Gachanga

Session: INTERNET COMPUTING TOOLS AND ANALYSIS

Tor – A Tool to Disguise Internet Communications

Victor Clincy, Padmaja Mudiraj

Session: PROCEEDINGS OF CONFERENCE ON COMPUTER GAMES DEVELOPMENT

The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing

Monte Carlo Resort, Las Vegas, Nevada, USA
June 26–29, 2006

Conferences:

The 2006 International Conference on Bioinformatics & Computational Biology

The 2006 International Conference on Computer Design & International Conference on Computing in Nanotechnology

The 2006 International Conference on Computer Graphics & Virtual Reality

The 2006 International Conference on Communications in Computing

The 2006 International Conference on Scientific Computing

The 2006 International Conference on Data Mining

The 2006 International Conference on e–Learning, e–Business, Enterprise Information Systems, e–Government, & Outsourcing

The 2006 International Conference on Engineering of Reconfigurable Systems & Algorithms

The 2006 International Conference on Embedded Systems & Applications

The 2006 International Conference on Foundations of Computer Science

The 2006 International Conference on Frontiers in Education: Computer Science & Computer Engineering

The 2006 International Conference on Grid Computing & Applications

The 2006 International Conference on Artificial Intelligence

The 2006 International Conference on Internet Computing & International Conference on Computer Games Development

The 2006 International Conference on Wireless Networks

The 2006 International Conference on Information & Knowledge Engineering

The 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognition

The 2006 International Conference on Machine Learning; Models, Technologies & Applications

The 2006 International Conference on Modeling, Simulation & Visualization Methods

The 2006 International Conference on Parallel & Distributed Processing Techniques & Applications & International Conference on Real-Time Computing Systems & Applications

The 2006 International Conference on Pervasive Systems & Computing

The 2006 International Conference on Security & Management

The 2006 International Conference on Software Engineering Research & Practice & International Conference on Programming Languages and Compilers

The 2006 International Conference on Semantic Web & Web Services

WORLDCOMP'06

June 26 - 29, 2006

Las Vegas, Nevada, USA

PDPTA/RTCOMP'06
SERP/PLC'06
BIOCOMP'06
IPCV'06
MSV'06
CGVR'06
ERSA'06
CDES/CNAN'06
ESA'06
ICOMP/CGD'06
SWWS'06
ICAI'06

MLMTA'06
SAM'06
DMIN'06
IKE'06
CSC'06
GCA'06
ICWN'06
PSC'06
CIC'06
FECS'06
EEE'06
FCS'06

Editor: H. R. Arabnia

University of Georgia, GA, USA