Modeling and Analysis of General Internet Signaling Transport Protocol (GIST) using Coloured Petri Nets

Atul Kumar
Lecturer(CSE & IT Dept.)
Baba Banda Singh Bahadur Polytechnic College
Fatehgarh Sahib(Punjab),
INDIA

Kanwalvir Singh Dhindsa
Associate Professor (CSE & IT Dept.)
Baba Banda Singh Bahadur Engineering College
Fatehgarh Sahib(Punjab),
INDIA

***Abstract:* - The NSIS (Next Steps in Signaling) working group has standardized a *General Internet Signaling Transport (GIST)* as the base protocol component of NSIS protocol stack to support a variety of signaling applications. The GIST basically provides the Routing and Transport service to the Upper Layer. GIST is not designed to set up or modify paths itself; therefore it is complementary to protocols like RSVP (Resource Reservation Protocol) rather than an alternative. The main aim of this paper is to use Coloured Petri Nets to model the basic working of the GIST protocol i.e a simple GIST example. Initial analysis refers that a basic model is constructed using Coloured Petri Nets and its proper working is verified.**

***Key-Words:* - Quality of Service, NSIS, RSVP, GIST, NTLP, Formal Verification, Coloured Petri Nets (CPN)**

I INTRODUCTION

Signaling refers to the information exchange concerning the establishment and control of a connection and the management of the network in the area of telecommunication and networking. For example, to set up Internet telephone calls, Session Initiation Protocol is used but the protocol is not responsible for transporting the voice data. In internet, Quality of Service has a very important role. Quality of Service (QoS) is defined in terms of providing service differentiation and performance assurance for Internet applications [30]. Resource reservation control mechanism is the main requirement for the achievement of Quality of Service. QoS provides different priority to different users. It guarantees a certain level of performance to a data flow in accordance with requests from the application program on the ISP (Internet Service Provider) Policy.

Various QoS Frameworks were developed to provide quality of Service. The first QoS framework that has been standardized by the IETF (Internet Engineering Task Force) is IntServ (Integrated Services), which uses for QoS signaling support, the Resource Reservation Protocol (RSVP). Another QoS framework that has been standardized by the IETF is DiffServ (Differentiated Services) Due to the shortcomings of RSVP and its current extensions, an alternative extensible signaling approach, Cross-Application Signaling Protocol, or CASP – is introduced for ensuring modularity, flexibility and security. This approach enables to effectively support generic IP signaling that can be used for various signaling scenarios, with enhanced protocol flexibility. The NSIS working group reused many ideas from CASP and standardized a General Internet Signaling Transport (GIST) as the base protocol component of NSIS protocol stack to support a variety of signaling applications.

With the aim of to rule out invalid actions, protocol designers subject their designs to validation. A general manner to do this is to build a software model of the protocol and simulate a large figure of usage situations. The model is performed on virtual devices in a simulated environment. Various techniques are available to test the protocol but one of the best ways is to use formal verification techniques. Formal methods encompass a variety of modeling techniques based on mathematics, which are applicable to computer systems [4]. They are useful in the construction and maintenance of complex communication protocols and allow protocol specifications to be formally analyzed and verified. Formal methods have already been applied to protocol engineering activities and mostly in communication protocols but have been seldom applied to the Internet protocol engineering activities [15] [16]. A wide range of formal methods have been developed [4]. *Coloured Petri Nets (CPN)* is a formal technique with a solid mathematical foundation which has been used for modelling many systems such as communication protocols [18].

In this paper, Coloured Petri Nets Tool is used to model and verify the working of GIST Protocol. Basic model of GIST is constructed with the aid of CPN tool in which general features and functionality of GIST protocol are included. The paper has been organized as follows. Section two presents the basic overview of RSVP. Section three includes a detailed explanation of GIST, its functions, operations and comparison with other protocols. A description of the CPN model of GIST in C-Mode is analyzed in fourth section. All assumptions and requirements and simulation results are also presented in this section. Finally, the conclusion is given in section five.

II OVERVIEW OF RSVP

RSVP is a resource reservation signalling protocol that is designed to be applied in an end-to-end communication path. It can be used by an application to make its quality of service (QoS) requirements known and reserve resources in all the network nodes in the path. RSVP has not enjoyed the level of deployment that might have been expected.

RSVP suffers from many limitations such as lack of scalability, lack of fragmentation and reliability. So a new protocol suite NSIS was developed in which RMD model is used to provide the Quality of Service which overcomes the limitations of RSVP Protocol.

A. Framework of RSVP

First Major framework which provides quality of service is IntServ. IntServ [24] is a per-flow based QoS framework with dynamic resource reservation. Its fundamental philosophy is that routers need to reserve resources in order to provide quantifiable QoS for specific traffic flows. RSVP is a protocol specified to mainly work with the IntServ framework. RSVP [25] serves as a signaling protocol for application to reserve network resources. To support a QoS application, RSVP is designed to be run on network routers and in end hosts [23]. RSVP requests resources for simplex flows. Therefore, RSVP treats a sender as logically distinct from a receiver, although the same application process may act as both a sender and a receiver at the same time [22]. Figure 1 shows the signaling scenario of RSVP.

Receiver-initiated reservation style is adopted by RSVP which is designed for a multicast environment and accommodates heterogeneous receiver service needs. RSVP works as follows [25]:

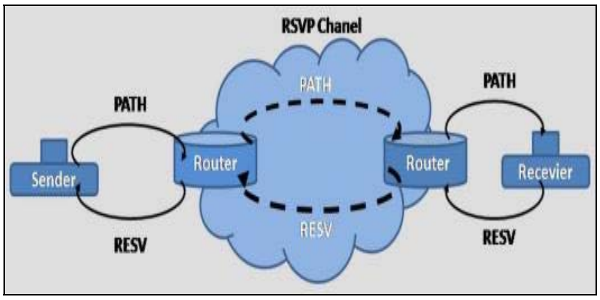

Fig 1: RSVP Signaling [22]

The flow source sends a PATH message to the intended flow receiver, specifying the characteristic of the traffic. As the PATH message propagates towards the receiver, each network router along the way records path characteristics such as available bandwidth. Upon receiving a PATH message, the receiver responds with a RESV message to request resources along the path recorded in the PATH message in reverse order from the sender to the receiver. Intermediate routers can accept or reject the request of the RESV message. If the request is accepted, link bandwidth and buffer space are allocated for the flow, and the flow-specific state information is installed in the routers. Reservations can be shared along branches of the multicast delivery trees.

B. Limitations of RSVP

RSVP has certain limitations which are as following::

- lack of fragmentation causing limited length of the transport units and lower link resource utilization
- Reliability problems due to the use of IP or UDP as transport layers, for the transport of the messages, instead of using e.g., TCP. The message delivery is assured only by retransmissions. This imposes constraints on the signaling
- lack of support for network mobility, which is one of the biggest problems currently in the wireless and ad-hoc networks in particular
- discovery and signaling message delivery are combined in one step which does not allow RSVP to make use of the available security solutions for Internet

To overcome these limitations IETF proposed a new framework, DiffServ, which contains NSIS protocol suite. NSIS's RMD-QOSM protocol provides the quality of service to the internet.

III THE GIST (GENERAL INTERNET SIGNALING TRANSPORT PROTOCOL)

The lower layer in the NSIS architecture defines a common protocol that all kind of signaling applications can use. Application specific functionality is given by the signaling protocols that form the upper NSIS layer. The main protocol used by NTLP to provide the transport of signaling messages is the General Internet Signaling Transport (GIST). GIST must be present if an upper layer NSIS protocol needs to be supported by a node. If some node on the sender-receiver path is not GIST enabled, then all NSIS messages are considered to be ordinarily data packets.

A. GIST Terminology

GIST terminology is given below and is shown in Fig 2 [10]:

Data flow**:** A set of packets identified by some fixed combination of header fields. Flows are unidirectional (a bidirectional communication is considered a pair of unidirectional flows).

Session**:** A single application layer flow of information for which some state information is to be manipulated or monitored. It is identified with a Session ID (SID) parameter

Sender**:** The node in the network which is the source of the packets in a flow. Could be a host, or a router (e.g. if the flow is actually an aggregate).

Receiver: The node in the network which is the sink for the packets in a flow.

Downstream**:** In the same direction as the data flow.

Upstream**:** In the opposite direction to the data flow.

Adjacent peer**:** The next node along the data path, in the upstream or downstream direction, with which a GIST node explicitly interacts. The GIST peer discovery mechanisms implicitly determine whether two nodes will be adjacent

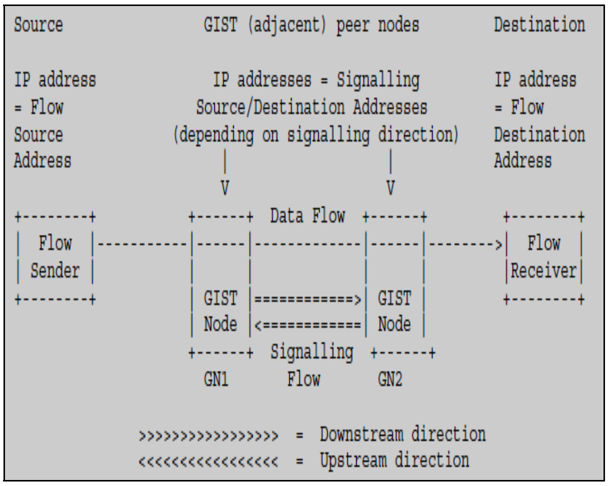

Fig 2: GIST Terminology [10]

B. Working of GIST

GIST has two major goals – one to provide routing and second, transportation of signaling messages. The *routing* determines how to reach the adjacent peer along the data path, and can be done independently for each direction of the connection. Two NTLP states are used for the routing – a routing state, used in the forwarding of the messages, and a message association state, used to relate incoming messages to a particular saved session. A *Message Association* is a connection between two explicitly identified GIST adjacent peers and a message, arriving from the signaling application, is connected to an established message association via the SID parameter of the message header.

Transportation is the delivery of signaling information from peer to peer. The signaling message delivery is divided in two transport modes, the Datagram Mode (D-mode) and the Connection Mode (C-mode). The *Datagram Mode (D-mode)* sends GIST messages between nodes without using any transport layer state or security protection and uses UDP encapsulation. The *connection Mode (C-mode),* on other hand, sends GIST messages directly between nodes using by default TCP as transport protocol. The choice of mode depends on the routing state and on the requirements coming from the signaling application. The GIST lower layer and the application layer on top of it communicate vie the interface, or API, defined between them. The primitives passed at the interface are of three groups – reliability or what transportation mode is desired; security or what security mode is required; and local processing or what special processing has to be done like prioritization, etc.

The messages generated at the GIST layer are [10]:

- ***GIST-Query messages*** are used in the first phase of the discovery procedure, 3-way handshake. It is always sent in datagram mode and leads to creation of routing state for the flow and also message association state if necessary.
- ***GIST-Response message*** is used in the second phase of the handshake and can be sent in datagram or connection mode. If a message association is needed but it is not created this is accomplished during this phase.
- ***GIST-Confirm message*** is the last phase of the discovery procedure an also can be in datagram or connection mode. If connection mode is used a message association must be established during the transfer of the previous two message types.
- ***GIST-Data message*** is used to encapsulate all messages coming from the NSLP layer.
- ***GIST-Error message*** reports errors occurring at the GIST level.
- ***GIST-MA Hello message*** is used to keep a message association state.

When a connection is to be established first the GIST discovery procedure is started. The procedure is between two peers and uses the GIST messages Query, Response and Confirm. As result the peers on the data path sender – receiver are discovered. Subsequently the signaling NSLP messages and the data are encapsulated in GIST Data messages. The GIST discovery procedure can be combined with the NSLP signalization to establish connection.

C. Routing State and Messaging Association Creation

A complete sequence of message exchange for GIST Routing state and Messaging Association is shown in Figure 3.

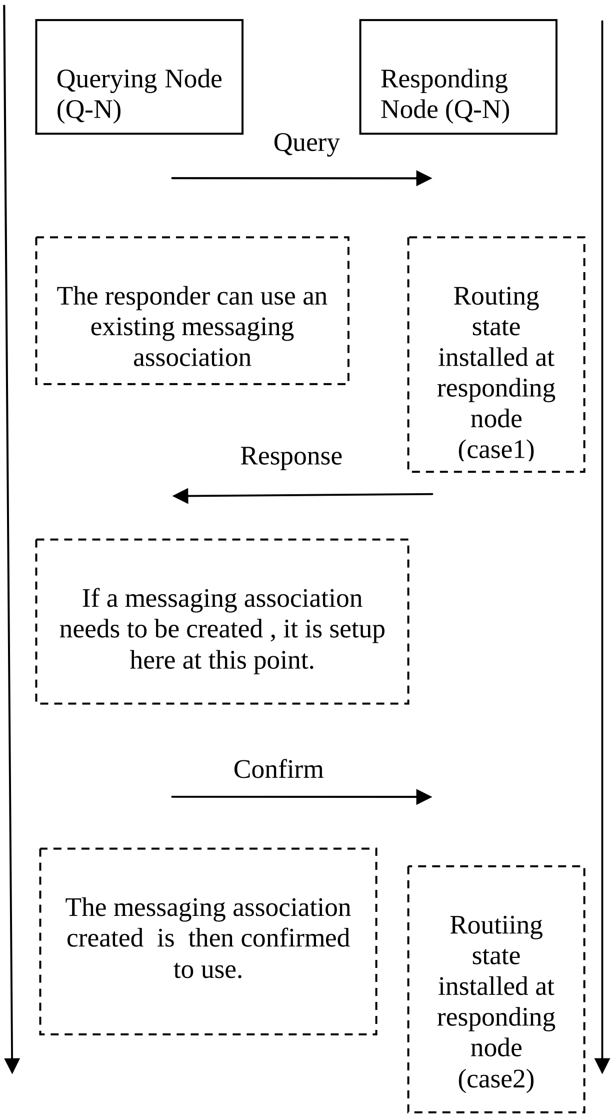

Fig 3: Message Sequence at State Setup [10]

The initial message in any routing state maintenance operation is a Query sent from a Querying Node. There is a router alert option so that the message can be interpreted at the Responding Node. Once a Query is received, the Responding Node must return a Response. The Response also includes Network Layer information of the Responding Node, which could be used by the Query Node to see if the Responding Node is already a known peer and to see if there is a Messaging Association that is already setup with this peer and if that Messaging association can be reused. Through the acknowledgement, the Querying Node and Responding Node can make an agreement. The Querying Node will always take an initiative to set up the Messaging Association once the Querying Node and the Responding Node have made the agreement.

After a Messaging Association has been setup, a confirm message must be sent out by this Message Association. At this point, the Messaging Association for downstream has been setup. The association can also be used in the upstream direction. The Routing State and Messaging

Association of GIST Protocol has been implemented using Coloured Petri Nets Tool and is shown in the next section.

IV. MODELING OF GIST PROTOCOL USING CPN 3.4.0

The GIST is modeled with the aid of the Design/Coloured Petri Nets tool. The basic model of the GIST is shown in Figure 4.

A. Places

There are nine places drawn as ellipses. The places named Quering (Q-N) and Responding (R-N) represents the query node and response node of GIST. These nodes are assumed to have sufficient capacity for flows that might be admitted. The place named NSLP LAYER is the initiator node that sends request to the GIST NODE to start the reservation process and wait for QoS.

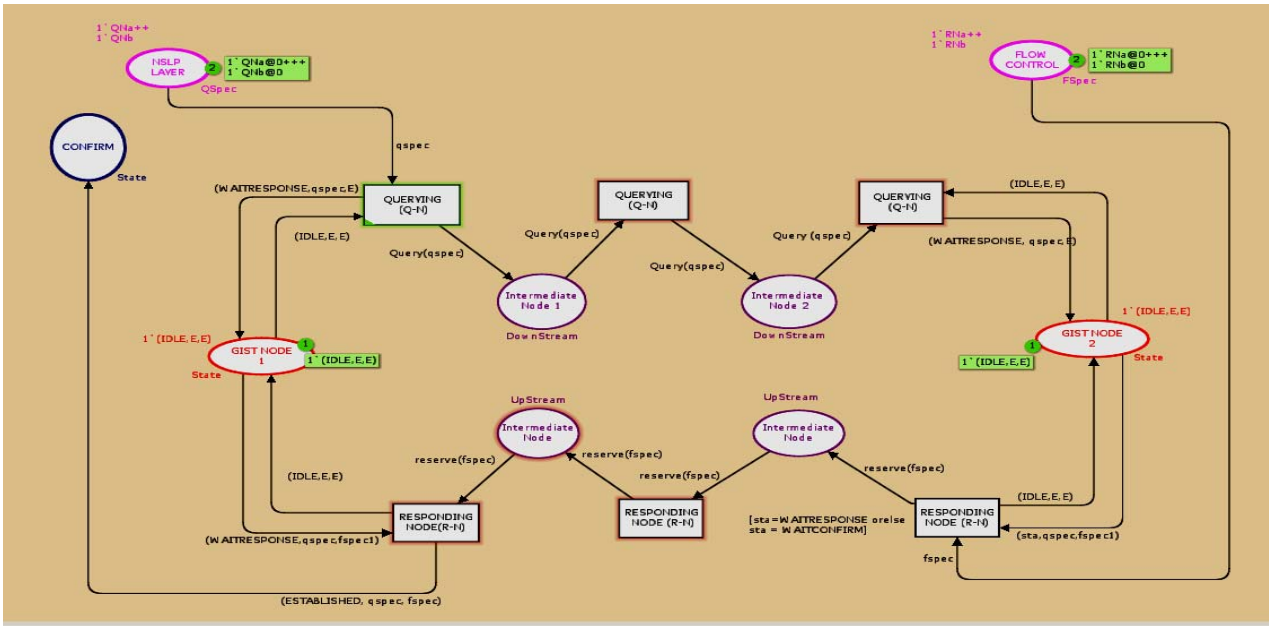

Fig 4. Basic Model of GIST

The place FLOW CONTROL is the responder that gives response for the request by NSLP LAYER. The places named Intermediate nodes are the stateless interior nodes that provide the communication path. A communication path includes all the intermediate devices along the route from the sender to receiver and vice versa.

B. Arcs

Arcs connect transitions and places. A transition may have input places connected by incoming arcs and output places connected by outgoing arcs. Arcs have expressions associated with them. They are located next to arcs and determine which tokens are removed or added to the places.

C. Transitions

Transitions represent the actions of the system. They are drawn as rectangles in Figure 4. There are six transitions in the example. The transition Querying1 (Q-N) models the action taken when the GIST NODE 1 sends a request with the traffic characteristics of the data flow. The reception and processing of a sender request is modelled by the transition Querying2 (Q-N). The transitions Responding Node 1, Responding Node 2 and Responding Node 3 are used to model the reception and processing of a reservation request.

D. Markings

Tokens are associated with each place. A token is a value, which belongs to the type of the place. The *marking* of a place is the multi-set of tokens present on the place. It is a *multi-set* since it may contain several tokens with the same value. For example, the place NSLP LAYER may have the initial marking 1'QNa or 1`QNb, which means that the place has two tokens. It means that the sender can send two requests with the same traffic values.

CPNs include the initial state of the system. It is called the *initial marking*. It is written on the upper left or right of the place. In the initial marking, each of the places Gist Node 1 and Gist Node 2 has a single token with the value (IDLE, E, E), which means that neither the reservation nor the traffic information has been sent yet (as indicated by the value E). Each of the places NSLP LAYER and FLOW CONTROL has an initial marking consisting of two tokens 1`QNa++1`QNb and 1`RNa++1`RNb, respectively. It means that the sender user has two traffic requests with the values QNa and QNb, and the receiver user has two reservation requests with the values RNa and RNb. Initially, the remaining places do not contain any tokens.

E. Variables

An arc expression is evaluated by assigning (*binding*) data values to variables. The result of the evaluation of an arc expression is a multi-set of tokens. The variable declaration is shown in Listing 1.

```
(*************** Variables ******************)
var sta: Status;
var fspec,fspec1: SFSpec;
var qspec: SQSpec;
```

Listing 1: Variable Declaration

F. Enabling and Occurrence of Transitions

A transition can occur if it is *enabled*. For a transition to be enabled in the current marking, it must be possible to *bind* (assign) data values to the variables appearing on the surrounding arc expressions and in the guard and the following conditions must be met. Firstly, each of the input arc expressions evaluates to tokens that are present on the corresponding input places. Secondly, if there is any guard, it must evaluate to true.

The occurrence of a transition removes tokens from the input places and adds tokens to the output places. The removed tokens are the result of evaluating the expressions on the corresponding incoming arcs, while the values of the added tokens are the result of evaluating the arc expressions on the corresponding outgoing arcs.

G. Types

Each place has an associated *type* or *colour set* which determines the type of data the place may contain. The type definitions are shown in listing 2. They are similar to types in programming languages

```
(****************** States ******************)
colset Status = with
IDLE|WAITRESPONSE|WAITCONFIRM |ESTABLISHED;
colset ParValues = with E|QNa|QNb|RNa|RNb timed;
colset STSpec = subset ParValues with [E,QNa,QNb] timed;
colset SFSpec = subset ParValues with [E, RNa, RNb] timed;
colset State = product Status * STSpec * SFSpec;
(***************Messages **************)
colset QSpec = subset ParValues with [QNa,QNb] timed;
colset FSpec = subset ParValues with [RNa, RNb] timed;
colset DownStream = union Query:QSpec + resverror: FSpec;
colset UpStream = union reserve : FSpec;
```

Listing 2: Color Set Definition

GIST NODE 1 and GIST NODE 2 places have the type State. State is the product of the type Status, SQSpec and SFSpec. Status is an enumeration type representing the four states (i.e. IDLE, WAITRESPONSE, WAITCONFIRM and ESTABLISHED). IDLE is the initial state for both the GIST NODE 1 and GIST NODE 2. WAITRESPONSE means that a sender request with the traffic characteristics of the data flow has been sent but no reservation request has yet been received. WAITCONFIRM means that the receiver has sent a reservation request. ESTABLISHED means that the sender has received a reservation request. SQSpec and SFSpec are subsets of the type ParValues. ParValues is an enumeration type, which defines the values (including the empty value E) the parameters can have. SQSpec represents the traffic characteristics of the data flow, which are stored as part of the state information. SFSpec represents the QoS characteristics of the data flow, which are also stored as part of the state information. For example, if the CONFIRM place contains the value

(ESTABLISHED,QNa,RNa), it means that a reservation request, RNa, has been sent for the data flow with QNa traffic characteristics.

V. CONCLUSION

The GIST protocol is designed to support different NSLP signalling applications. This protocol consists of existing transport protocols. The main functionality of GIST is essentially to carry and deliver signalling messages to the appropriate destinations. This function includes the discovery of the right NSIS peer, the use of the right or required transport protocol and the installation and maintenance of session states. It sends the traffic requests and reserve resources. Implementations and testing are the only mechanisms used so far to validate the functionality of GIST Internet Draft. GIST is specified and verified formally using Coloured Petri Nets. The protocol is modeled in such a manner so as to demonstrate that the protocol provides the service expected by the user. The analysis of the GIST model demonstrates that the protocol behaves as expected, given a number of significant assumptions and limitations. In this research work two intermediate nodes are used that are used to communicate with the end nodes. The basic working of the GIST protocol is shown, that is, how the traffic requests are sent and how resources are reserved and acknowledgement is received. By modeling the protocol in Coloured Petri Nets Tool we have formally verified the working of the GIST protocol.

REFERENCES

[1] A. Bader, G. Karagiannis, L. Westberg, C. Kappler, T. Phelan, H. Tschofenig, G. Heijenk, "QoS Signaling across Heterogeneous Wired/Wireless Networks: Resource Management in DiffServ Using the NSIS Protocol Suite", *Proceedings of the Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, August 2005, pp: 22-24, Orlando, Florida, USA..

[2] A. Bader, C. Kappler, T. Phelan, "RMD-QOSM: The NSIS Quality of Service Model for Resource Management in DiffServ", *Proceedings of the 79th Internet Engineering Task Force,* RFC 5977, IETF, 7-12 November, 2010.

[3] A. Bader, L. Westberg, G.Karagiannis, C. Kappler, T. Phelan, "RMD-QOSM: The Resource Management in DiffServ QOS Model", IETF, December, 2005.

[4] A. Verma, "Formal Verification of Ad Hoc N/W Routing Protocols", *International Journal of Advanced Research in Computer Science*,Volume 2, No. 4, July-Aug 2011.

[5] "Coloured Petri Nets Documentation", available at: www.cpntools.org.

[6] D. Durham and R. Yavatkar, "Inside the Internet's Resource Reservation Protocol" Wiley, USA, 1999.

[7] G. Karagiannis, A. Báder, G. Pongrácz, A. Császár, A. Takács, R. Szabó, L. Westberg, "RMD– a lightweight application of NSIS", 11th International Conference on Telecommunications Network Strategy and Planning Symposium, June 2004, pp: 211-216.

[8] H. Tschofenig and D. Kroeselberg, "Security Threats for Next Steps in Signaling (NSIS)", Network Working Group, RFC 4081, June 2005.

[9] H. Bai, M. Atiquzzaman, and W. Ivancic, "Running Integrated Services over Differentiated Service Networks: Quantitative Performance Measurements", Aerospace and Electronic Systems Honeywell Aerospace 3660 Technology Drive, Minneapolis, MN 55418, USA, Vol. 4866, No.1, PP:11-22,.

[10] H. Schulzrinne, R. Hancock, "GIST: General Internet Transport Signaling", IETF Draft, June, 2009.

[11] I. Mahadevan and K. M. Sivalingam, "Quality of Service Architectures for Wireless Networks: IntServ and DiffServ Models", Mobile Computing Workshop atIntl. Symposium on Parallel Architectures, Mobile Algorithms and Networks, Perth, Australia, June 1999, pp: 420-425.

[12] K. Jensen and Lars M. Kristensen, "Coloured Petrinets: Modeling and validation of concurrent system", Springer, July 2009.

[13] K. Jensen, "An Introduction to the Practical Use of Coloured Petri Nets", In: W. Reisig and G. Rozenberg (eds.): Lectures on Petri Nets II: Applications, Lecture Notes in Computer Science vol. 1492, Springer-Verlag 1998, pp: 237-292.

[14] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use", Springer-Verlag, 2nd edition, Vol.1, April 1997.

[15] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use", Springer-Verlag, 2nd edition, Vol.2, April 1997.

[16] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use", Springer-Verlag, Vol.3, April 1997.

[17] K. Jensen, "Special Section on Coloured Petri Nets", International Journal on Software Tools for Technology Transfer, Springer, Vol. 3, No. 4, 1998.

[18] L.M. Kristensen, S. Christensen, and K. Jensen, "The Practitioner's Guide to Coloured Petri Nets", *International Journal on Software Tools for Technology Transfer*, Springer, 1998, Vol. 2, No. 2, pp 98-132.

[19] L.M. Kristensen and K. Jensen, "Specification and Validation of an Edge Router Discovery Protocol for Mobile Ad-hoc Networks", In Integration of Software Specification Techniques for Applications in Engineering. Springer Lecture Notes in Computer Science vol. 3147, Springer-Verlag 2004, pp: 248-269.

[20] L. Westberg, A. Csaszar, G. Karagiannis et al., "Resource Management in Diffserv (RMD): A Functionality and Performance Behavior Overview", Proceedings of 7th International Workshop on Protocols for High Speed Netwroks, Vol 2334, 2002, pp: 17-34.

[21] M. E. Villapol and J. Billington, "Generation of a Service Language for the Resource Reservation Protocol Using Formal Methods", Proceedings of the 11th Annual International Symposium of the International Council on Systems Engineering, Melbourne, Australia, July 1-5, 2001.

[22] M. E. Villapol and J. Billington, "Modeling and Initial Analysis of the Resource Reservation Protocol using coloured Petri Nets", Proceedings of the Workshop on Practical Use of High-Level Petri Nets, Aarhus, Denmark, June 27, 2000, pp 91-110.

[23] M. E. Villapol and J. Billington, " A Coloured Petri Net Approach to Formalising and Analysing the Resource Reservation Protocol", CLEI Electronic Journal, Vol. 6, No. 1, paper 1, 2003.

[24] R. Braden, D. Clark, and S. Shenker "Integrated Services in the Internet Architecture: an Overview", RFC 1633, IETF, June, 1994.

[25] R. Braden, S. Berson, S. Herzog and S.Jamin, "Resource Reservation Protocol (RSVP) – Version 1: Functional Specification", RFC 2205, IETF, September, 1997.

[26] R. Hancock, G. Karagiannis, J. Loughney, "Next Steps in Signaling: Framework", RFC 4080, IETF, June, 2005.

[27] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, IETF, Dec. 1998.

[28] S. Christensen and J. B. Jorgensen, "Teaching Coloured Petri Nets: Examples of Courses and Lessons Learned", Springer-Verlag Berlin Heidelberg, pp. 402–412, 2004.

[29] S. Verma and A. Kaur, "Modeling and Initial Analysis of the RMD-QOSM (Resource Management in DiffServ) Protocol using Coloured Petri Nets", International Journal of Computer Applications, Foundation of Computer Science, New York, USA, volume 50, number 18, 7873-1180, July 2012.

[30] T. Murata, "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, No. 4, April, 1989, pp 541-580.

[31] W. Zhao, D. Olshefski and H. Schulzrinne, "Internet Quality of Service: An Overview", Columbia University Research Report, February 2000.

[32] X. Fu, H. Schulzrinne, A. Bader, D. Hogrefe, C Kappler, G. Karagiannis, H. Tschofenig, S. Vanden Bosch, "NSIS: A New Extensible IP Signaling Protocol Suite", IEEE Communications Magazine, Vol. 43, Issue 10, October 2005,pp: 134-141