

End-to-End Resource Reservation in IP Mobility Scenarios: A Survey

Xing Tao and Lin Hai*

*International School of Software, WuHan University
LuoJia Shan, Wu Chang
430079 China
{xingtao, lin.hai}@whu.edu.cn*

Abstract

In order to provide guaranteed quality of service for real-time applications in the next IP-based mobile networks, end to end resource reservation should be executed. Two IETF protocols, RSVP and QoS NSLP, have been devised for resource reservation. But both of them are developed without considering mobility management protocols. Tremendous previous works have been carried out to extend these protocols to improve the performance when they are used in IP-based mobility scenarios for resource reservation. This survey provides a description and analysis of these previous works. We list some critical issues regarding resource reservation in IP-based mobility scenarios, and analyze the solutions proposed to address these issues. Also, we point out some opened issues that need to be resolved before resource reservation is widely used in mobility scenarios.

Keywords: *IP Mobility, Network Mobility, QoS, Resource Reservation, RSVP, QoS NSLP, Advance Reservation, Reservation Aggregation*

1. Introduction

In the last few years, we notice a tremendous growth of portable devices (*e.g.* Mobile phones, Tablets, *etc.*), that support unique or multiple wireless interfaces. One reason for this growth is the low cost and the facility to deploy wireless networks. In wireless/mobile networks, users can change anytime their service points while keeping the communication with others. One of the most important requirements to optimize portability and to provide a continuous connectivity is the efficient support of mobility. For this reason, IETF (Internet Engineering Task Force), has developed Mobile IP (MIP), version 4 [1] and version 6 [2], to support the mobility of devices. A mobile device is a Mobile Node (MN) which can change its point of attachment to the Internet and then its IP address. The goal of MIP is to provide a host the ability to stay connected to the Internet “anywhere” and “anytime”.

To support real-time multimedia applications (*e.g.* Voice-over-IP, Video-over-IP, and IP TV) in mobile-IP supported networks, Quality of Service (QoS) should be provided for these networks. Two major models are proposed by IETF to guarantee QoS in the IP-based networks: Integrated Service (IntServ) [3] and Differentiated Service (DiffServ) [4] models. In IntServ model, network resources are explicitly identified and reserved. Network routers classify incoming packets and use reservations to provide QoS. In DiffServ model, resources are not explicitly reserved. Instead, traffic is differentiated into a set of classes, and network nodes provide priority-based treatment according to these classes. Resource Reservation Protocol (RSVP) [5] is a resource reservation setup protocol designed for the IntServ model. However, RSVP was designed when mobility was in its infancy and therefore does not take account of mobility characteristic. Lots of works [6], [10]

have been carried out to extend RSVP to make it applicable in mobility scenarios. However, people found that, due to RSVP's limitation, these extensions need to change the basic logic of RSVP, and they become more and more complex. This is why another signalling protocol, Next Steps in Signaling (NSIS), which has been developed by the IETF NSIS working group, considers the mobility characteristic for its QoS signaling. This protocol intends to design a signaling protocol which is more suitable for mobility scenarios, and will probably replace RSVP protocol. But it must be noted that this group addresses not only QoS signaling protocol, but also other signaling applications, such as firewall and NAT control [12].

The requirements and challenges of resource reservation for IP Mobility can be found in [1], [13]. In this paper, we give a survey that describes and analyses the previous works on reservation for mobility scenarios from the following aspects:

- *Interaction with mobility protocols,*
- *Obsolete reservation release,*
- *Reservation in advance,*
- *Reservation aggregation.*

Also, we will discuss some related approaches (*e.g.* path-decoupled or path-coupled) regarding resource reservation in mobility scenarios.

Resource reservation protocols are developed independently of Mobile IP protocols. To adapt efficiently resource reservation protocols in mobility scenarios, the resource reservation protocol must interact with Mobile IP protocols. For example, Mobile IP protocol should inform the reservation protocol about the MN's mobility status (*e.g.* a handover being impending, a handover being ended), so the reservation protocol can decide when to reserve resources for the MN's next location. Another issue is, with Mobile IP, reservation protocol messages are invisible to intermediate routers of the IP tunnel between Home Agent (HA) and Foreign Agent (or MN) due to the IP tunnel encapsulation. Hence it is necessary for mobile IP to act differently regarding the reservation message than data packets [14][15]. To improve the performance of mobile IP, some extensions, *e.g.* Hierarchical MIPv6 (HMIPv6) [17] and Fast MIPv6 (FMIPv6) [17], have been proposed to manage the mobility more efficiently. It is desired to enable reservation protocol to work with these extensions.

Another issue regarding resource reservation in mobility scenarios is to release obsolete reservations. Reservations for the MN's current location become obsolete after its departure. These obsolete reservations should be released as quickly as possible. Otherwise, they would block other resource requests. It is relatively simple to release obsolete reservations in the cases where a reservation proxy performs reservations on behalf of the MN, because the reservation proxy can be responsible for the obsolete reservation release [7], [19], [20]. In the case where the reservations are performed without the reservation proxy, it requires a new entity, often the Crossover Node (CRN) [1], [21] that is intersecting node between the old path and the new path, to take care of the obsolete reservation release. If the CRN is also incapable, a new method which resorts on reservation's soft state mechanism can be used for this task [22].

To avoid service disruption after the MN's handover, reservation for the MN's next location should be performed before the arrival of the MN. This is called reservation in advance (or advance reservation), and which is a critical task in reservation for mobility scenarios. To do this, a simple method is to perform reservation on all MN's neighbouring locations or all the locations that a MN is expected to visit [6][7]. However, this mechanism involves a tremendous waste of resources, because the advance reservation is not used before the MN moves to these locations. Hence, it is more acceptable to reserve resources only on a neighbouring location which will be probably visited [8]-[9]. Obviously, the latter

method reduces resource waste, but it raises the risk that the advance reservation is not yet completed when the MN connects to the new location. To optimize the resource utilization while keeping continuous QoS guarantee, Lin *et al* [23] tries to calculate an optimal trigger time to start the advance reservation for the next reservation based on the parameters such as the MN's velocity and the advance reservation latency.

Finally, the issue arises when a set of hosts moving collectively as a unit, which refers to the situation of Network Mobility (NEMO) [18]. NEMO is an extension of MIPv6, and which is used to manage the mobility of all hosts in a mobile network. In NEMO, all flows between Mobile Network Nodes (MNNs) and Correspondent Nodes (CN) go through the path from mobile router to home agent or from home agent to mobile router. If these flows perform resource reservation independently, it leads to a high number of states stored at routers existing between home agent and mobile router and a high number of signaling messages processed at these routers. Therefore, aggregating these individual reservations into one reservation (or multiple reservations) should be handled in NEMO mobility [25], [27]. The reservation aggregation protocol proposed by IETF can be used in this case.

For simplicity reason, RSVP only supports the receiver-initiated and path-coupled approaches, not the sender-initiated and path-decoupled approaches. Yet, our analysis on mobility scenarios illustrates that it is also interesting to consider the sender-initiated and path-decoupled approaches. Fortunately, QoS NSLP is more flexible and can support all these approaches. We will discuss these approaches in this survey.

This survey will elaborate these challenges and provide the reader with an overview of the most important previous contributions to address these challenges. But we will also identify some opened issues that have not yet been considered currently, and discuss some possible solutions for these issues.

The remainder of this survey is organized as follows: Section II contains a brief overview of two resource reservation protocols: RSVP and QoS NSLP. Section III analyses the interaction challenge between the reservation protocols and mobility protocols. Section IV discusses the challenge of reservation in advance and its related works. In Section V, the challenge of release of obsolete reservation is analysed. Then the challenge of reservation aggregation is discussed in Section VI. We discuss the sender-initiated and receiver-initiated approaches, as well as path-coupled and path-decoupled approaches in Section VII. Finally, Section VIII contains concluding remarks.

2. RSVP and NSIS

RSVP [5] is a receiver-initiated resource reservation protocol. It constructs the reservations by exchanging the PATH and RESV messages between a sender and a receiver. The sender (such as video server) periodically delivers the PATH message, and each PATH message containing QoS parameters is forwarded to the receiver(s). When the PATH message is delivered from a sender to a receiver, all RSVP capable routers on the path intercept the message and set the PATH states. After the receiver receives the PATH message, the receiver should reply by sending the RESV message with desired QoS parameters. The RESV message is delivered back to the sender along the reverse path of PATH message. If the required resources on the path are available, the soft-state reservations are established. Otherwise, the *RESVErr* messages will be replied to the receiver.

RSVP was designed when node mobility was in its infancy and therefore does not consider mobility characteristic [6]-[7]. In RSVP, flow identifier, *e.g.* a 5-tuple, is used to identify a flow. However, MN's address is changed after each handover, so the flow identifier is altered as well. Consequently, the same application data flow is

considered as a different flow at the network layer. The MN must establish new reservation in all intermediate routers along the flow path whenever the MN changes its attachment point. As a result, more signaling messages are exchanged between the source node and the destination node and more processing is required. So, the application suffers from more delay. From the point view of security, RSVP does not offer proper security protection, but uses existing security protocols. However, security support, for example authentication and key management, is required in the next generation mobility scenarios. Another problem is the RSVP's choice of transport mechanism that imposes constraints on network architectures and signaling applications. For example, IP or UDP is used for carrying RSVP messages across the network where PATH messages are addressed end-to-end with an IP router-alert option. RSVP uses non-adaptive retransmissions for reliability. Since fragmentation of messages is not allowed, the message length is limited to the Maximum Transport Unit (MTU) size.

Due to these shortcomings of RSVP, the IETF NSIS working group designed another protocol suite in order to accommodate new signaling needs. As a result, an extensible IP signaling architecture referred to as NSIS was developed. This protocol consists of two separated layers [28]: the lower layer called NSIS Transport Layer Protocol (NTLP) [30] provides a generic transport service for the higher layer called NSIS Signaling Layer Protocol (NSLP). QoS NSLP [29] is one of NSLP and designed for QoS signaling.

QoS NSLP can be used for any QoS model, *e.g.* IntServ or DiffServ. This is realized by separating QSPEC (an object which contains all information specific to a QoS model) [32] from QoS NSLP. Reservation specific parameters, *e.g.* available bandwidth or token bucket sizes, encapsulated in the QSPEC object, are carried from one QoS NSLP node to another in order to configure corresponding nodes to satisfy the requested resource. QoS NSLP treats this object as an opaque object. Hence, various QoS models can be designed, and these do not affect the specification of the QoS NSLP protocol. At each QoS NSLP node, the Resource Management Function (RMF) handles the content of the QSPEC. The local QoS model describes how RMF should interpret QSPEC, and how to grant and configure the resource. In order to use effectively resource reservation when the MN's address is changed (because of mobility), NSIS introduces a session identifier. A session identifier is a cryptographically random number used to uniquely identify a signaling session and a signaling state, independent of the flow identifier. Therefore, reservations along the unchanged path that are created when MN stays at the previous location can be reused after handover.

To reserve resources along path from sender to receiver, QoS NSLP defines four messages (Figure 1.):

- **RESERVE:** RESERVE message is the only message that manipulates QoS NSLP reservation state. It is used to create, refresh, modify and remove such state.
- **QUERY:** QUERY message is used to request information about the data path without making reservation. This functionality can be used to 'probe' the network for path characteristics or for support of certain QoS models, or for initiating a receiver-initiated reservation.
- **RESPONSE:** the RESPONSE message is used to provide information about the result of a previous QoS NSLP message, *e.g.* confirmation of a reservation or information resulting from QUERY. Note that if RESERVE message or QUERY message requests a RESPONSE message, they must include a Request Identification Information (RII) object. Otherwise, no RESPONSE message can be sent back unless an error occurred.

- NOTIFY: this message is used to convey information to a QoS NSLP Entity (QNE). For example, if one node detects a route that has changed, it can send this information to other nodes through NOTIFY message.

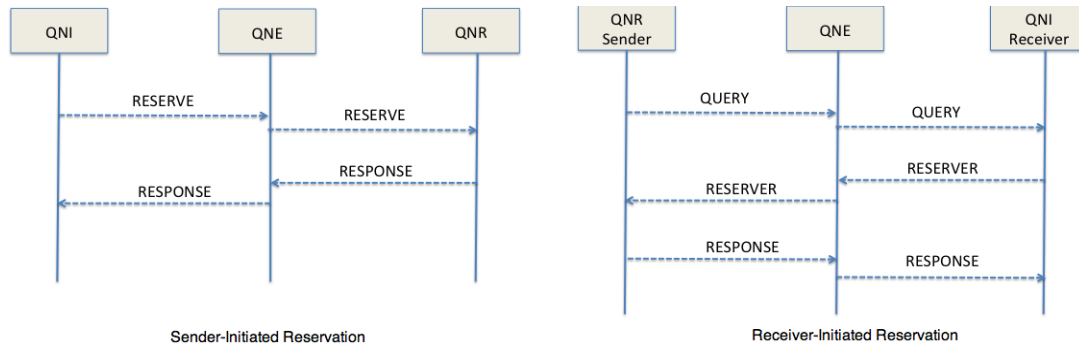


Figure 1. NSIS QoS NSLP (a) Sender-initiated (b) Receiver-initiated

QoS NSLP messages are sent from NSLP peer to NSLP peer rather than from end to end. Upon receiving these messages from its neighbour QNE, this QNE processes these messages, and then generates a new message based on the receiving message and sends to the next QNE.

Reservation with QoS NSLP can be initiated by a sender or receiver, which we call the sender-initiated reservation or the receiver-initiated reservation respectively (Figure 1.). In the sender-initiated case, the sender initiates a RESERVE message, each node along path from the sender and the receiver performs configuration based on the received message. The receiver replies with a RESPONSE message if RII (Request Identification Information) object is included in the RESERVE message. In the receiver-initiated case, the sender sends a QUERY message to the receiver. QUERY message gathers information along path and causes a reverse-path state to be installed. This reverse-path state enables the RESERVE message, which will be sent by the receiver toward the sender, to be forwarded along the reverse of the path that the QUERY message took. Similar to the sender-initiated approach, any node may include a RII in its RESERVE message to require a RESPONSE message.

From the above analysis, we can see that QoS NSLP possesses an advantage on RSVP for being used in mobility scenarios, because the former takes account of mobility characteristic. For example, a session identifier being independent of node's IP address makes that the previous reserved resource can be reused after the mobile node's handover; supporting both sender-initiated and receiver-initiated approaches makes the mobile node performs reservation more efficiently (see Section VII). But still, like RSVP, the original design of QoS NSLP is for wired network case. To apply it to mobility scenario, extension is required [1].

3. Interaction with Mobility Protocols

As we described above, a major issue raised when the RSVP is adopted for mobility scenario is that all reservations performed when the MN connects to the Previous Access Router (PAR) become obsolete after the departure of the MN. It cannot reuse reservations along the common path (from the previous location and the new location) and has to perform again resource reservation from end to end. Hence, in order to enable the RSVP to work in mobility scenario, the first modification is to change the RSVP semantics to include, like QoS NSLP, a unique identifier to identify a session and this identifier is independent of the MN's Care-of-address [20]. Another issue (for both RSVP and QoS NSLP) is that, as described above, reservation messages are invisible to intermediate routers of the IP tunnel. To address this issue, Terzis *et al.* [14] propose RSVP Tunnel.

That is a nested RSVP being established between the tunnel end-points (for example, between home agent and foreign agent). Then, upon receiving the original PATH and RESV messages, the home agent and the mobile router exchange an extra pair of PATH or RESV messages, called tunnel PATH and tunnel RESV messages. Tunnel PATH and RESV messages are carried in normal IP, instead of IP-in-IP datagram, and thus the routers on the tunnel can recognize these messages and reserve resources for the MN accordingly. The same principal can be found in NSIS Tunnel [15] to address this invisibility issue existed in IP tunnel.

Then a simple interaction between the RSVP (or QoS NSLP) and Mobile IP (MIP) protocol is that MIP, after receiving Binding Acknowledgement (BA) message, informs RSVP that an IP handover occurs, and it needs to perform resource reservation from the new location. Obviously, this procedure takes a long time to reserve resources for its new location, and which inflicts a notable degradation in application-level performance. A. Belhoul *et al.* [33] propose a new protocol MARSVP to combine these two protocols by including mobility-specific information in RSVP messages, so as to reduce the signaling overheads and resource reservation latency. MARSVP introduces two new RSVP objects. One is the BU object that is used to store the BU message. The other is the BAcK object that is used to store the BAcK messages. Since no changes are required to be made to the legacy RSVP-enabled routers, RSVP signaling continues to operate in its conventional manner, unaffected by these two new objects. Only the mobility related nodes (*e.g.* CN, HA) are configured to be aware of these objects. Thanks to these objects, no more Binding Update procedure is required, so the signaling message and the reservation latency are minimized.

Yi Sun *et al.* [16] propose a protocol, called Fast RSVP, to reserve resources for mobile IPv6. The protocol adopts a cross-layer design approach where two modules (RSVP module and Mobile IPv6 module) at different layers cooperate with each other. Fast RSVP divides a handover process with QoS guarantees into two stages: 1. Setting up of the resource reservation neighbour tunnel. 2. Resource reservation on the optimized route. It can help a mobile node to realize fast handover with QoS guarantees as well as to avoid resource wasting by triangular routes, advanced reservations and duplicate reservations.

In [19]-[20], RSVP' interaction with the Hierarchical MIP is proposed. Just like Hierarchical MIP, in which Mobility Anchor Point (MAP) manages local mobility and the home agent is not aware of the mobility, the change of reservations due to local mobility is also kept within the local domain. A new entity QoS Agent [19] or RSVP Mobility Proxy (RSVP-MP) [20] (either of which is referred as reservation proxy in following) is introduced to be responsible for local resource reservation and acts as a translator between the local reservation message and the external reservation message. That is, when the RSVP-MP receives the local reservation message, it swaps Local CoA (LCoA) in the message header with Domain CoA (DCoA) then forwards to exterior, and vice versa. Therefore, a local mobility that does not introduce new DCoA will not change reservations of the outside of local domain. The reservation message sent to outside is only used do refresh the reservation states. Since the reservation proxy cannot detect the MN's mobility itself, it must be informed by another entity of the MN's mobility. This can be done either by the MN itself [19] or by MAP [20]. Obviously, the former enables the reservation proxy to detect the MN's mobility quicker (normally, the MN is the entity that detects the mobility in first). However, it requires the MN to implement additional functionality for discovering the reservation proxy address. Elleignand [34] integrates also the RSVP into HMIPv6 by using some Fast MIPv6 characteristics. The L2 handover trigger conformed as a Fast handover is used by the MN to send a message to the MAP. Since this message includes the new AP identifier, the MAP is capable to identify to which Access Router (AR) the MN will move. Then it behaves as reservation proxy to reserve resources from itself to the new AR. In [32], QoS is provided for F-HMIPv6 that

is the combination of FMIPv6 and HMIPv6. Instead of using exiting QoS signalling protocol, authors propose a simple protocol that communicates both traffic specification and service specification to provide Diffserv QoS.

The principal objective for interaction with the mobility protocol is to enable the reservation protocol to be aware of the mobility. The reservation protocol can be informed before or after the mobile node's handover. Yet, to keep continuous QoS guarantee, it is desired that the reservation protocol should be informed before the handover, so it can perform advance reservation (see next Section) for the mobile node's next location.

4. Advance Reservation

The most important task for reservation in mobility scenarios is to let MNs seamlessly obtain reserved resources at the location where they are expected to move, so as to avoid service disruption. This is why most of previous works focus on it. For example, in Figure 2, when the MN moves from the Previous Access Router (PAR) to the New Access Router (NAR), the reservations along the new path (between the NAR and the CRN) should be already performed before the arrival of the MN. In the literature, three strategies to provide advance reservation are proposed.

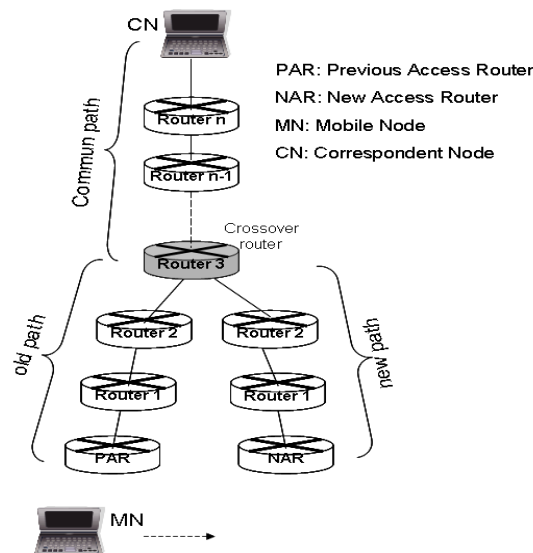


Figure 2. Resource Reservation in Mobility Scenarios

The first is to perform reservation on all MN's neighboring locations or all the locations that a MN is expected to visit once a MN initiates a session. Two important propositions of this type of strategy can be found in [6]-[7]. Talukdar *et al.* [6] propose MRSVP (Mobile RSVP), an extension of RSVP enabling a MN to make advance resource reservations at a set of locations it may possibly visit during the lifetime of the connection. These locations are included in Mobility Specification (MSPEC). Then active reservation is set up from the MN's current location, and passive reservation is set up from each of other locations in its MSPEC. When the MN moves to a new location within its MSPEC, the active reservation from/to its previous location will be turned into a passive reservation and the passive reservation from/to its new location has to be converted to an active reservation. To optimize resource utilization, passive reservation can be used by some flows requiring weaker service guarantees.

A similar proposition can be found in [7], where authors propose a RSVP extension based on multicast. Under this approach, RSVP messages and actual data packets are delivered to a MN using IP multicast routing. The multicast tree, rooted at each source node, is modified dynamically every time a MN handovers to a neighboring location.

Hence, the mobility of the MN can be considered as a transition in multicast group membership. Like MRSVP, reservations are classified as conventional reservation which corresponds to active reservation in MRSVP, predictive reservation which corresponds to passive reservation in MRSVP, but is only created from/to the neighboring locations surrounding the current cell of the MN. Then those predictive reservations that are temporarily used by other data flows are called *temporary reservations*. Temporary reservations will be preempted when the MN that made the predictive reservation moves into current location (*i.e.* when predictive reservation made by this MN becomes active).

The above approaches suffer from several drawbacks. First, advance reservation (passive or predictive reservation) on all the locations that a MN is expected to visit (or on all its neighboring locations) involves a tremendous waste on resources even if it can be temporarily used by other flows. Second, these approaches require routers to maintain a lot of state information regarding active and passive reservations. An improvement of this type of reservation is that the advance reservation is only performed on a neighboring location that will be visited. Therefore, advance reservation is only performed when a MN enters an overlapping zone [8]-[9]. In [8], Hierarchical Mobile RSVP (HMRSVP) integrates RSVP with the Mobile IP regional registration protocol and makes advance resource reservations only when a MN moves into the overlapping zone of the boundary cells of two different regions. To further reduce resource waste, B. Moon *et al.* [9] proposed that the advance reservation is performed when signal strength of its neighbor node is above a predetermined threshold TRSVP. As illustrated in Figure 3, a MN moves from AR_A to AR_B. HMRSVP performs the advance reservation from/to AR_B when the MN enters the overlapping zone, *i.e.* at point A, while the proposition of [9] triggers the advance reservation at point B. Obviously, by delaying the advance reservation, B. Moon's proposition can minimize resource waste. Moreover, it can increase the advance reservation's utilization, because the reserved advance reservation has more chance to be used (nearer to the AR_B domain, more chance that the MN handover to it). In this proposition, soft handover is considered, *i.e.* a MN can be in contact with more than one BS (or AP) at a time, typically used in CDMA system. When a MN detects that the signal strength of its neighbor node is above a predetermined threshold TRSVP, the reservation from/to this new location is performed. First, the MN sends a RSVP request message to the corresponding AR that acts as a proxy agent of the MN to reserve resources. If the MN is a data sender, the proxy agent sends a PATH message to the destination. The CRN that intercepts this message replies with a RESV message to reserve resources along the new path. If the MN is a data receiver, the proxy agent sends a router update message that is used to notify the CRN the handover of the MN. The CRN then initiates a PATH message to the proxy agent. The latter replies a RESV message to reserve resources along the new path.

In contrary, M. Chang *et al* [41] think, by using HMRSVP, the MN may suffer temporary disruption if the resource reservation latency exceeds the time duration for the MN to pass through the overlapping zone. The probability of this disruption increases when the MN performs inter-domain handover. Therefore, they propose that the adjacent domains should have the overlapping cell. A MN has to cross one or more overlapping cell(s) in order to handover to a new domain, and advance resource reservations for inter-domain handovers are processed while a MN resides in the overlapping cell to ensure that the MN has enough time to perform resource reservation before it handovers to the new domain.

From their propositions and arguments, we can observe that a critical point for the advance reservation is the time when the advance reservation should be started. This time is called advance reservation trigger time. If the advance reservation is performed too early, wasted reservations still exist. While if the advance reservation is performed too late, the reservation interruption, which means reservation from/to the new location is not yet established when the MN moves to this location, may occur. Hai *et al* [23] calculate

this trigger time by maximizing success probability which is the probability that MN's session can be terminated with success (not blocked due to lack of resources or resources not yet being reserved when the MN connects to the new location). In their calculation, the success probability depends on the blocking probability which is due to lack of resource, the probability that a MN performs handover, and the probability of reservation interruption (reservation interruption occurs when a MN connects to a new location but the reservations for this new location are not yet completed). They also noted that reservation delay impacts on advance reservation performance. That is, less reservation delay incurs more success probability.

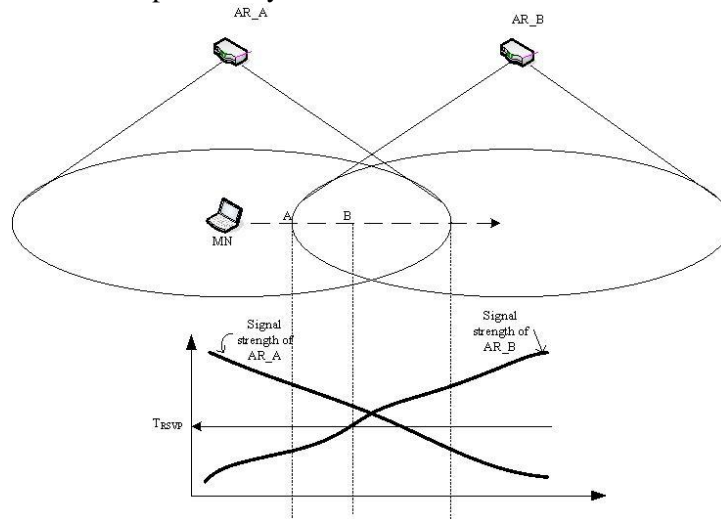


Figure 3. Advance Reservation by a Pilot Strength Threshold

The third strategy, instead of establishing advance reservation from the next AR to the destination, performs advance reservation from its current AR to next possible ARs [10] [42]. This method minimizes the amount of advance reservation (the advance reservations are made only on one-step paths from the current AR to next possible ARs) by introducing more active reservation (due to the increased path length, Figure 4). In these propositions, an anchored AR (AR1 in Figure 4) is introduced. The anchor intercepts firstly the packets addressed to a MN, and then forwards these packets to the next AR. The next AR in turn forwards to its next, until the current AR of the MN. The path from the anchored AR to the current AR by passing all intermediate ARs is called forwarding path, *e.g.* the path AR1->AR2->AR3->AR4 in the Figure. Obviously, the forwarding path cannot be too long. Two types of forwarding schemes are introduced in [10]: region-based and movement-based schemes. The former scheme restricts the forwarding path to be concatenated within the same domain. The latter scheme restricts the maximal length of the forwarding path to a certain threshold. Once the forwarding path is renewed, the new visited AR becomes the new anchored AR, meaning it will receive packets addressed to the MN, and should be responsible for forwarding these packets to the MN if the MN is attaching to another AR.

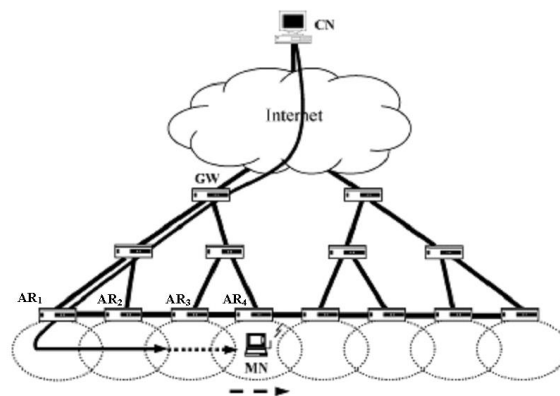


Figure 4. Advance Reservation with Forwarding Path

Instead of making full advance reservation, E. Alnasouri *et al.*[36] propose a semi-proactive scheme, in which the advance reservation is not actually made, but necessary information for reservation (*e.g.* required resource) is transfer first to those nodes which are potentially the next access routers for MN. This can reduce the reservation establishment once MN connects to a new access router. Also, in this paper[36], one step-forwarding path (between the current AR and previous AR) is established for temporary usage, and which is deleted after the resources are reserved between the new paths.

The above discussion is based on the assumption that there are available resources when perform advance reservations. We do not consider the case where advance reservations may be failed if they are done at each handover. In [40], author works on this issue in the case of both sender and receiver sides being mobile node and try to select the paths (between sender's ARs and receiver's ARs) on which advance reservation success probability can be maximized. They use semi-Markov chains and a binary linear programming formulation that takes into account handoff blocking probability of paths and estimated signalling delays for the path selection.

For advance reservation, since the MN is not yet present at the next location, it cannot be responsible for performing advance reservation. Advance reservation is executed by the reservation proxy [6], [19], [20] or the CRN [38] or the next AR [9]. According to our analysis, it is recommended that, in the case without reservation proxy, the CRN should be assigned to reserve resource for flows from the CN to the MN, while the AR should take the responsibility for the flows from the MN to the CN. With this method, the CRN or AR can send the reservation message (RESERVE message) immediately without discovering the data path in first. This will improve the advance reservation performance. The CRN is an important entity for reservation in advance. Also it plays an important role for release of obsolete reservation that will be discussed in the next section, in which we will also discuss how to discover the CRN before the MN's handover.

5. Release of Obsolete Reservation

Another crucial issue of resource reservation in mobile environments is to release obsolete reservations along the old path after the departure of a MN. As illustrated in Figure 2, after the departure of a MN, reservation states along the old path (the path between the PAR and the CRN) become obsolete, so they should be released as quickly as possible. It should be noted that normally the way in which MN releases these obsolete reservations just before it leaves the domain is not efficient, because this will release resources along the common path (the path between the CRN and the CN) as well, which need to be re-allocated once the MN connects to the NAR. In

mobility case, often it is desired to retain the old reservation along the common path in order to avoid resources being re-allocated.

In those RSVP extensions with reservation proxy [6]-[7],[19]-[20], it is the reservation proxy that takes the action to release the obsolete reservation. For example, in [20], the release of obsolete reservation is performed by the RSVP-MP upon detecting that the MN has connected to a new AR. The RSVP-MP sends a PathTear/ResvTear message toward the old LCoA, *i.e.* the old AR. The reservations along the old path, as well as the wireless reservation are then released. In the case where the notion of passive reservation is introduced [6]-[7][19], they consider that active reservation is never changed to obsolete reservation directly. The active reservation becomes first passive reservation after the departure of the MN. Then it becomes obsolete if the MN performs again handover, and which makes impossible for the MN to come back with just one handover. Since the reservation proxy is aware of the reservation state (active, passive or obsolete), it is responsible to release these obsolete reservations.

In the case where no reservation proxy exists, often it is the CRN that should initiate to release the obsolete reservations [1], [21]: the CRN receiving a new reservation message finds that an existed reservation state has already been created for this flow, but for a different interface, then sends a *teardown* messages (PathTear or ResvTear) to release the obsolete reservations. This method to release the obsolete reservations is simple and effective if the CRN is near to access routers. However, if the CRN is located far away (the situation for inter-domain handover), it may still take long delay to release the obsolete reservation states on the routers along the old path. In Figure 2, if router *n* is the CRN, it should wait for the new reservation message sent by MN from its new location after its handover; this message crosses the whole path from the NAR to router *n*. Then, upon receiving this message, router *n* sends a message to release the obsolete reservations along the old path. This message also crosses the whole path until the PAR. This makes the PAR incur the longest delay to tear down the obsolete reservation compared with the other routers along the old path. Unfortunately, the access router is often a bottleneck in mobility resource reservation case.

Even though we discuss that it is inefficient for the CRN to release the obsolete reservation when the CRN is far away from access routers, this mechanism is still relatively simple while RSVP is used for resource reservation. This CRN mechanism for resource release becomes complicated when the QoS NSLP is used. For security reason, QoS NSLP emphasizes “RESERVE message must only be sent towards the QoS NSIS Receiver (QNR)” [29]. This makes the obsolete reservation release based on the CRN method sophisticated. For example (Figure 2), if the reservations that the MN performs when it connects to its PAR are from itself to the CN (QNR), *i.e.* RESERVE message is sent from the MN to the CN. When it connects to the NAR, reservation states cannot be released in the direction from the CN to the MN. Instead, upon receiving the RESERVE message from a new location, the CRN issues a NOTIFY message towards the MN's old CoA. The previous AR stops the NOTIFY message, and sends RESERVE message (with Teardown bit set) towards the CN, to release the obsolete reservation [1]. Obviously, this increases significantly the complexity of ARs which should be enough intelligent to decide when the NOTIFY message should be forwarded (to MN) and when it should be stopped. Also each AR should be capable to issue the RESERVE message on behalf of the MN.

Alternatively, Lin *et al* [22] propose to change the timer of reservation state, so as to release the obsolete reservation quickly. The reservation state management of QoS NSLP is soft-state approach in which reserved resource state expires unless periodically refreshed by the receipt of a signaling message indicating that the state

should continue to remain installed. Two relative time parameters are maintained by soft state: one is the refresh period which serves to generate RESERVE message to refresh reservation state; the other is the local state's lifetime whose default value is 90 seconds [1]. They propose to give a new timer called HLT (Handover local state's LifeTime) to the soft state just before the MN leaves the current location. Since this timer is much smaller than default value, the obsolete reservations along the old path will be released quickly (upon the expiration of HLT) while the reservations along the common path are not expired before the arrival of the reservation messages from the new location. Then an optimal value for this new timer is calculated by minimizing the total cost of re-allocating resources along the common path and retaining obsolete reservations along the old path.

It is mentioned at the beginning of this section that it is not recommended for the MN to send a message to release the obsolete reservations just before the handover, since this will release also the reserved resources along the common path. However, if this teardown message is only forwarded along the old path and stopped at the CRN, the obsolete reservations are released quickly while the reservations along the common path are not released, and then can be re-used when the MN connects to the new location. This is not yet addressed in the literature, but we think it will be interesting to work on it. For this mechanism, the critical point is to find the CRN before handover. One of the possible methods is to use Mobile Intelligent Agent (MIA) [36][37]. MIAs are characterized by their ability to roam across wide-area networks, operate autonomously on nodes (within the running environments for the MIAs), and perform tasks on behalf of the MN. In [37], a MN issue a Forwarding-Point MIA to locate the Forwarding Points (*i.e.* the CRNs) for all location the MN may visit next. Another possibility [38] is to establish a route table to access routers on all involved QNEs. QNE can then determine if it is the CRN.

However, these methods to discover the CRN before handover procedure are in the early stage. Many problems need to be resolved in the future. We think that discovery of CRN before handover is an important task for reservation in mobility scenarios, because it can not only improve the performance of obsolete reservation release, but also the performance of advance reservation as discussed above. In the next session, we will discuss another reservation issue in the case where sets of nodes move collectively, we talk about NEMO mobility.

6. Reservation in Advance

In NEMO mobility, if mobile network nodes perform reservation individually, it would result in a high number of reservation states stored at routers from home agent to mobile router and a high number of signaling messages processed at these routers. To avoid this situation, aggregate reservation (or reservation aggregation) is proposed for NEMO mobility [25], [27],[24]. In [24], to support QoS-handover, a QoS incorporated registration process is performed whenever an MR enters the network mobility service domain. This incorporated registration is not only used for mobility management, but for resource reservation as well. The MR computes the aggregate QoS requirements of all mobile network nodes and sends the information to the BS. In this section, we will review current works on reservation aggregation.

Reservation aggregation is that a single aggregate reservation state is created for a set of individual flows that share the same routers in an "aggregation region" [43], as illustrated in Figure 5. While the reservation aggregation is applied in NEMO mobility, for those flows from mobile network to Internet (or those flows from Internet to mobile network), the mobile router (or the home agent, ingress router in Figure 5.), checks the request of each per-flow reservation and determines if this request can be aggregated, while the home agent (or the mobile router, egress router

in Figure 5.), de-aggregates per-flow reservation from aggregate reservation. The routers along the path from the mobile router to home agent only see the reservations for aggregation but not the individual reservations inside.

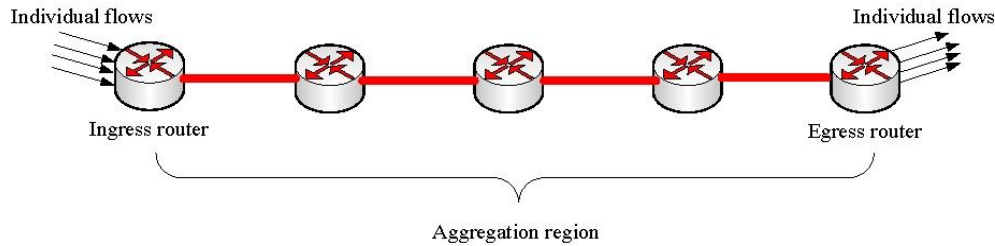


Figure 5. Reservation Aggregation

It should be noted that not all individual flows could be aggregated into a single aggregation. Only those flows that have the same or at least similar QoS requirement can be aggregated together [44]-[45]. Therefore, the ingress router should firstly check the individual reservation flow, and then decide to which aggregation this flow can be integrated. This procedure is called *flow partition*. Then the ingress router should create an *aggregated Traffic Specification (TSpec) of a reservation aggregation* which can capture the total requests of aggregated flows. Finally, a mechanism is required to avoid frequent adjustment of aggregation size. This is especial important when NEMO is implemented in public transportation vehicle, where individual flows change frequently and result in fluctuation on reservation aggregation. In consequence, tremendous signalling exchanges between the mobile router and the home agent, which in turn introduces an enormous burden on the intervening routers and incur bandwidth overhead. Therefore, [43] recommends that an *aggregation policy*, which “maintains the amount of bandwidth required on a given aggregate reservation by taking account of the sum of the bandwidths of its underlying end-to-end reservations while endeavouring to change it infrequently”, should be used in the reservation aggregation.

6.1. Flow Partition

Flow partition means to partition different flows into different aggregation, so as to maximize the aggregation performance. This is necessary because aggregating flows with quite different QoS requirement leads to more resources being reserved than the sum of individual reservations. Hence it is recommended that flows with the same (or similar) QoS requirement should be aggregated together [44], [45]. This implies, for improving aggregation performance, coexistence of multiple aggregate reservations on the same aggregate path. J. Ehrensberger [45] defines different aggregate reservations according to requests' delay budget, *i.e.* each aggregation is defined with an interval such that these intervals partition the range of possible values of the delay budget of all flows. An arrival flow chooses the aggregate reservation that corresponds with its delay budget. They also mention that, since the underlying flow profile can be varied, a better approach consists in mapping a new arrival to an aggregate reservation that minimizes the total reserved resources. This approach requires all possible mappings of the arrival flow to every aggregate reservation to be calculated firstly, the best mapping then is chosen. Schmitt, J. *et al.* [44] also point out that aggregated flows should have an equal or at least similar maximum delay requirement, and the aggregate reservations are ordered according to maximum delay requirement. That is to say, if aggregation *a* has smaller value of maximum delay than aggregation *b*, then all flows aggregated in aggregation *a* have smaller maximum delay than all flows aggregated in

aggregation b . However, there are still some problems left open, for example, how many aggregated reservations should be created according to individual flows; when should a new aggregate reservation be created instead of integrating the arrival flow into an existing one.

6.2. Aggregated Traffic Specification

It is known that a good capture of the Traffic Specification (TSpec) can optimize resource utilization. When individual flows are aggregated, one simple way to describe the aggregated traffic specification is to sum the individual TSpec, called summed TSpec. Schmitt, J. *et al.* [44] argue that using this summed TSpec wastes resources, because some reserved resources will never be used. Therefore, they propose a cascaded TSpec to describe aggregated traffic specification, and prove that an admission control based on this cascaded TSpec can optimize aggregate reservation. Besides this parameter-based admission control, Jingyu Qiu *et al.* [46] propose a measure-based admission control for aggregate reservation. They give the probability that no loss or delay bound violations will occur when admitting a new arrival into the aggregation as follows:

$$P \left\{ \max_{k=1,2,\dots,T} k\tau(R_k + r_k - C) \leq Cd \right\}$$

where $k\tau$ is the time interval during which the peak rate is measured, R_k is the current peak rate and r_k is peak rate of the new arrival flow, C is service rate and d is maximum delay requirement. In other words, this is the probability that, for any time interval, the amount of buffered packets is less than the authorized amount under the maximum delay requirement. Then according to this probability, the new arrival is determined whether to be admitted or not.

6.3. Aggregation policies

Normally, an aggregation should be adjusted upon every arrival and departure of its individual flows. However, this will, as described above, introduce an enormous burden on the intervening routers and incur bandwidth overhead. Therefore, the frequency of aggregation adjustment should be minimized. To do this, Huirong Fu *et al.* [47] propose to adjust the aggregate reservation periodically. At each adjustment, the maximum resources that will be requested in the next period are reserved, so excess resources should be reserved when performing adjustment. To predict this maximum resource requested in the next period, authors make a pattern of flow arrivals, namely flow arrivals consist of a sinusoid with random phase and additive white uniform noise [47]. This mechanism is limited in practice where such a pattern is difficult to match. In other words, it is difficult to predict the amount of maximum requests in the next period. In the case of prediction failure, a request will be rejected when all reserved resources are used. Another problem with this proposition is the choice of time period. If the time period is chosen to be large, more resource underutilization (due to excess reservation) or more rejection occurs, otherwise, the frequent adjustment problem arises again. Lin *et al* [26] calculate an optimal value for the amount of excess reservation by minimizing the total cost which consists of underutilization cost, signalling cost and request holding cost (due to adjustment). Then aggregation adjustment is only performed when the total requests of individual flows either exceed the aggregation's capacity (size) or go below a lower threshold (which is also calculated by minimizing the total cost). Likewise, at this adjustment, new excess reservation is performed and the new lower threshold is calculated.

Another way to reduce the frequency of aggregation adjustment is to set a waiting period, and then only one adjustment is performed by considering all requests received during this period. Two policies for setting the waiting period are proposed in [25], Temporal Operating Policy (TOP) and Cardinal Operating Policy (COP). The former configures the waiting period as a fixed time interval, the latter configures it as a varying time interval during which the received number of requests is fixed. All received requests during the waiting period are held until the time of the next adjustment. Then they are integrated together into the aggregation. One advantage of these policies is that no excess resource is reserved when performing adjustment. Intuitively, these policies incur little resource underutilization (underutilization is only caused by the departure of underlying flows). However, since all received requests should wait for the next adjustment, additional waiting cost is introduced by those policies. Other policies for aggregation size adjustment can be found in [27][50].

These adjustment policies discussed above suggest that reservation should not be adjusted on each session's arrival or departure. This results on unused resource reservations. In [49], these unused resources are sharing between non-real-time flows by using proposed Idle bandwidth sharing (IBS) algorithm. In this algorithm, if a new session only contains non-real-time flows, it uses the unused resources, and is set up directly without the need of resource reservation. To do this, A Non-Real-time Queue (NRQ) is allocated to buffer its packets in their proposition. In the case of the arrival of a session with real-time flows, normal procedure should be performed. That is, the session is held until the aggregation is adjusted.

The above discussing focus on the case of MR having single wireless interface. In a more complicate case where MR has multiple wireless interfaces, scheduling aggregate reservation in different interfaces is a challenge. S.Z. Ahmad *et al.* [41] use a multi-server scheduler to compensate such conditions with the aggregated capacity of multiple links.

In summary, when an ingress router (home agent or mobile router) performs aggregate reservation, it should firstly create different aggregations for partitioning individual flows. Then, for each aggregation, it creates aggregated traffic specification to capture the total requests of aggregated individual flows. Finally, excess reservations should be added to the aggregation (the aggregated traffic should take account of these excess reservations). With all these considerations, the reservation aggregation is then performed between the ingress router and the egress router.

7. Different Approaches for Reservations in Mobility Scenarios

7.1. Sender-initiated or Receiver-initiated

In mobility scenarios, always MN, rather than CN, is first aware of mobility. Obviously, it is expected that the MN initiates immediately the reservation after the handover, rather than firstly notify CN its mobility, and then the CN initiates the reservation process, because the latter scheme takes more time for resource reservation. And it is discussed that the reservation delay that impacts significantly on the reservation performance should be minimized.

To enable the MN to initiate the resource reservation, sender-initiated approach should be applied when the MN is data sender, while receiver-initiated approach should be applied when the MN is data receiver (note in this case, a data path should be pre-established before the MN's arrival, so the MN can perform immediately the reservation upon its arrival). Obviously, the RSVP that is designed to support only receiver-initiated approach is not suitable for this reservation performance optimization. Fortunately, QoS NSLP enables both the sender-initiated and the

receiver-initiated approaches. But it does not clarify how to decide which approach, sender-initiated or receiver-initiated, to be used. Therefore, a protocol of negotiation about which reservation approach being used should be designed, and this negotiation should be performed when establishing a session between two nodes. However, this is still an open issue.

It is clear, as described above, when a MN communicates with a fixed node, it is the MN that should initiate the resource reservation. However, a problem arises when two MNs communicate with each other. A simple solution is that both the sender-initiated and receiver-initiated are used for the same session. That is, when a data-sender MN performs handover, it immediately sends RESERVE message to reserve resources for the new location, *i.e.* sender-initiated reservation is applied. While a data-receiver MN performs handover, it also sends RESERVE message, *i.e.* receiver-initiated reservation is applied. However, QoS NSLP does not support this kind of reservation. Only one approach can be used for a flow, sender-initiated or receiver-initiated. Therefore, a reservation selection must be applied in this case. A possible selection strategy can be for example that the MN that performs more handovers should initiate reservation, but this is also an open issue.

7.2. Path-coupled or Path-decoupled

The above discussion on resource reservation considers that all reservations are performed with path-coupled approach, which means the reservation signalling message follows exactly the data path. Recently, people points out that the path-decoupled approach (in which reservation signalling does not follow exactly the data path, typically, it crosses certain central reservation control entities) have some advantages on the path-coupled approach in the case of reservation for mobility [49].

First, the setup delay of path-decoupled reservation is much less than that of path-coupled reservation. This characteristic comes from the fact that the path-decoupled approach can install reservation states at all involved routers simultaneously, while, in path-coupled approach, a router begins installing reservation state only after the end of its previous router's reservation process. We have mentioned that less reservation setup delay brings better performance in advance reservation. Therefore, path-decoupled reservation can improve reservation performance.

Second, in path-decoupled reservation, a central QoS control is responsible for policy control (determines whether the user is authorized to make the reservation) and admission control (determines whether the network has sufficient available resources to supply the requested QoS). This removes policy control and admission control from routers along the data path, and which makes the routers more efficient to forward data packets. Moreover, the policy control and admission control, when they are moved to the QoS control, are performed more efficiently. With the path-decoupled approach, the policy control is executed only once as long as the MN stays in its domain, while with the path-coupled reservation, the policy control should be performed each time a MN moves. Since the central QoS control possesses some necessary information to make policy control immediately without inquiring after each router the available capacity, the procedure duration can be minimized.

However, the central QoS control entity is a complicated entity. To manage resource reservation, it should be aware of the actual network topology and routing information, which introduces the scalability problem. Moreover, the path-decoupled approach is not supported in RSVP, and NSIS working group does not yet consider this approach either. Yet this approach begins to attract the attention from people working on resource reservation for mobility scenarios. NSIS working group will probably include this approach for their future work.

8. Conclusion

Providing QoS guarantee to mobile users is one of the important requirements for the current Internet evolution. Real-time applications require the end-to-end resources to be explicitly reserved. However, reservation protocols are originally designed for wired networks (even though QoS NSLP takes account of some mobility characteristics, it is still original for wired networks). This raises some issues when these protocols are applied in mobility scenarios. The survey overviews these raised issues and the related works addressed to them.

The interaction with the mobility protocol makes the reservation protocol aware of the MN's mobility. Without the advance reservation consideration, most previous works let the reservation be triggered once the handover procedure is ended. However, as we analysed, the advance reservation is the most important task to achieve seamless QoS guarantee in the case of mobility. The reservation should be triggered before or during handover procedure, so the resources for the next location can be reserved before the arrival of the MN. We also observed that, to perform reservation in advance, a reservation proxy is often introduced for this task. It should be noted both RSVP and QoS NSLP do not consider proxy reservation. These reservation protocols need to be extended for proxy reservation. But up to now, no work has been carried out for this end. Release of obsolete reservation is relatively simple when RSVP is used, while this becomes more complicated when QoS NSLP is applied, because NSIS assumes that security should be firstly considered. Then it limits the signalling forwarding direction, which complicates the release procedure. The aggregation issue arises in NEMO scenarios. It should be carefully handled, since in some situation aggregation incurs more resources to be consumed than individual reservation.

QoS NSLP is more suitable than RSVP, not only because it originally takes account of mobility characteristic, but it is more flexible as well. We also discuss that, in some situations, it is desired to use the sender-initiated approach than the receiver-initiated approach, and the path-decoupled approach than the path-coupled approach. QoS NSLP is designed by considering all approaches, while RSVP can only be the receiver-initiated and the path-coupled.

Clearly, several unresolved issues remain to this date. Therefore, it is hoped that this survey will serve as a reference on the most relevant technical issues and that it will assist the interested reader in identifying the most challenging and interesting parts of the design of resource reservation for mobility scenarios.

References

- [1] C. Perkins, "IP Mobility support for IPv4, Revised" IETF RFC 5944, (2010).
- [2] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6", IETF RFC 6275, (2011).
- [3] R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, (1994).
- [4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, (1998).
- [5] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification", RFC 2205, (1997).
- [6] A. Talukdar, B. Badrinath and A. Acharya, "MRSVP: a resource reservation protocol for an integrated services network with mobile hosts", *The Journal of Wireless Networks*, vol. 7, (2001), pp. 5–19.
- [7] W. Chen and L. Huang, "RSVP mobility support: a signaling protocol for integrated services internet with mobile hosts", in: *Proceedings of INFOCOM'00*, IEEE, vol. 3, (2000), pp. 1283–1292.
- [8] C. Tseng, G. Lee and R. Liu, "HMRSVP: A Hierarchical Mobile RSVP Protocol", *ACM/Baltzer Wireless Networks*, vol. 9, (2004), pp. 95-102.
- [9] B. Moon and A. H. Aghvami, "Quality-of-Service Mechanisms in All-IP Wireless Access Networks", *IEEE Journal on Selected Areas in Communication*, vol. 22, iss. 5, (2004), pp. 873-888

- [10] S. C. Lo, G. Lee, W. T. Chen and J. C. Liu, "Architecture for Mobility and QoS Support in All-IP Wireless Networks", *IEEE Journal on Selected Areas in Communication*, vol. 22, iss. 4, (2004), pp. 691-705.
- [11] T. Sanda, X. Fu, S. Jeong, J. Manner and H. Tschofenig, "NSIS Protocol Operation in Mobile Environments", *IETF RFC 5980*, (2011).
- [12] M. Stiemerling, H. Tschofenig, C. Aoun and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", *IETF RFC 5973*, (2010).
- [13] H. Chaskar, "Requirements of a Quality of Service (QoS) Solution for Mobile IP", *IETF RFC 3583*, (2003).
- [14] A. Terzis, J. Krawczyk, J. Wroclawski and L. Zhang, "RSVP operation over IP tunnels", *IETF RFC 2746*, (2000).
- [15] C. Shen, H. Schulzrinne, S. Lee and J. Bang, "NSIS Operation Over IP Tunnels", *IETF RFC 5979*, (2011).
- [16] Y. Sun, Y. C. H. Zhang, Y. L. Song and E. Dutkiewicz, "Fast RSVP: Efficient RSVP Mobility Support for Mobile IPv6", *Springer Journal of Wireless Personal Communications*, vol. 60, no. 4, (2011), pp. 769-807
- [17] H. Soliman, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", *IETF RFC 4140*, (2005).
- [18] R. Koodli, "Mobile IPv6 Fast Handovers", *IETF RFC 5268*, (2008).
- [19] N. Huang and W. Chen, "RSVP Extensions for Real-time Services in Hierarchical Mobile IPv6", *ACM/baltzer Mobile Networks and Applications*, vol. 8, iss. 6, (2003), pp. 625-634.
- [20] S. Paskalis and A. Kaloxylou, "An Efficient RSVP-Mobile IP Interworking Scheme", *ACM/baltzer Mobile Networks and Applications*, vol. 8, iss. 3, (2003), pp. 197-207.
- [21] L. Peters, I. Moerman, B. Dhoedt and P. Demeester, "Q-MEHROM: Mobility support and resource reservations for mobile senders and receivers", *Elsevier Computer Networks*, vol. 50, iss. 8, (2006), pp. 1158-1175.
- [22] H. Lin and H. Labiod, "Release of Unnecessary Resource Reservation in Mobility case", *Proceedings of IEEE ISCC'08, Marrakech*, (2008).
- [23] H. Lin, and H. Labiod, "Advance reservation optimization in IP-based mobile networks", *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, Canary Islands Spain*, (2009).
- [24] V. Devarapalli, R. Wakikawa, A. Petrescu and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", *RFC 3963*, (2005).
- [25] M. A. Malik, S. S. Kanhere, M. Hassan and B. Benatallah, "On-Board RSVP: An Extension in On-Board IP Networks" in *Springer LNCS Series, A. S.*, vol. 3326, (2004), pp. 264-275
- [26] H. Lin and H. Labiod, "RVP: A New Policy for Aggregate Reservation", *Proceedings of IEEE Globecom 2008, New Orleans*, (2008).
- [27] J. Wang, Y.Hsu and C. Tseng, "A Mobile Bandwidth-Aggregation Reservation Scheme for NEMOs", *Wirel. Pers. Commu.*, vol. 44, (2008), pp. 383-401.
- [28] C. W Lee, M. C Chen and Y. S. Sun, "Protocol and architecture supports for network mobility with QoS-handover for high-velocity vehicles", *Springer journal of Wireless Networks*, (2012).
- [29] R. Hancock, "Next Steps in Signaling: Framework", *RFC 4080*, (2005).
- [30] H. Schulzrinne and R. Hancock, "GIST: General Internet Signaling Transport", *IETF RFC 5971*, (2010).
- [31] J. Manner, G. Karagiannis and A. McDonald, "NSLP for Quality-of-Service Signaling", *IETF RFC 5974*, (2010).
- [32] G. Ash, A. Bader, C. Kappler and D. Oran, "QoS NSLP QSPEC Template", *IETF RFC 5975*, (2010).
- [33] A. Belhou, Y. A. Şekercioğlu and N. Mani, "Mobility-Aware RSVP: A framework for improving the performance of multimedia services over wireless IP-based mobile networks", *International Journal of Communication Systems*, vol. 32, iss. 4, (2009), pp. 569-582
- [34] S. Elleingand and S. Pierre, "FH-RSVP scheme for intra-site handover in hierarchical mobile IPv6 networks" *Elsevier Computer Communications*, vol. 30, iss. 2, (2007), pp. 416-427.
- [35] N. V. Lopes, M. J. Nicolau and A. Santos, "A QoS-enabled resource management scheme for F-HMIPv6 micro mobility approach", *Springer Journal of Telecommunication Systems*, vol. 52, no. 1, (2013), pp.341-357.
- [36] A. Karmouch and V. A. Pham, "Mobile Software Agents: An Overview", *IEEE Comm. Magazine*, vol. 36, no. 7, (1998).
- [37] J. T. Wang, J. S. Yang and C. C. Tseng, "An intelligent agent-based mobile resource reservation scheme", *Proceedings of 11th International Conference on Parallel and Distributed Systems, Fuduoka, Japan*, (2005).
- [38] S. Lee, M. Kim, K. Lee, S. Seol and G. Lee, "Seamless QoS Guarantees in Mobile Internet Using NSIS with Advance Resource Reservation", *Proceedings of the 22nd international Conference on Advanced information Networking and Applications AINA*, (2008).
- [39] E. Alnasouri, A. Mitschele-Thiel and A. Diab, "Handling Mobility Management and QoS Aspects in All-IP Networks", *Springer Journal of Mobile Networks and Applications*, vol. 16, (2011), pp. 83-95

- [40] E. S. Silva, J. Farines and M. Diaz, "A Path Selection Criterion for Advance Resource Reservation", *New Technologies, Mobility and Security (NTMS), IFIP International Conference on*, (2011).
- [41] M. Chang, M. Lee and H. Lee, "An RSVP based seamless resource reservation scheme for wireless mobile networks", *Elsevier Computer Communications*, vol. 30, iss. 1, (2006), pp. 81-92.
- [42] G. C. Lee, T. P. W. C. C. T., "Resource Reservation with Pointer Forwarding Schemes for the Mobile RSVP", *IEEE Communication Letters*, vol. 5, iss. 7, (2001), pp. 298-300.
- [43] F. Baker, C. Iturralde, F. L. Faucheur and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", *IETF RFC 3175*, (2001).
- [44] J. Schmitt, M. Karsten, L. Wolf and R. Steinmetz, "Aggregation of guaranteed service flows", *Proceedings IEEE IWQoS 1999, London*, (1999), pp. 147-155
- [45] J. Ehrensberger, "Resource demand of aggregated resource reservations", *Proceedings ECUMN 2000, Colmar, France*, (2000), pp. 56-61.
- [46] J. Qiu and E. W. Knightly, "Measurement-based admission control with aggregate traffic envelopes", *IEEE/ACM Transaction on Networking*, vol. 9, (2001), pp. 92-101.
- [47] H. Fu and E. W. Knightly, "A Simple Model of Real-Time Flow Aggregation", *IEEE/ACM Transactions on Networking*, vol. 11, (2003), pp. 85-93
- [48] S. Z. Ahmad, M. A. Qadir, M. S. Akbar and A. Bouras, "Analysis of Multi-Server Scheduling Paradigm for Service Guarantees during Network Mobility", *Springer Journal of Wireless Personal Communication*, vol. 63, (2012), pp. 177-197.
- [49] J. Hillebrand, C. Prehofer, R. Bless and M. Zitterbart, "Quality-of-Service Signaling for Next Generation IP-based Mobile Networks", *IEEE Communications Magazine*, vol. 42, no. 6, (2004), pp. 72-79.
- [50] G. Kamel, A. Mihailovic and A. H. Aghvami, "Case analysis of a Cost-Optimal QoS aggregation policy for network mobility", *IEEE Communications Letters*, vol. 12, iss. 2, (2008), pp.130 – 132
- [51] J. Liao, Q. Qi, T. Li, Y. Cao, X. Zhu and J. Wang, "An optimized QoS scheme for IMS-NEMO in heterogeneous networks", *International Journal Commun. Syst.*, vol. 25, (2012), pp. 185-204.

