Modeling and Initial Analysis of the RMD-QOSM (Resource Management in DiffServ) Protocol using Coloured Petri Nets

Sapna Verma
Punjabi University Regional Centre for IT & Mgmt.
Mohali

Amardeep Kaur
Assistant Professor in (CS)
Punjabi University Regional Centre for IT & Mgmt.
Mohali

ABSTRACT

The NSIS (Next Steps In Signaling) working group within IETF has developed a new signaling framework. One of the purposes of this framework is to support quality of service provisioning. RMD-QOSM is the protocol that delivers quality of service to end users. RMD-QOSM protocol is a simple, effective and scalable resource reservation method that provides edge-to-edge communication in a Differentiated Services (DiffServ) domain. Main functions of RMD-QOSM are congestion control and admission control. The aim of this paper is to use Coloured Petri Nets to model some features of the protocol. Initial analysis refers to that a basic model is constructed using Coloured Petri Nets and its proper working is verified.

Keywords

Quality of Service, NSIS, RMD-QOSM, Formal Verification, Coloured Petri Nets (CPN)

1. INTRODUCTION

QoS plays a very significant role in Internet. Qos is defined in terms of providing service differentiation and performance assurance for Internet applications [30]. Providing different kind of services to different users according to the particular requirements of the user is referred to as service differentiation. Performance assurance addresses bandwidth, loss, delay and delay variation. To provide Quality of Service to internet various QoS frameworks were developed. The first QoS framework that has been standardized by the IETF (Internet Engineering Task Force) is IntServ (Integrated Services) [24], 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) that uses RMD model to provide quality of service [27].

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 [31].

Protocol designs should subject to validation to rule out invalid actions. 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 RMD-QOSM Protocol. Basic model of RMD-QOSM is constructed with the aid of CPN tool in which general features and functionality of RMD-QOSM protocol are included.

The paper has been organised as follows. Section two presents the basic overview of RSVP. Section three includes a detailed explanation of RMD-QOSM, its functions, operations and comparison with other protocols. A description of the CPN model of RMD is analysed in forth section. All assumptions and requirements and simulation results are also presented in this section. Finally, the conclusion is given in section five.

2. OVERVIEW 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]:

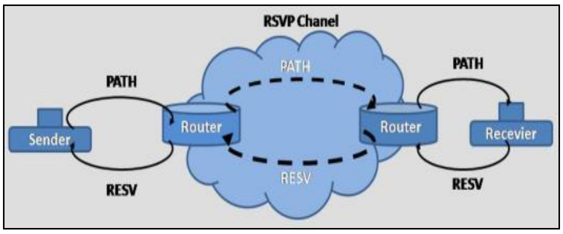

Fig 1: RSVP Signaling

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.

RSVP has certain limitations which are the following [31]:

- 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 signalling
- 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 signalling 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.

3. RMD-QOSM

The Differentiated Services (Diffserv) architecture was introduced as a result of efforts to avoid the scalability and complexity problems of IntServ [11] [9]. RMD was introduced as a method for dynamic reservation of resources within a Diffserv domain. It describes a method to provide admission control for flows entering the domain and a congestion handling algorithm that is able to terminate the flows in case of congestion due to a sudden failure (e.g., link, router) within the domain [7].

In RMD, scalability is achieved by separating a fine-grained reservation mechanism used in the Edge nodes of a Diffserv domain from a much simpler reservation mechanism needed in the Interior nodes. Typically, it is assumed that Edge nodes support per-flow QoS states in order to provide QoS guarantees for each flow. Interior nodes use only one aggregated reservation state per traffic class or no states at all. In this way, it is possible to handle large numbers of flows in the Interior nodes. Furthermore, due to the limited functionality supported by the Interior nodes, this solution allows fast processing of signaling messages [2].

RMD is a technique that introduces dynamic reservation and admission control into the Diffserv architecture network [7]. RMD complements the Diffserv architecture by applying per flow classification, conditioning and admission control functions to the Edges of a Diffserv domain and using simple per traffic class (per DSCP) admission control functions on the interior (core) nodes of the Diffserv domain.

The RMD QoS Model allows entities that are external to the RMD domain to signal reservation requests to Edge nodes in the RMD domain. The RMD Ingress Edge nodes can classify the incoming session flows into traffic classes and signals resource requests for the corresponding traffic class along the data path, along the intermediate interior nodes, to the Egress Edge nodes for each session flow [2]. The Egress nodes receive and reconstitute the original requests and forward them outside the RMD domain along the data path towards the final destination.

Figure 2 shows an RMD network with the respective entities. RMD-QOSM is an Edge-to-Edge (intra-domain) QoS Model that, in combination with the QoS-NSLP and QSPEC specifications, is designed to support the requirements mentioned above: Minimal impact on Interior node performance; Increase of scalability; Ability to deal with severe congestion [2] [3].

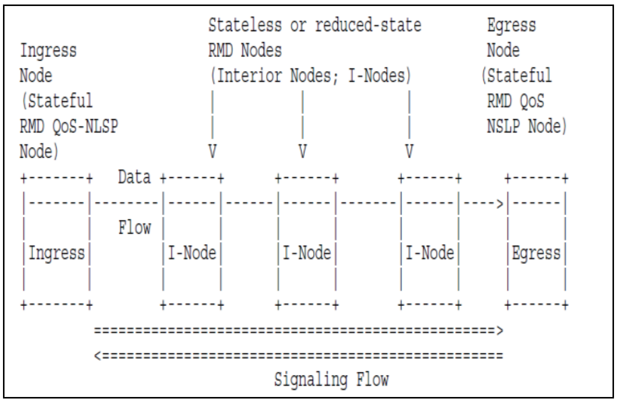

Fig 2: Actors in RMD-QOSM

3.1 RMD Model within QoS NSLP

The RMD QoS Model (RMD-QOSM) makes use of the defined QoS-NSLP messages, but adding a specific RMD-QSpec object [1][2].

The RMD-QOSM QSpec object contains three fields, the <RMD-QOSM QoS Description>, the Per Hop Reservation <PHR RMD-QOSM control information> container (PHR container) and the Per Domain Reservation <PDR RMD-QOSM control information> container (PDR container). The RMDQOSM QoS Description container is only processed by the edge nodes. The two control information containers are processed by edge and interior nodes.

3.1.1 RMD-QOSM QoS Description

The RMD-QOSM QoS Description field contains two QoS objects as shown in Table 1.

Table 1. QoS Description Field

Parameters	**Description**
Bandwidth	The bandwidth used by the flow
PHB-CLASS	Indicates the recommended DSCP value for the flow

3.1.2 PHR RMD-QOSM control information container (PHR container)

The Per-Hop Rservation (PHR) container supports the resource reservation procedure and is processed by all nodes on the flow path. Table 2 shows the parameters which are used in the PHR container.

Table 2. PHR Container

Parameters	**Description**
Flags	Flags for service use to support parameter handling
Container ID	
1	PHR_RESOURCE_REQUEST: This container type is used to initiate or update the PHR reservation in the intra-domain nodes that are on the path between the Ingress and Egress nodes.
2	PHR_REFRESH_UPDATE: It is used to refresh the PHB reservation soft state on all nodes located on the communication path between the Ingress and Egress nodes.
3	PHR_RELEASE_REQUEST: With this container type the reserved resources will be released for a particular flow
Reserved	Bits reserved for future use
Length	The length of the parameter in bytes
Overload %	Gives the percentage of the overload in case of severe congestion.
M	If a QNE node cannot reserve the specified resources it sets the M bit in order to notify the Egress QNE
Admitted Hops	Counts the number of nodes in which the reservation was successful. Set to 0 in the ingress node
B	If set bi-directional reservation is requested

3.1.3 PHR RMD-QOSM control information container (PDR container)

The Per-Domain Reservation (PDR) container provides additional support to the PHB container for the connection establishment. The PDR container is in the base of the end-to-end communication. The description of its fields is shown in Table 3.

Table 3. PDR Container

Parameters	**Description**
Flags	Flags for service use to support parameter handling
Container ID	
4	PDR_RESERVATION_REQUEST: used for resource reservation
5	PDR_REFRESH_REQUEST: used for resource refresh
6	PDR_RELEASE_REQUEST: used for resource release
7	PDR_RESERVATION_REPORT: result of reservation procedure
8	PDR_REFRESH_REPORT: result of refresh procedure
9	PDR_RELEASE_REPORT: result of release procedure
10	PDR_CONGESTION_REPORT: indicates severe congestion
Reserved	Bits reserved for future use
Length	The length of the parameter in bytes
S	Indicates severe congestion occurrence. 0 for no congestion, 1 for congestion
M	Indicates node possibility to reserve resources. 1 indicates insufficient resources
Max admitted hops	Counts the number of nodes where the reservation was successful. Set to 0 in the ingress node
B	If set bi-directional reservation is in place
Overload %	Indicates the level of overload detected. Every node checks its own level and if necessary updates it.

3.1.4 Basic Operation of RMD-QOSM

Figure 3 shows the sender initiated reservation technique based upon which the basic model of RMD-QOSM is drawn. The initiator of the request (QNI) is the sender of the data flow. QNI generates a RESERVE message with the initiator QSPEC. At each intermediate QoS NSLP aware node first authentication and policy control are performed. Second, the control information is processed and the QSPEC is sent to the RMF. The RMF performs the resource check-up and if the reservation is admitted, a reservation and an operational state are installed and the RESERVE message is sent to the next peer [2] [3]. If any of the nodes do not have sufficient available resources, then no states are installed and a RESPONSE message is returned right away to the initiator. When the receiver (QNR) gets RESERVE message if the RII object is included a RESPONSE is initiated. QNR also installs states if the requested resources are free.

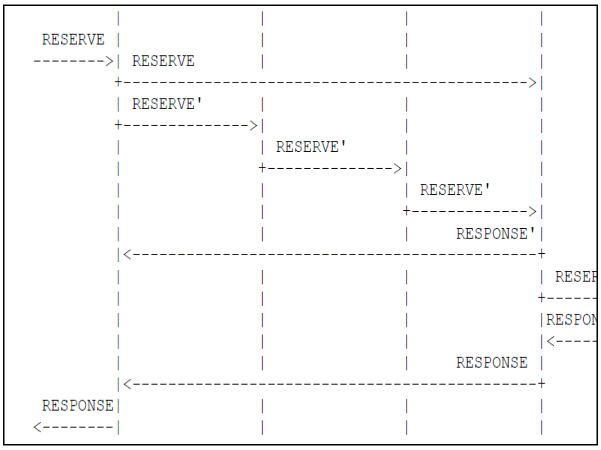

Fig 3: Sender Initiated Reservation with Reduced State Interior Nodes

The basic model of the RMD-QOSM is shown in Figure 5. It is divided into five parts: QNI (Sender User), QNE (Ingress), QNE (Interior1), QNE (Interior2), QNR (Receiver User).The basic components of the CPN model are described as follows.

Places

There are eight places drawn as ellipses. The places named Ingress and Egress represents the QNE statefull Ingress and Egress nodes of RMD. These nodes are assumed to have sufficient capacity for flows that might be admitted. The place QNI is the initiator node that gives request to the Ingress to start the reservation process and wait for QoS.

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 Figure 6. They are similar to types in programming languages The place QNR is the responder that gives response for the request by QNI. The places named Interior1, Interior2, ReceivertoInt2 and Int2toInt1 are the stateless interior nodes that provide the communication path

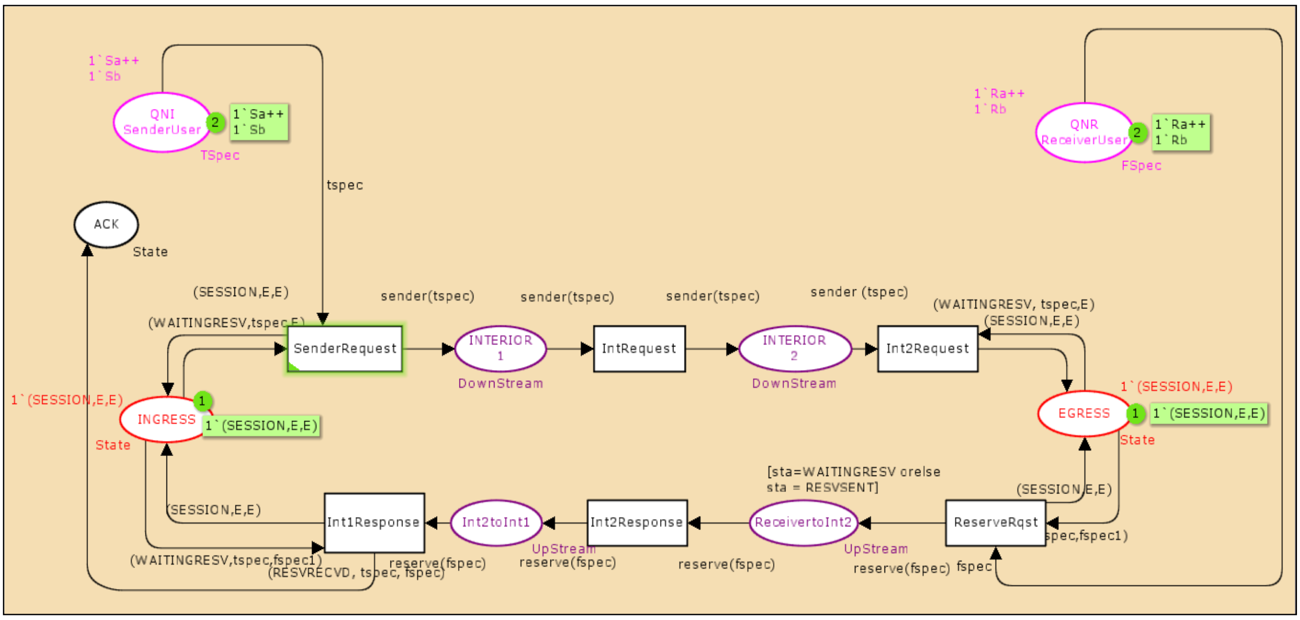

Fig 5: Basic Model of RMD-QOSM

4. CPN MODEL OF RMD-QOSM PROTOCOL

The RMD-QOSM is modeled with the aid of the Design/CPN tool. The model of the RMD-QOSM service specification consists of mainly two pages illustrated in Figure 4. The structural view has been designed based on the major RMD-QOSM service events and is intended to increase the readability of the model.

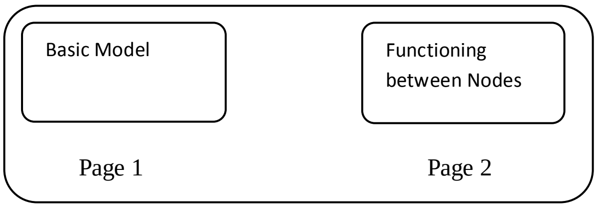

Fig 4: Abstract view of the RMD-QOSM Model implemented in CPN

```
(****************** States ******************)

    colset Status = with SESSION|WAITINGRESV
                         |RESVSENT|RESVRECVD;
    colset ParValues = with E|Sa|Sb|Sa|Sb;
    colset STSpec = subset ParValues with [E,Sa,Sb];
    colset SFSpec = subset ParValues with [E,Ra,Rb];
    colset State = product Status * STSpec * SFSpec;

  (**************Messages **************)

    colset TSpec = subset ParValues with [Sa,Sb];
    colset FSpec = subset ParValues with [Ra,Rb];
    colset DownStream = union sender:TSpec +
                              resverror: FSpec;
    colset UpStream = union reserve: FSpec;
```

Fig 6: Colour set definitions

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 QNI may have the initial marking 2`Sa, which means that the place has two tokens, each with the value Sa. 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 Ingress and Egress has a single token with the value (SESSION, 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 QNI and QNR has an initial marking consisting of two tokens 1`Sa++1`Sb and 1`Sa++1`Sb, respectively. It means that the sender user has two traffic requests with the values Sa and Sb, and the receiver user has two reservation requests with the values Ra and Rb. Initially, the remaining places do not contain any tokens.

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 SenderRequest models the action taken when the ingress sends a request with the traffic characteristics of the data flow. The reception and processing of a sender request is modelled by the transition IntRequest. The transition ReserveRqst models the action taken when the egress generates a reservation request. The transitions Int2Response and Int1Response are used to model the reception and processing of a reservation request.

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.

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 Figure 7.

Fig 7: Variable Declaration

```
(*************** Variables ******************)
                var sta: Status;
                var fspec,fspec1: SFSpec;
                var tspec: STSpec;
```

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.

The page drawn in Figure 8 shows the functioning between the nodes i.e functions between the ingress and interior node and functions between the interior and egress node.

The transition QNI REQUEST is enabled when a token, which resides in the QNI (SenderUser) is sent to the Ingress node. The status of the INGRESS is different from CLOSED. An occurrence of the transition updates the token, which resides in the QNI, so that the state of the INGRESS indicates the new requested TSpec. If the status of the INGRESS is equal to SESSION, it is also updated so that the token residing on the QNI place indicates that the sender is ready to receive a reservation request (WAITINGRESV). Otherwise, it is not required to update the status of the sender because it already indicates that a TSpec has been requested.

The IntRequest is enabled when a token residing in the EGRESS place indicates that the status of the receiver is different from CLOSED and NOSENDER, and the incoming TSpec (*tspec*) is different from the one received previously (*tspec1*). An occurrence of the transition updates the token residing on the EGRESS place so that the value of the TSpec is the one received from the provider. If the status of the receiver is different from SESSION, it is also updated so that it indicates that the receiver is ready to send a reservation request (WAITINGRESV). In addition, a token representing a sender information flow (i.e. *sndrefresh*) is added to the INTERIOR place to allow multiple SenderError to occur.The DISCARD FROM INTERIOR is enabled when a token representing a sender flow (i.e. *sender* or *sndrefresh*) resides in the INTERIOR place. Also, a token, which resides in the INGRESS place, indicates that the status of the sender is either CLOSED or SESSION or the TSpec (*tspec*) carried in the sender flow is different from the one sent previously (*tspec1*). An occurrence of the transition removes the token representing the sender flow from the INTERIOR place.

The SenderReleaseReq transition models the action taken when the sender user leaves the session. This transition is enabled when the status of sender indicates that the INGRESS node has not left the session. An occurrence of the transition updates the state of the INGRESS so that the token residing on the INGRESS place indicates that the status of the INGRESS is equal to CLOSED. Also, any TSpec or FSpec information is deleted by updating the token with the E value. If a sender request has been sent, a token, which represents a *senderrel* flow, is added to the INTERIOR place (i.e. a *senderrel* flow is available for the provider to process).

The SenderRelInd transition is enabled when a token in the EGRESS place indicates that the QNR is still in the session (i.e. the receiver status is different from CLOSED). Also, a token representing *senderrel* resides in the INTERIOR place. An occurrence of the transition updates the status of the EGRESS to NOSENDER, which means that there is no sender on the session. It also updates the TSpec and FSpec values to E, which indicates that there is no data traffic nor reservation information at the receiver.

The DISCARD FROM EGRESS is enabled depending on the following conditions. If a token, which resides in the INTERIOR place, represents a INGRESS flow, the status of the EGRESS is equal to CLOSED or NOSENDER or the incoming TSpec (*tspec*) is equal to the one received previously (*tspec1*). Otherwise, if the token represents a *resvconf* or a *senderrel* flow, the status of the EGRESS is equal to CLOSED or NOSENDER. Finally, if the token represents a *sndrefresh* flow the transition is not enabled.

An occurrence of the transition removes the token representing the downstream flow from the INTERIOR place.
The ADMISSION CONTROL is enabled if the request is received by the INTERIOR node that is beyond its capacity. Like in this we have used the function erraccept(tspec) which says that only Sa will be accepted. If the request Sb or Sc will be received by the INTERIOR node then this transition will be enabled and the error will be stored in Err Db.

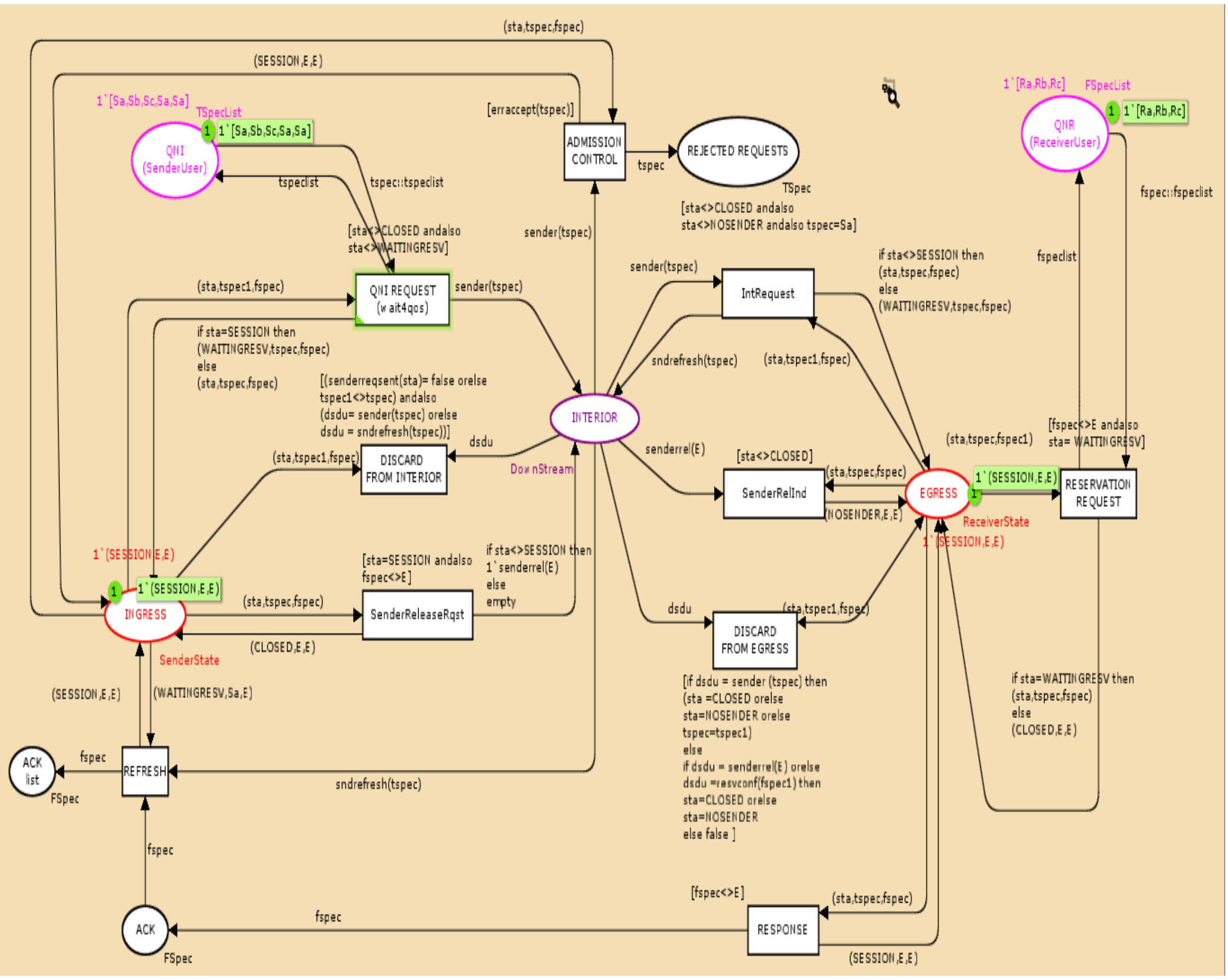

Fig 8: Functioning between the Nodes

5. CONCLUSION

RMD-QOSM is a protocol to provide the QoS for some applications like multimedia and real time. It sends the traffic requests and reserve resources. Implementations and testing are the only mechanisms used so far to validate the functionality of RMD-QOSM. RMD-QOSM is specified and verified formally using Coloured Petri Nets. The protocol is modelled in such a manner so as to demonstrate that the protocol provides the service expected by the user. The analysis of the RMD-QOSM model demonstrates that the protocol behaves as expected, given a number of significant assumptions and limitations. In this research work two interior nodes are used that are used to communicate with the end nodes. In the first page normal working of the RMD-QOSM protocol is shown, that is, how the traffic requests are sent and how resources are reserved and acknowledgement is received. In the second page functioning between the nodes are shown. Some limitations are added like request other than Sa will be rejected and the error message will be received due to insufficient resources at that particular node. It is also shown that how the data is transferred from ingress to egress. By modeling the protocol in Coloured Petri Nets Tool we have formally verified the working of the RMD-QOSM protocol.

6. 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*, 2005.

[2] A. Bader, C. Kappler, T. Phelan, "RMD-QOSM: The NSIS Quality of Service Model for Resource Management in DiffServ", RFC 5977, IETF, October, 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, 2004.

[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.

[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.

[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, 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, 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, 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, 2002.

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

[30] W. Zhao, D. Olshefski and H. Schulzrinne, “Internet Quality of Service: An Overview”, Columbia University Research Report, 2000.

[31] 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, 2005, pp 134-141.

.