Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

A Policy-Based Resource Reservation Service for Maritime Tactical Networks

David Kidston and Isabelle Labbé

The work described in this document was sponsored by the Department of National Defence under Work Unit 15CV.

Defence R&D Canada – Ottawa
TECHNICAL MEMORANDUM
DRDC Ottawa TM 2007-006
January 2007

Canada

A Policy-Based Resource Reservation Service for Maritime Tactical Networks

David Kidston
Communications Research Centre

Isabelle Labbé
Communications Research Centre

The work described in this document was sponsored by the Department of National Defence under Work Unit 15CV

Defence R&D Canada – Ottawa

Technical Memorandum

DRDC Ottawa TM 2007-006

January 2007

Abstract

As part of a research effort to provide enhanced communications capabilities in a maritime tactical network, a Policy-Based Traffic Management System was developed to dynamically control link usage and perform end-to-end QoS. A desirable management service in the maritime network is end-to-end guaranteed QoS for critical application flows. This type of service is most commonly useful for real time applications (such as VOIP or video), but can also be used for critical data transfers (such as a specific image transfer or chat session). By entering a service request, an operator can specify the application flow to be considered, the desirable treatment (currently in terms of bandwidth only), as well as any other pertinent information including the priority of the request.

A policy-based resource reservation service has been developed that guarantees bandwidth for an application flow by per-hop admission control. This document describes the resource reservation service (RRS) including the design decisions and the algorithms developed to support it. The RRS is organized around a main admission control algorithm that consists of four phases. In the first phase, global link information is used to generate potential routes between the source and destination of the requesting flow. The second phase probes the potential routes to determine if sufficient resources are available on all links. In the third phase, an acceptable path is selected and committed. Finally in the fourth phase the reservation is maintained until the flow terminates or the network can no longer support its requirements.

Résumé

Dans le cadre des travaux de recherche sur l'amélioration des capacités de communication des réseaux tactiques maritimes, on a mis au point un système de gestion du trafic axé sur les politiques qui permet de contrôler de façon dynamique l'usage des liens ainsi que d'assurer la qualité de service (QS) de bout en bout du réseau. Parmi les services de gestion du trafic souhaitables, voire même nécessaires dans un réseau maritime, est celui d'un service de réservation qui garantit une QS de bout en bout pour les flux d'applications jugés critiques. Ce type de protection est surtout utile pour les applications en temps réel (comme la voix sur IP ou la vidéo), mais il peut aussi servir aux transferts des données importantes (comme le transfert d'images ou une session de clavardage). Les demandes de réservation sont faites en précisant le flux d'application à prendre en considération, le traitement souhaité (seulement du point de vue de la bande passante pour l'instant) ainsi que tout autre renseignement essentiel, dont le degré de priorité de la demande.

On a élaboré un service de réservation des ressources axé sur les politiques qui utilise le contrôle d'admission distribué pour garantir la bande passante pour un flux d'application critique. Le présent document donne une description détaillée de ce service de réservation des ressources (SRR) et des algorithmes qui ont été développés. Le SRR repose sur un algorithme de contrôle d'admission qui comprend quatre étapes. Premièrement, il utilise l'information générale des liens de communication pour déterminer les divers chemins possibles entre la source et la destination du flux visé par la demande. Deuxièmement, il sonde ces chemins pour savoir si chaque segment dispose des ressources nécessaires. Troisièmement, il choisit et réserve un chemin acceptable. Finalement, il maintient la réservation jusqu'à la fin du flux ou aussi longtemps que le réseau peut satisfaire les exigences de la demande.

Executive summary

The management of maritime mobile networks represents a challenge for several reasons. Maritime networks are composed of heterogeneous links that are error-prone, variable, and offer only low bandwidth communication capacity. This implies that traffic should be directed over the type of bearer that supports it, make best use of the capacity that is available, and limit network management traffic to an efficient signalling protocol while being robust to transmission errors. In addition to these physical limitations, there are the dynamic network topology changes caused by the mobility of the naval units (nodes), and the security considerations arising from the use of a wireless medium that add to the complexity of the management tasks. Also, the hierarchical command structure of maritime deployments must be respected to ensure that control is not completely centralised so that some levels of network control are available to the commander of each maritime node. Finally, the limited availability of skilled network operators in the navy must also be considered. Providing the ability to carry out the control and management of communication resources in the most automated and transparent manner will minimise the human resource burden and the skill level required from the operation personnel.

In order to deal with these issues, a Policy Based Traffic Management (PBTM) system has been developed based on policy-based management and service-oriented approaches. The resource reservation service (RRS) is one of the four management services that were initially developed to support traffic management in the maritime environment. The RRS is described in this document. This service has been developed to guarantee bandwidth for an application flow by per-hop admission control in a maritime network. Not only does the RRS have to coordinate its operation with existing routing and QoS systems, but it must operate with low bandwidth, high reliability, and of course securely. The issues for admission control specifically related the maritime environment include the calculation of the available link bandwidth (especially for line of sight radio links), and link failures due to node mobility.

The RRS is organized around a main admission control algorithm that consists of four phases. In the first phase, global link information is used to generate potential routes between the source and destination of the requesting flow. The second phase probes the potential routes to determine if sufficient resources are available on all links. In the third phase, an acceptable path is selected and committed. Finally in the fourth phase the reservation is maintained until the flow terminates or the network can no longer support its requirements.

The first phase of the algorithm is mainly concerned with the generation of one or more routes from the source to the destination generated when a new resource reservation has been entered by the user. In order to generate routes on demand, the network topology is constantly updated by monitoring the OSPF link database which gives the routers current view of the available links in the network. Using this topology, one of three simple routing algorithms is used.

In phase two, probes are sent from the source to destination along the route(s) generated in the first phase. The admission control protocol in this work is similar to RSVP but modified for the maritime environment. Admission Control decisions are performed at each hop along the selected route(s). Probes travel on their assigned route from the source to destination one node at a time. The network resources along a path are evaluated at each node along the route. Currently the only resource considered is bandwidth. If sufficient resources exist to meet the reservation request stored in the probe for the desired link at the current node, the residual bandwidth of the link is noted in the probe and forwarded to the next node. If insufficient resources are found at a node, the probe is still forwarded as long as there are enough lower priority flows that could be pre-empted to free enough resources to meet the needs of the current request. Otherwise a fail message is sent to the destination.

The final decision of which of the successful probe's route to commit is up to the destination in phase three. When the destination has decided upon a route, a confirmation message is sent back along that route with each RRS updating the configuration of the router so that the flow is treated in the reserved class. In order to determine if a new request should be committed, the system again examines the other reservations that are using the local links. If enough unallocated resources remain in the pool, the resource allocation is enforced on the router. If insufficient resources are available, preemption is attempted as in phase two and if successful the reservation is enforced. Otherwise a failure message is sent to the source and no changes are made at the router.

When a confirmation message reaches the source, the resource reservation enters its active maintenance phase four. The reservation remains active until the request expires or fails. Requests can be terminated by a number of events including termination by the user, end of the reserved period, preemption by a higher priority flow, or failure of a link on the reserved route.

This protocol has several advantages including: load balancing, which is achieved through the use of multiple probes and the preference for large residual bandwidth; fault-tolerance, which is achieved through the use of time-outs and acknowledgements, bidirectional reservations, which allow reservations to be made in both directions at once, and reservations made in advance, which allow reservations to be organized before they are required. A preliminary simulation shows that the scheme is effective in this environment, though it has yet to be compared to RSVP.

While this protocol provides the basic services required for an efficient and robust resource service for critical flows in maritime networks, there are still many areas for future work in this area. Many of the algorithms were selected for simplicity of design where more advanced offerings should be tested where available. Similarly many optional features mentioned in the document remain to be investigated.

David Kidston, Isabelle Labbé. 2007. A Policy-Based Resource Reservation Service for Maritime Tactical Networks. DRDC Ottawa TM 2007-006. Defence R&D Canada-Ottawa

Sommaire

La gestion des réseaux mobiles maritimes constitue un défi pour plusieurs raisons. Les réseaux maritimes sont composés de liens de communication hétérogènes qui offrent de hauts risques d'erreur de transmission, qui sont variables et qui offrent une capacité de communication à bande passante étroite. Outre les limites physiques, les changements dynamiques de la topologie des réseaux causés par la mobilité des unités navales (nœuds) et les considérations de sécurité découlant de l'utilisation d'un support sans fil accroissent la complexité. De plus, les appareils de communication sont fabriqués par différentes entreprises qui n'ont pas la gestion des réseaux intégrés à l'esprit lors de la conception.. De plus, la structure de commandement hiérarchique des déploiements en mer doit être respectée. Cependant, le contrôle ne doit pas être qu'uniquement centralisé et un certain degré de contrôle doit aussi être laissé aux commandants de chaque navire (nœud). Finalement, la rareté des opérateurs de réseaux dans le milieu maritime est un autre élément important.

Dans le but de résoudre ces problèmes, on a mis au point un système de gestion du trafic qui combine l'approche de la gestion axée sur les politiques et l'approche orientée vers les services. Le service de réservation des ressources (SRR), qui est abordé dans ce document, est l'un des quatre services de gestion qui a été élaboré pour soutenir la gestion du trafic dans le contexte maritime. Ce service a été conçu afin de garantir la bande passante pour un flux d'application et utilise un contrôle d'admission distribué. Le SRR doit non seulement coordonner son fonctionnement avec le protocole de routage et les mécanismes de qualité de service (QS) disponibles, mais il doit aussi être robuste, utiliser une faible bande passante et respecter les principes de sécurité des communications. Les principaux problèmes reliés à l'environnement maritime et avec lesquels le contrôle d'admission doit composer sont notamment la détermination de la bande passante disponible (particulièrement pour les liens radio en visibilité directe) et la disparition de liens causée par la mobilité des nœuds.

Le SRR repose sur un algorithme de contrôle d'admission qui comprend quatre étapes. Premièrement, il utilise l'information générale des liens de communication pour déterminer les divers chemins possibles entre la source et la destination du flux visé par la demande. Deuxièmement, il sonde ces chemins pour savoir si chaque segment dispose des ressources nécessaires. Troisièmement, il choisit et réserve un chemin acceptable. Finalement, il maintient la réservation jusqu'à la fin du flux ou tant que le réseau peut satisfaire les exigences de la demande.

La première étape de l'algorithme consiste principalement à déterminer au moins un chemin entre la source et la destination précisée par l'utilisateur lors de sa demande de réservation. Pour permettre la détermination des chemins sur demande, la topologie du réseau est continuellement mise à jour grâce à la surveillance de la base de données des liens OSPF qui fournit l'état actuel des liaisons disponibles du réseau. On applique sur cette topologie, l'un des trois algorithmes d'acheminement.

Au cours de la deuxième étape, on envoie des sondes de la source à la destination en empruntant les divers chemins déterminés à la première étape. Le protocole de réservation qui a été développé ressemble au protocole RSVP, mais il est adapté à l'environnement maritime. Les décisions de contrôle d'admission sont prises à chaque nœud, tout au long des chemins sélectionnés. Les sondes parcourent leur chemin de la source à la destination un nœud à la fois. Les ressources sont évaluées à chaque nœud, tout au long du parcours. La bande passante est la seule ressource considérée pour l'instant. Si les ressources du nœud satisfont aux exigences de la demande de réservation stockée dans la sonde, alors la bande passante résiduelle de la liaison est notée dans la sonde qui est ensuite acheminée au nœud suivant. Si les ressources du nœud sont insuffisantes, alors la sonde est tout de même acheminée, à condition qu'il y ait assez de flux moins prioritaires afin de permettre une préemption et de libérer les ressources nécessaires pour répondre aux besoins de la demande. Sinon, un message d'échec est transmis à la destination.

À la troisième étape, la destination choisit le meilleur chemin parmi ceux franchis avec succès par les sondes. Quand son choix est fait, un message de confirmation est renvoyé par le chemin choisi. Ceci permet à chaque SRR de chaque nœud de mettre à jour la configuration du routeur pour que le flux soit traité dans une classe réservée. Dans le but de déterminer s'il faut faire une nouvelle demande, le système examine de nouveau les autres réservations qui utilisent le même lien local. S'il y a assez de ressources disponibles, alors l'attribution des ressources est effectuée dans le routeur. S'il n'y a pas assez de ressources disponibles, on a recours à la préemption, comme à la deuxième étape. Si la procédure réussit, la réservation est mise en application; si la procédure échoue, un message d'erreur est envoyé à la source, et aucune modification n'est faite au routeur.

Quand la source reçoit le message de confirmation, la procédure de réservation des ressources passe à la quatrième étape, celle du maintien actif. La réservation reste active jusqu'à l'expiration ou à l'échec de la demande. Les demandes peuvent être interrompues par un certain nombre d'événements, dont l'annulation par l'utilisateur, la fin de la période de réservation, la préemption par un flux plus important ou la disparition d'un lien de communication sur le chemin réservé.

Ce protocole comporte plusieurs avantages : la juste distribution du trafic sur plusieurs liens, qui est rendu possible par l'utilisation de nombreuses sondes et la sélection du chemin avec la bande passante résiduelle la plus élevée; la tolérance aux fautes, qui se concrétise grâce à l'utilisation de délais d'expiration et d'accusés de réception; la réservation bidirectionnelle, qui permet de faire des réservations dans les deux directions à la fois; et la réservation à l'avance, qui autorise l'organisation des réservations avant qu'elles ne deviennent nécessaires. Une première simulation a démontré que ce modèle est efficace et offre une bonne performance dans l'environnement maritime. Ce modèle n'a toutefois pas encore été comparé au protocole RSVP.

Malgré le fait que le protocole développé assure un service des réservation robuste et efficace pour les flux cruciaux des réseaux maritimes, certains aspects peuvent être encore améliorés. Certains des algorithmes ont été choisis pour leur simplicité, mais il

faudrait aussi mettre à l'essai des solutions plus avancées. De la même manière, certaines fonctions optionnelles mentionnées dans ce document doivent encore être approfondies.

David Kidston, Isabelle Labbé. 2007. A Policy-Based Resource Reservation Service for Maritime Tactical Networks. DRDC Ottawa TM 2007-006. Recherche et développement pour la défense Canada - Ottawa

This page intentionally left blank.

Table of contents

List of figures

List of tables

Acknowledgements

The authors would like to thank the Directorates of Maritime Requirements Sea (DMRS) and the Maritime Ship Support (DMSS) of the Department of National Defence, for advice provided and technical discussions on maritime tactical networks.

This page intentionally left blank.

1. Introduction

As part of a research effort to provide enhanced communications capabilities in a maritime tactical network, a Policy-Based Traffic Management System (PBTM) [1] was developed to dynamically control link usage and perform end-to-end QoS. The system combines the service-oriented architecture (SOA) and policy-based network management approach.

Four traffic management services have been implemented as part of this system. These services are: access control, traffic prioritization, trunk utilization, and resource reservation. While the first three of these services improve the operation of mobile networks, the ability to manage QoS for critical communications requires additional support. For this reason the fourth service was developed to provide per flow guaranteed end-to-end QoS. This resource reservation service uses a policy-based admission control protocol to support priority and preemption as well as hard bandwidth guarantees. This technical note expands on the description of the resource reservation service (RRS) given in [1]. It describes the reservation protocol as well as the admission control algorithms that were developed and implemented to support the service.

The RRS is organized around a main admission control algorithm that consists of four phases. In the first phase, global link information is used to generate potential routes between the source and destination of the requesting flow. The second phase probes the potential routes to determine if sufficient resources are available on all links. In the third phase, an acceptable path is selected and committed. Finally in the fourth phase the reservation is maintained until the flow terminates or the network can no longer support its requirements. A simulation of this service has been completed and initial results are presented in this document.

This report begins in Section 2 with a brief outline of the issues and assumptions made in the design of the service. This is followed in Section 3 by an overview of the admission control algorithm and mechanisms which were used. The mechanisms used in each of the four phases of the main algorithm are described in Sections 4 through 7 respectively. The report continues in Section 8 with a summary of the advantages of this approach. It then presents some preliminary simulation results in Section 9, with future work in Section 10 and then concludes in Section 11.

2. Assumptions and Issues

In order to develop the main admission control algorithm for the RRS several assumptions about the target network's routing and QoS capabilities were made. Assumptions have also been made about the policy services provided by the underlying policy-based network management system (outlined in [1]). This section describes the underlying network in terms of these systems. Section 2.1 provides an overview of the maritime environment for which this service has been developed. Section 2.2 gives a brief overview of the policy-based management framework upon which the reservation service is designed to operate. Finally, Section 2.3 discusses the issues involved in providing an admission control in a maritime environment.

2.1 The Maritime Environment

The maritime tactical environment imposes unique limitations on information sharing. Maritime units operate in a low bandwidth environment over heterogeneous communications media. The efficient use of bandwidth and exchange of information is essential to successfully complete the mission. A typical maritime network is shown in Figure 1.

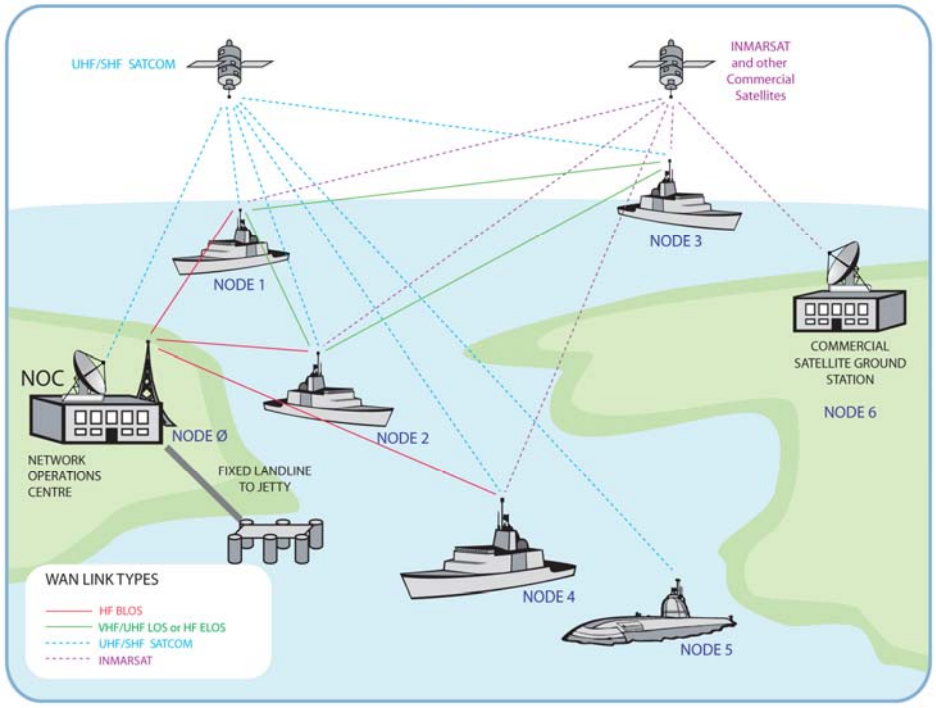


Figure 1. *A Typical Maritime Mobile Network*

A maritime network is composed of a variety of strategic and tactical communications links. The communications bearers that are available to transfer information within the network are: commercial satellite (e.g. INMARSAT B), ship-to-ship satellite nets (UHF SATCOM, SHF SATCOM), HF extended and beyond line-of-sight (HF ELOS/BLOS) and UHF/VHF Line-of-sight (LOS).

A maritime network is characterized by the following;

- moderate mobility speed (dynamic topology),
- error-prone, low and variable bandwidth communications links,
- a NOC where high-level management decisions are made,
- nodes with a variety of communications equipment,
- the communications network is multi-hop
- limited availability of network operators, and
- additional security considerations due to the wireless medium,

Several additional assumptions have been made about this environment;

- the routing protocol is OSPF,
- links are bidirectional (a requirement of OSPF),
- mobility is slow enough that OSPF will converge (moderate mobility),
- diffserv QoS mechanisms are supported within the IP-based network, and
- it is initially assumed the network is comprised of a single administrative domain (both security and management)

These characteristics must be accommodated by any network management service designed for operation in the maritime environment.

2.2 Underlying Policy-Based Management Architecture

The service oriented policy-based network management architecture upon which this service is based is described in [1] and presented in Figure 2.

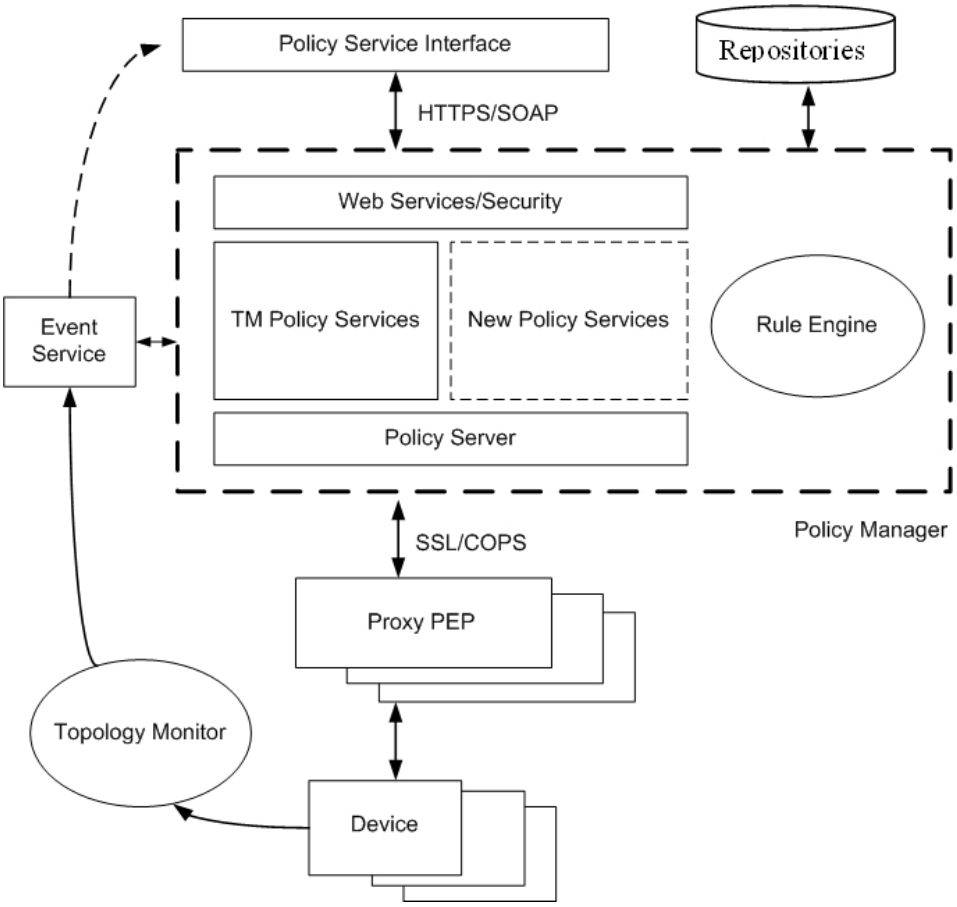


Figure 2. *High level System Architecture*

The main components of the architecture are: the policy service interface, which accepts policy from operators while assigning roles to devices; the policy manager, which interprets high level policy and pushes low level commands out to policy-enabled resources (also known as PEPs); the proxy PEP, which takes the low level policy and configures their associated devices to conform with policy; repositories, which store the high level policies; a topology monitor, which notes changes in the network connectivity; and finally an event service, which helps to distribute events from the policy system and the underlying network (topology monitor).

This architecture supports the resource reservation service (RRS) by providing policy supported decisions about potential routing of the admitted flows. The management system that was developed is based on the premise that diffserv alone will provide prioritization for non critical traffic while the RRS will provide an intserv-like mechanism, similar to RSVP, for critical flows. Access to the bandwidth, reserved for use by the RRS, will be limited using the admission control protocol described in this document. The diffserv configuration on the network devices (routers) will remains relatively static (configured by the policy system) with applications assigned to those classes based on policy. The configuration required to support the RRS is done on the

fly by the policy system when the admission control protocol successfully allocates resources.

2.3 Issues for Admission Control

There are several issues that must be dealt with when designing an admission control algorithm for the RRS. Not only does the RRS have to coordinate its operation with existing routing and QoS systems, but it must operate with low bandwidth, high reliability, and of course securely.

The characteristic that is considered when determining whether a particular link can handle a new service request is the residual bandwidth. When the link is dedicated and provides stable bandwidth the residual bandwidth can be determined by looking at the bandwidth available to be reserved and the amount currently reserved. This is the case in the maritime environment for most satellite and BLOS HF links where the media is not shared (point-to-point type of link) and in operation can often be characterised as either available at full capacity or not available (binary). Residual bandwidth is then directly calculable by subtracting the reserved bandwidth from the reserveable pool (currently a policy-defined percentage of the nominal transmit capacity). Thus residual bandwidth = (policy-defined pool multiplier[1] * nominal link capacity) – capacity currently reserved.

However, for LOS (UHF/VHF) links the media is shared and residual bandwidth cannot be reliably determined. There are currently no standards for QoS support in the MAC of these media. Attempts to implement QoS in these environments would include probing, and cross-layer communication or instrumentation of the MAC with proprietary SNMP MIBs. These methods are also not standardised and any attempt at direct measurement is likely to introduce significant overhead. It is currently assumed that LOS links aren't suitable for resource reservations (i.e. neither stable nor reliable). It is recommended that more robust methods be investigated for resource reservations over such links.

The RRS currently only considers non-LOS links as suitable for reservations. A potential enhancement would be to accept routes that include LOS links. Instead of reserving resources end-to-end, bandwidth would be set aside only on the portions of the route that are non-LOS and diffserv used in the unreliable LOS portion. This would imply some mapping between intserv and diffserv at the LOS edges with reserved flows likely using the real-time class or highest Assured Forwarding class in the LOS portion. This enhancement is left as future work.

There are other parameters that must be considered at reservation time before a reservation can be admitted. One of these parameters is the priority of the request. If a request is of a higher priority it may be acceptable to drop or degrade existing lower priority reservations if that would provide sufficient resources for the new flow. In the proposed system, reserved flows that are pre-empted will not be blocked but revert to

[1] The policy-defined pool multiplier is currently set to 50%.

diffserv treatment. This diffserv treatment is determined by policy [1] and is automatically imposed when a reservation fails for either reason.

Another issue in the maritime environment is how to deal with node mobility and the related change in networking environment. How do reservations persist or change when link status (utilization and connectivity) along the route changes with time? The approach adopted here is that reservations should be updated “periodically” and monitored “constantly.” When a reservation is updated the future stability of the path is considered. When a link on a reserved path fails, the reservation will both be degraded and take the default route until (depending on policy) a new reservation can be made, or the reservation is dropped and the user informed of the error. This mechanism is described further in Section 7.3.

3. The Main Admission Control Algorithm

Resource reservations requests consist of a source S, destination D, resource requirements Q and policy requirement P. The admission control algorithm that has been developed to support the RRS uses this quadruple to determine if the network has sufficient resources to meet the resource requirements. If the flow can be routed from S to D along route R while meeting all requirements Q and P at each intermediate link, the algorithm will admit the flow and make the appropriate resource reservations along R. If no route R is found, the flow is rejected. Typical desirable attributes of such an algorithm include efficient signalling, load balancing, secure access, and in the case of a maritime network, fault tolerance.

The mechanisms developed to provide admission control are based on a reactive as opposed to pro-active model that takes the view that while routing and QoS signalling should be separate, they can work together. Pro-active models that maintain reservations, ready for use in advance, would have a high overhead and are not suited for this environment.

The main admission control algorithm is divided into four phases. Phase one of the algorithm is described in Chapter 4. In order to support policy based routing, the topology of the network is discovered using information already available on the access router. The OSPF routing protocol provides information on what links are currently available in the network which can be used to determine connectivity and link type as outlined in Section 4.1. This topology information is used to generate potential reservation routes dynamically. The routing algorithms will ignore links that violate trunk utilization policies and/or do not offer sufficient bandwidth. The algorithm then generates a number of routes for load balancing and greater chance of call acceptance. The route generation algorithms that were developed are described in Section 4.2.

A proprietary robust signalling protocol designed for the maritime environment has been developed. Admission Control decisions are performed at each hop along the selected route(s). Resource request handling is done locally at each node in the route by the local resource reservation service, a service within the policy system co-located with the WAN router of the node. No admission or policy information is maintained at the router level. This second phase of the protocol is outlined in Chapter 5.

A novel capability of this algorithm is that a reservation in the reverse direction (destination to source) can be made at the same time as the forward direction (source to destination). Bidirectional reservations can be especially useful when the application has significant traffic in the reverse direction that needs protecting at the same time, such as VOIP or ftp downloads. Making reservations in both directions at once reduces overhead and latency while ensuring that the reservation is symmetric (it reserves at the same nodes for use in both directions). Section 8.4 describes this capability further.

The third phase of the protocol is provided in Chapter 6. Once one or more reservation probes have reached the destination, the selection of the best route is done according to the following criteria: number of flows that would be preempted by taking the route, priority of flows preempted on the route, minimum bandwidth available on the route, and the number of hops on the route. The algorithm for determining which route to choose is further explained in Section 6.1. When the destination has decided upon a route, a confirmation message is sent back along that route with each RRS updating the configuration of the router so that the flow is treated in the reserved class. In order to determine if a new request should be committed, the system keeps track of which reservations are using which local links. Each link has a pool of bandwidth available for reservation. Currently up to 50% of a link bandwidth may be reserved, but this value is configurable by policy. It is important to mention, here, that this bandwidth is available for other traffic if it is not being used by the reserved flows. This mechanism for resource allocation and enforcement is described in greater detail in Section 6.2.

When a confirmation message reaches the source, the resource reservation enters its active maintenance phase (phase four). A reservation maintenance algorithm is used to keep the resource allocation active until the request times out or is cancelled. This mechanism is described in Section 7.1. The protocol supports priority, preemption, and fault-management. Priority based preemption is supported by allowing users to assign priority to service request. Lower priority calls will be pre-empted only if insufficient reserve-able bandwidth is available. Fault management is achieved by reacting to topology changes (e.g. link failures) by pre-empting reservations that use the failed link. All nodes along the route are then notified. The preemption and fault management capabilities of this service are described in Section 7.2 and 7.3 respectively.

The main algorithm is presented in flow diagram format in Figure 2 and in pseudo code in Figure 3 below.

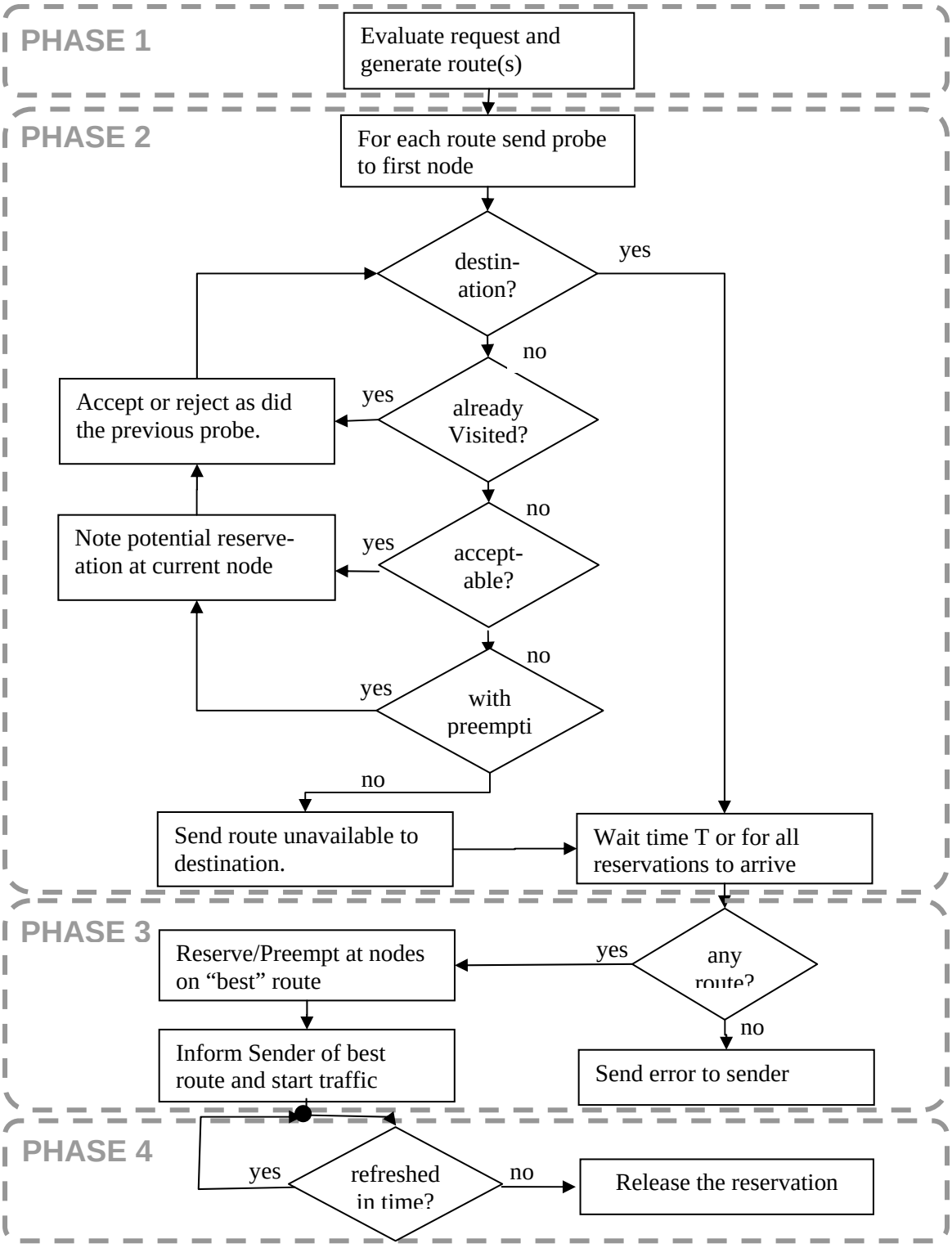


Figure 3. *Flow Diagram of Main Admission Control Algorithm*

Algorithm: AdmissionControl

Input: a graph G with link descriptors Li, source S, destination D, routing policies (including priority) P, and resource requirements RR (may be bidirectional).

Output: reserved route R (and alternate routes R')

1. from RR and P determine priority number of resource routes K to establish
2. using one of the Path Generation algorithms (Section 4.2 generate the K best policy acceptable routes (routeset)
3. probe the K routes from S to D in parallel determining if RR can be satisfied on the links at that particular time. At each node N in route R element of routeset
4. if another probe by this reservation has made a partial reservation at this node on the same link proceed to the next node in route R
5. if the new flow can be accepted at N[2] make a partial reservation, note residual bandwidth[3] in probe, and proceed to the next node in route R
6. if the new flow can be accepted at N but only by pre-empting existing lower priority reservation(s) make partial reservation, but record details of pre-emptable flows in probe, and proceed to the next node in route R
7. if the flow cannot be accepted, the probe should be marked unsuccessful and sent directly to D.
8. once K probes have arrived at D[4] or timeout has occurred at D:
9. if at least one successful probe has reached the destination, use the Route Selection algorithm (Section 6.1) to decide which reverse path should be confirmed by a probe sent node by node in the reverse direction to S.
10. if only unsuccessful probes, send error to sender.
11. once S receives notification from D, return reserved route R. Optionally, alternate successful routes R' can be returned if noted at D.
12. send maintenance messages along reserved route R at policy-defined interval so that each node that does not receive the probe in three times the interval will release the associated completed reservation.

Figure 4. *AdmissionControl Algorithm Pseudo-Code*

Note that the AdmissionControl algorithm uses a two-phase commit mechanism where a route is first probed (phase 2) and then confirmed (phase 3). Another possibility especially useful in unidirectional networks would be to assume reservations will be successful and make the required changes to the router when the network is first probed. In this "single-phase commit" scheme reservations must have short time outs and frequent refresh/maintenance probes to ensure good utilization. The drawback is

[2] There is also a case here where there may be a partial reservation by a lower priority flow that must be pre-empted. This could be included in this case, but more though should be put into the order in which it would be counted (for instance compared to an actual reservation on even lower priority…)

[3] Residual bandwidth is recorded in % bandwidth available and will overwrite a higher remaining bandwidth in the probe.

[4] The number of probes sent for a reservation (K) is included in each probe so that the receiver will realise when all probes have arrived.

the increased overhead and router configuration activity when a reservation is not successful. This mechanism has not been explored.

It should be noted that timeouts play a critical role in this system. In phase 3 AdmissionControl does not send negative acknowledgements to the rejected routes where partial reservations have been made. This implies that partial reservations will remain for a period when not needed. A relatively short refresh “local” timeout is used for partial reservations to ensure that partial reservations which are not confirmed within a period of time are released (see Chapter 5). Timeouts are also used to release reserved resources that are not refreshed by a maintenance probe within a certain period of time (see Section 7.1). The choice of an appropriate timeout value (how long to wait for an event) is an area for further investigation in this system and is discussed in more detail in Section 8.2.

Finally, an enhancement has been proposed to the timing of the algorithm. The concept of future time has been added so that reservations may be made in advance. Though it is arguable that this would be valuable in a maritime environment, the capability is detailed in Section 8.5.

4. Route Generation (Phase 1)

To support policy based routing, the topology of the network is discovered using information already available on the access router. The OSPF routing protocol provides information on what links are currently available in the network. This can be used to determine connectivity and link type. The network topology is generated from this information as described in Section 4.1.

Using this database, a route may be found from source to destination using standard routing algorithms such as the Dijkstra algorithm [2]. Two improvements are proposed here. First, before a route is generated all links that are not policy acceptable or do not have sufficient raw bandwidth for the request are marked to be ignored. Thus links that cannot handle the reservation will not be probed. Second, multiple routes are generated and probed in parallel. Since the route with the most residual bandwidth is often chosen from amongst the multiple routes probed the reservation load will be balanced amongst the links in the network. The full route selection algorithm is presented in Section 6.1. Three algorithms are proposed in Section 4.2.

Note that reservations in the proposed scheme may be unidirectional (reserved only from source to destination) or bidirectional (reserved from the source to destination and destination to the source at the same time). Since most applications usually send traffic in both an upstream and downstream directions it makes sense for reservations be made in both directions at the same time. This avoids the problem of different routing of traffic in opposite directions or that reservations can be made in only one direction due to limited resources. Bidirectional reservations may be explicitly requested or handled transparently by the reservation system based on the traffic type. See Section 8.4 for more details on how bidirectional reservations are made and Annex B for an example.

4.1 Topology Discovery

The network resource parameter that is currently reserved for service requests is the residual bandwidth. When the link is dedicated and provides stable bandwidth (either on or off at a certain rate) the residual bandwidth can be calculated by looking at the nominal bandwidth of the link and subtracting the amount currently reserved. This is applicable in the mobile maritime environment for most satellite links and BLOS HF links where the media is not shared and can be characterised as either available at full capacity or not available (binary). The reservation protocol assumes nodes are aware of the residual bandwidth on all outgoing links.

LOS (UHF/VHF) links on the other hand are a shared media and as such residual bandwidth cannot be reliably determined. Also, there are currently no standards for QoS support in the MAC of these links. Attempts to introduce QoS in these environments include probing (to determine available bandwidth), and cross-layer communication (assuming the MAC layer is instrumented to report QoS information

using a method such as a proprietary SNMP MIB). These methods have achieved little success. For these reasons, LOS is considered unsuitable for reservations and LOS links are currently not included for route generation. Although LOS links are ignored when hard QoS is considered, they may be used when soft QoS (preferential treatment) is enforced.

Topology information is extracted from the domain routing protocol OSPF. OSPF regularly sends Link State Advertisements (LSAs) to distribute knowledge of the domain's connectivity information. Each router stores a complete set of the most recent LSAs in a Link State Database (LSDB). This list is dynamic. As links fail or become active new LSAs are distributed. From the standard OSPF LSDB, the topology discovery module can obtain the following info:

- all links in the domain with their associated cost metric.
- node connectivity (which links go with which nodes)

By setting the OSPF link costs according to the type of link, the characteristics of the links can be determined directly as shown in Table 1. This includes the nominal bandwidth of the link, and some of the OSPF configuration parameters including the dead time, hello time and retransmit time. This method has been used before in the military context [3].

Table 1. *Equating OSPF Cost to Link Type and Bandwidth Capacity*

OSPF COST	LINK TYPE	BANDWIDTH (KBPS)	DEAD TIME (S)	HELLO (S)	RETX (S)
800	SHF SATCOM	128	40	10	5
750	INMARSAT	64	40	10	5
1150	UHF LOS	64	40	10	5
1300	UHF SATCOM	32	120	30	10
1900	HF BLOS	9,6	120	30	10

Using all of the information provided above, the following data is maintained;

- a list of all currently connected nodes and links in the network (populated from the OSPF LSDB of the local router)
- the baseline bandwidth of each link (populated from the OSPF cost-metric vs. bandwidth chart)
- the current pool size (a percentage) for the "reservable" class (defined by policy)

- the current amount of bandwidth reserved per local link (populated by the RRS as flows are admitted, terminated, pre-empted, etc.)

This information is used for route generation and resource allocation (see below).

4.2 Routing Algorithms

Three different route generation algorithms are proposed here as part of the admission control service. The three algorithms can be summarised as;

1. Use the best route or none at all.

2. Repeatedly remove the best route from the graph and next try the best route from the remaining graph (completely disjoint)

3. Iteratively remove one or more of the "poorest" links of the best route from the graph and next try the best route from the remaining graph (partially disjoint)

The main advantage of probing multiple paths is to discover the "best" current reserved path available. Hopefully several paths will be discovered and the receiver will have a choice of selecting the path such that the reservation can be made with minimal impact on the existing flows. Another advantage is that probing multiple paths promotes load balancing. Where default routing forces all traffic over the "best" link, multiple routes are considered hopefully identifying the least loaded links to be reserved. This allows the load to be balanced both at individual nodes and throughout the network.

One of the potential advantages of probing (partially) disjoint routes is that alternate acceptable routes could be maintained for later use. If one or more of the usually un-reserved alternate acceptable paths were also reserved and maintained from the source through the use of maintenance messages it would be possible to immediately redirect reservations that have been impacted by preemption or link failures. Alternately only the acceptable routes could be communicated with the source which upon failure could attempt to reserve on another route before redirecting traffic. Such mechanisms are left for future investigations.

Note that in the proposed admission control algorithm alternate routes are tried in parallel and thus no feedback on overloaded links found in the process is given. A possible alternative, not investigated here, is to try the routes in series removing offending links one at a time as they appear in the attempt to make the reservation. This method was rejected as adding greatly to the overhead and delay in reservation setup. On the other hand it would eventually find an acceptable path (if one exists). Due to mobility no optimal paths exist forever and this is the reason this method is not recommended.

4.2.1 Best-Path Routing Algorithm

The routing algorithms take three main data points as input. They take the network graph and link descriptions from the routing protocol (likely to be OSPF), the destination node from the reservation, and the routing policies from the policy system. The Best-Path algorithm (Figure 5) returns a single one policy acceptable route, which is likely to be all that is needed for most reservations. This algorithm however does not allow for load balancing and does not optimise the route path.

Algorithm: Best-Path

Input: a graph G with link descriptors Li, source S, destination D, and flow descriptor F with routing policies P

Output: a path from S to D that traverses only policy acceptable links in the least hops with greatest available bandwidth if such a path exists

1. remove from graph G all links Li that do not meet the resource requirements of F or policy requirements P
2. attempt to find a route R from S to D in G with least number of hops and using links with the highest bandwidth when there is a choice
3. if no such path exists, return an error otherwise return R.

Figure 5. *Best-Path Algorithm Pseudo Code*

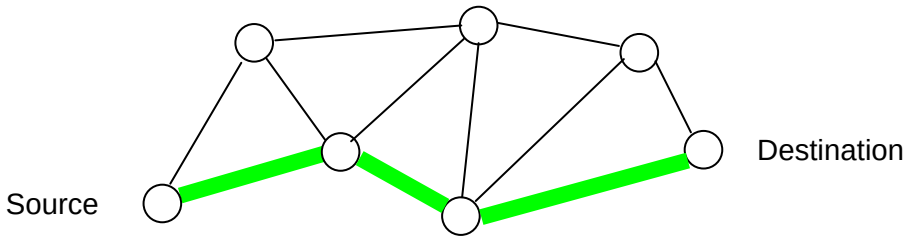


Figure 6. *Best-Path Example*

For example, in Figure 6 there is a single path chosen with the least number of hops for source to destination. This path is the result returned.

4.2.2 Multiple-Disjoint-Path Routing Algorithm

The Multiple-Disjoint-Path algorithm and the Multiple-Partially-Disjoint-Path algorithm provide load balancing by probing multiple routes at once and choosing the optimal path (most residual bandwidth). The main difference between the two is that the former provides paths that are completely disjoint in that no links in one path are present in any other path generated by the algorithm. The pseudo code is given in Figure 7.

Algorithm: Multiple-Disjoint-Path

Input: a graph G with link descriptors Li, source S, destination D, flow descriptor F with routing policies P, and desired number of disjoint paths K.

Output: up to k policy acceptable disjoint paths from S to D.

1. path_exists:=true, routeset={}
2. remove from graph G all links Li that do not meet policy requirements P
3. while (path_exists and K > 0) {
4. attempt to find a route R from S to D in G with least number of hops and using links with the highest bandwidth when there is a choice
5. if no such path exists, return a routeset
6. otherwise add R to routeset, remove all links in R from G, and K=K-1.

Figure 7. *Multiple-Disjoint-Path Algorithm Pseudo Code*

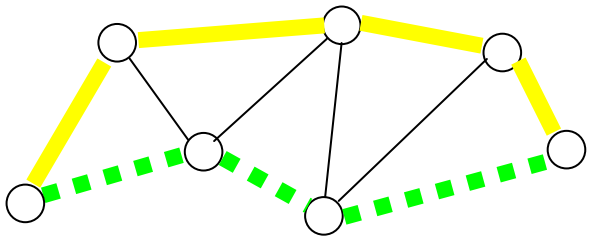


Figure 8. *Multiple-Disjoint-Path Example*

As shown in this example, two paths may be found. When links from the first path are removed as indicated by the green dashed line, an alternate route is still available as shown by the solid yellow line. These two paths would be returned by the algorithm.

4.2.3 Multiple-Partially-Disjoint-Path Routing Algorithm

The Multiple-Partially-Disjoint-Path algorithm is disjoint only in single links that are likely to fail or are likely to be congested (as decided in advance by link type). The partially disjoint algorithm will generate a number of routes in series. The first route generated is the least cost route from source to destination while ignoring links that are not policy acceptable for the requested application. The second route uses the same algorithm, but also ignores the highest cost link of the best route previously generated. The third route also uses the same algorithm, but ignores the highest cost link of the two previous routes. This can continue until no more routes are possible. The algorithm pseudo code is presented in Figure 9. Examples of this algorithm are given in Figure 10, and Figure 11.

Algorithm: Multiple-Partially-Disjoint-Path

Input: a graph G with link descriptors Li, source S, destination D, flow descriptor F with routing policies P, and desired number of different paths K with at least L different links.

Output: up to k policy acceptable partially disjoint paths from S to D.

1. path_exists:=true, routeset={}
2. remove from graph G all links Li that do not meet policy requirements P
3. while (path_exists and K > 0) {
4. attempt to find a route R from S to D in G with least number of hops and using links with the highest bandwidth when there is a choice
5. if no such path exists, return routeset
   otherwise add R to routeset, remove the L links with lowest bandwidth in R from G, and K=K-1.
   }

Figure 9. *Multiple-Partially-Disjoint-Path Algorithm Pseudo Code*

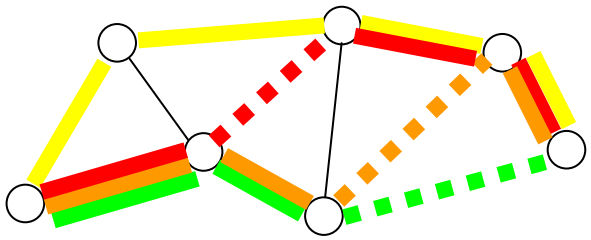


Figure 10. *Multiple-Partially-Disjoint-Path Example with L=1*

In the example above four different paths could be found. When the first path (green) has the dashed link removed, the orange path may be found and so on. Thus up to four paths may be returned by this instantiation of the algorithm where single links are removed. Note that the exact link removed will depend on some metric not shown in the diagram and many different possible combinations of paths are thus possible.

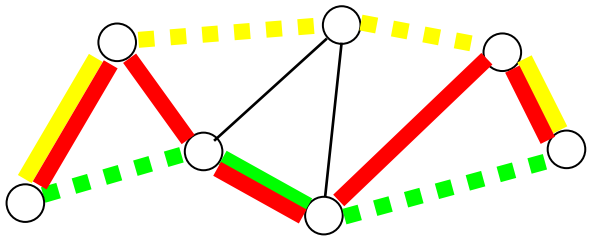


Figure 11. *Multiple-Partially-Disjoint-Path Example with L=2*

In Figure 11 two links are removed from each previously discovered path. The graph has been displayed in the case where three routes are discovered, though again the exact links removed will depend on information not shown here. Note that for both of the disjoint algorithms a poor choice of the first or subsequent path and/or which links are removed may mean that less disjoint routes are discovered that could potentially exist. More complex algorithms for discovering such routes exist but these simple algorithms are suggested for initial investigations. More complex algorithms may be investigated as future work.

5. Admission Control Probing (Phase 2)

The admission control protocol in this work is similar to RSVP [4] but modified for the maritime environment. RSVP was found to be unsuitable for three reasons. First, RSVP assumes unidirectional reservations where in the maritime environment most reservations are bidirectional. Second, RSVP uses the default routing to attempt reservations and does not probe multiple routes in parallel. In the low bandwidth maritime environment the default route would be quickly overloaded and attempting alternate routes will increase the call acceptance rate. Finally, although the RSVP standard has provisions for carrying policy control information, most implementations do not support this capability[5]. This is required for communication with the RRS at each hop in the reservation to determine whether the flow should be admitted or not (depending on both local policy and the policy carried by the resource request).

Instead a proprietary robust signalling protocol designed for the maritime environment has been developed. Admission Control decisions are performed at each hop along the selected route(s). Probes travel on their assigned route from the source to destination one node at a time. The network resources along a path are evaluated at each node along the route. Currently the only resource that is considered is bandwidth. Each link has a pool of bandwidth available for reservation. A policy configurable percentage of the total bandwidth (currently set at 50%) of a link may be reserved. If sufficient resources exist to meet the reservation request stored in the probe for the desired link at the current node, the residual bandwidth of the link is noted in the probe and forwarded to the next node.

If insufficient resources are found at a node, the current reservations are examined to see if preemption of lower priority flows would free enough resources to meet the needs of the current request. If sufficient resources are still not available, a failure message is sent to the destination (the destination makes the final decision of which route to reserve). If sufficient resources would be made available, the request is forwarded to the next node in the route including information on the flows that would be preempted if this route were used. The preemption algorithm is given in Section 6.2.

It is important to note that no change is made to active reservations or to the router during route probing. The purpose of route probing is simply to determine if a reservation is possible along any of the generated routes. A copy of the resource request is stored at each node in the hope that a confirmation will eventually arrive. At that point low level (configuration) policy will be generated. If the confirmation has not arrived in a preset amount of time the “pending resource request” record is purged.

[5] It is the case for the RSVP implementation in Cisco routers.

Also note that if multiple probes for the same reservation for the same link arrive at a node the same accept/reject actions are taken as before. This may happen when the multiple partially disjoint routing algorithm of Section 4.2.3 is used.

Finally, this method assumes nodes are contacted in series. This has the advantage of limiting protocol overhead to approximately (P+1)*R inter-node messages per route where P is the number of probes and R is the average number of intermediate nodes between the source and the destination where reservations must be made. Since no nodes are contacted beyond a choke point where a link did not have the required bandwidth, this method may have an even lower overhead. Another advantage is that each hop can ensure that the next router is connected as expected by the route in the service request. If it is not connected or the resources required are not available the protocol will abort and send a deny message to the destination indicating failure of this route. A probe taking some other route will hopefully succeed. Alternate methods where nodes are contacted in parallel with $O(PR^2)$ messages have not been investigated

6. Route Selection and Enforcement (Phase 3)

The final decision of which of the successful probe's routes to commit is up to the destination. The route selection algorithm waits until all reservation probes from the sender have been received or a time out has occurred indicating the remaining probes should be considered lost. The selection of the best route is done according to the following criteria: number of flows interrupted (lower total number preferred), priority of flows interrupted (lower maximum priority preferred), residual bandwidth available (minimum of remaining bandwidth after reservation would be made at each hop with a higher total value preferred for data-base applications), and number of hops in the route (lower value preferred for real-time applications). The detailed algorithm for determining which route to choose is further explained in Section 6.1.

When the destination has decided upon a route, a confirmation message is sent back along that route with each RRS updating the configuration of the router so that the flow is treated in the reserved class. In order to determine if a new request should be committed, the system again examines the other reservations that are using the local links. If enough unallocated resources remain in the pool, the resource allocation is enforced on the router. If insufficient resources are available, preemption is attempted as in phase two and if successful the reservation is enforced. Otherwise a failure message is sent to the source and no changes are made at the router.

The preemption algorithm is thus considered at two different points during admission control. First, when a service request first reaches a node, and second when a reservation confirmation arrives. Only in the case of the reservation confirmation are reservations actually pre-empted. For requests, the list of reservations that would currently have to be pre-empted to admit the flow are added to the probe to aid the destination in deciding on the best route to be confirmed. For confirmations, if the link does not have sufficient "reservable" resources to meet the request the only alternative to rejecting the reservation is to pre-empt lower priority flows. This is only done if such preemptions would free sufficient resources, otherwise the request is rejected. The preemption algorithm used in both cases is presented in Section 6.2.

6.1 Route Selection Algorithm

If no successful probes reach the destination, a reservation failure message is sent to the source so the user can be notified. If only one route is successful, that route is used by the reservation and must be confirmed in the resource allocation and enforcement phase. When multiple successful probes reach the destination, a single route must be chosen. Several factors may influence the choice of route. The following factors are taken into account: the number of flows that would be pre-empted along with their priority, the minimal residual bandwidth, and the distance (hop count). The route selection algorithm that has been developed is as shown in Figure 12:

Algorithm: Route-Select

Input: the acceptable routes, minimum residual bandwidth of each route and pre-empted reservations associated with each routing (including bandwidth and priority of each reservation)

Output: the selected route

1. Pick the route with no preemptions, If more than one select the flow with the
2. Least number of pre-empted flows. If equal then select the flow with the
3. Lowest maximum priority flow pre-empted. If there is a tie, look at the application type;
4. For UDP-based applications, use least number of hops. If there is a tie, use the greatest amount of residual bandwidth;
5. For TCP-based applications, use the greatest amount of residual bandwidth. If there is a tie, use the least number of hops;
6. Use the Probe that was first to arrive if a tie still exists

Figure 12. *Route-Select Algorithm Pseudo Code*

The rational behind this algorithm is that real-time applications are more delay sensitive than they are bandwidth sensitive. Placing them on routes that are close to saturation may be advisable if the delay is reduced. Since delay is most often a factor of hop-count, the lower hop count is favoured for these applications. On the other hand, applications with heavy bandwidth requirements are usually TCP-based. Since such applications are more concerned with total data transmitted, a longer delay may be tolerated and thus routes with greater residual bandwidth are preferred.

6.2 Preemption Algorithm

The preemption scheme that was implemented is based on strict priority where lower priority flows may always be interrupted by higher priority flows. A list of flows that would have to be pre-empted to make room for the new reservation (if possible) is provided as output. For a resource request, it confirms whether a reservation would be able to reserve sufficient bandwidth were the reservation to be committed immediately. No committed reservations are affected at this point, the sole purpose being to fail a service request if insufficient resources can be found. On the other hand when the preemption algorithm is used for a resource commit the service will tear down the less important reservations to free up the bandwidth and commit the new reservation. The algorithm pseudo code is presented in Figure 13.

Algorithm: Preempt

Input: the current residual bandwidth of the link, the bandwidth required for the new reservation and the list of lower priority reservations already using that link.

Output: a list of flows to be pre-empted (empty if fail)

1. The largest (in terms of reserved bandwidth), lowest priority flow is pre-empted first.
2. If insufficient bandwidth has been released, the next largest flow at the same priority is pre-empted.
3. If the current priority level has no more reservations, the next lowest priority level is similarly emptied one reservation at a time from largest to smallest
4. Once sufficient bandwidth has been made available (residual + pre-empted bandwidth >= requested bandwidth), no more preemptions are made.
5. If insufficient bandwidth has been released with all lower priority flows pre-empted, return an empty set to signal preemption failure.

Figure 13. *Preemption Algorithm Pseudo Code*

A possible improvement to this algorithm would be at line 4. When it is found that the current reservation being considered at line 1 or 3 for preemption would free more than enough bandwidth for the new reservation, look for the smallest reservation at the same priority level that would satisfy the bandwidth requirement and pre-empt it instead. This improvement would slightly lessen the prejudice of the preemption algorithm against large reservations while maintaining the same total number of flows pre-empted. An idealised backpack style optimisation of this part of the algorithm is left as future work.

Second, a possible improvement to the main admission control algorithm would be to attempt to use potential preemptions during phase two to proactively reroute low priority reservations. Again looking at the characteristics of the environment, there are likely to be critical links that are likely to become overburdened. If a high priority reservation is exploring such a link and must pre-empt the lowest flow, the low priority flow could make an immediate “route update” attempt that ignores its existing reservation on the contested link. With the higher priority reservation blocking that link it will (hopefully) find an alternative routing before the high priority reservation forces it to remove itself. The route update mechanism is presented in Section 7.4.

7. Reservation Maintenance (Phase 4)

When a confirmation message reaches the source, the resource reservation enters its active maintenance phase. A reservation maintenance algorithm is used to keep the resource allocation active until the request times out or is cancelled. Requests can be terminated by a number of events including termination by the user, end of the reserved period, preemption by a higher priority flow, or failure of a link on the reserved route.

During the maintenance phase, keep-alive messages are sent along the reserved path at a policy-defined interval. Each RRS on the path must receive one of these messages within another policy-defined interval or the reservation is considered to have terminated. This "natural" termination causes the RRS to simply remove the reservation from its list freeing the associated bandwidth and reversing any router configuration that has been made. The reservation maintenance algorithm is given in Section 7.1.

It may happen that existing reservations are interrupted and do not terminate naturally from user intervention or expiry of the request. This may be due to preemption by the confirmation of a higher priority request (as outlined in Section 7.2) or by failure of a link on the reserved path (as outlined in Section 7.3). A proposed method for reducing the frequency of such interruptions by rerouting ongoing flows is given in Section 7.4.

7.1 Reservation Maintenance Algorithm

The Maintenance algorithm maintains the resource reservations over time. It is geared to refresh the confirmed reservations until they reach a set expiry time. At a policy-defined time interval, refresh messages are sent by the source node of an active reservation along the route selected by the AdmissionControl algorithm. Each node in the path resets the maintenance timeout counter when it receives an appropriate refresh message. If the maintenance timer were to expire (it is set to a policy-defined multiple N of the maintenance refresh interval in order to allow for temporary failure of the link) the reservation is released. The algorithm is given in Figure 14.

Algorithm: **Maintenance**

Input: list of reservations, and reservation id from the maintenance probe.

Output: a maintenance message, a release message, or nothing.

1. locate the local reference for the reservation identified in the maintenance probe
2. If the reservation is found, restart its associate maintenance timer and forward the probe to the next node on the reservation's route, or if this is the hop before the destination it can be discarded.

3.	If the reservation is not found send a release message is sent to the source identified in the maintenance message. All nodes on the path will release the associated reservation. If the reservation is bidirectional, the release message is also sent down the path to the destination.

Figure 14. *Maintenance Algorithm Pseudo Code*

If a node no longer has an associated reservation, a release message is sent along the path to the source (and destination in the case of bidirectional reservations). All nodes along the path will release the associated reservation. This may occur when a reservation was previously released due to preemption by a higher priority reservation or a link failure. When the reservation is released and the message does not reach the source this will leave many nodes with the incorrect assumption that the reservation is still active. Reservation maintenance thus provides an additional mechanism to clean up orphaned reservations where the resources are no longer available end-to-end.

In a maritime network the maintenance of existing reservations could be seen as an inherently impossible task. Since link outages and bandwidth fluctuations are to be expected, guarantees cannot be rigidly interpreted and a more statistical model must be used. The approach taken here is twofold. First to periodically update reservations (which will otherwise release the resources they are holding for a particular flow) and second to actively repair routes which may have been invalidated by a change in routing (currently unimplemented). In the simplest case a probe is sent from source to destination along the agreed upon path refreshing timers at each node. An alternate reservation maintenance and route update mechanism (proposed in Section 7.4) is similar to the basic admission control mechanism but repeats phase two of the main algorithm on a regular basis to determine if a better route exists.

With slow mobility the period between maintenance updates can be relatively long (tens of seconds or minutes) but an appropriate value should be determined empirically and set by policy. It is likely that the tempo of the operation will change the rate at which links will fail and reform and thus influence the appropriate maintenance period.

In a more general sense appropriate values for all timeouts in the system will need to be investigated. This would include the time to hold a partial reservation, the time to hold a locally confirmed reservation, the time at a receiver to wait for other routes that are being probed in parallel, and the time for the sender to wait before giving up on a reservation response from the networks before re-trying or giving up. There are also timers for the acknowledgement for the reliable messages.

The timeout values will also be critical in establishing viable load balancing. It is expected that the simulation (the preliminary results of which are presented in Chapter 9) will provide appropriate values. The current values being used are given in Annex A.

An alternative to the use of keep-alive messages for determining when a reservation has terminated is the use of explicit signalling. In this case when a reservation completes a message is sent from the source down the reserved route releasing the reservation at each node until it reaches the destination. This method would reduce the number of control messages but increase the chance of orphaned reservations (that have terminated but block resources at some node(s) on the route.) This method has not been investigated. Note that both mechanisms could be used. That would allow the refresh timeout to be set for a relatively long period of time.

7.2 On Preemption

When a reservation has been pre-empted during the commit phase of the main algorithm (phase three), a release message is sent to the source of the reservation. The source will release the reservation by refraining from sending further keep alive messages. All remaining nodes on the pre-empted reservation's path will eventually release their associated bandwidth when they do not receive a keep-alive message.

The current implementation of this final phase has release messages sent in series to both source and destination following the reserved route releasing the reservation as it is sent. This takes more bandwidth but increases call acceptance for the period of time it takes for reservations to time out since the reserved bandwidth would be made available earlier. Both of these alternatives to preemption termination are being investigated.

Currently when a flow is pre-empted, notifications are sent to all nodes on the route. This is accomplished by sending a release message in each direction: one towards the sender and one towards the destination. All nodes which receive the message release the corresponding reservation.

This method is robust to failure since even if the release message is lost, a refresh message from the sender will find no matching reservation on the local node which will send a release message back to the sender. This will cause no more refresh messages to be sent to the other nodes on the route and they will eventually time out and release the reservation.

The treatment of pre-empted flows is similar to the case of link failures. The difference is that in the case of a link failure, the event is given time to possibly fix itself whereas in the case of preemption, the event must be treated immediately (irreversible).

If not treated appropriately, preemptions may lead to inefficient resource utilization. Therefore another signalling message, the "Reservation Release" message has been added in the protocol.

Depending on the situation, the issuing of the message will be handled as follows:

- The reservation refresh messages need to be frequent enough (typically every 5-10s). The nodes would timeout after X*Refresh period.

- If a node times out after missing X number of refresh messages, the node simply releases the resources locally (deletes the reservation). If later, the node receives a Refresh message for a non-existing reservation, the node issues a "Reservation Release" (or Tear Down) message up the path (i.e. each node relays towards the source) and if the reservation was bidirectional, it also issues it down the path (i.e. each node relays towards the destination).

- If a (committed or partially committed) reservation is pre-empted (because of the commit of another reservation), the node issues a Reservation Release message down the path as well as up the path (each node relays the message in the appropriate direction until it reaches the ends, both source and destination).

- If a node receives a Commit message and its' processing fails i.e. the associated reservation does not exists anymore (pre-empted or timed out) or it cannot be admitted anymore (resources have changed while waiting for the commit to come back), the node drops the commit and issues a Reservation Release message down the path as well as up the path (each node relay the message in the appropriate direction until it reaches the ends, both source and destination).

The protocol has been made semi-reliable. Hop-by-hop acknowledgments have been added to the following messages: the fail message, the service request, the service commit and the release messages. This is required since links are wireless and error-prone. Acknowledgements are discussed in more detail in Section 8.3.

7.3 On Link Failure

A failure may occur to a link with active reservations. When a fault/restoration of a link modifies available resources, the topology monitor/event service (parts of the policy system) will notify the RRS of the changes. When a link is flagged as no longer in the topology, the RRS will recalculate whether admitted flows can still receive the QoS asked for. All reservations that are currently using a failed link will be released after a preset amount of time as if they had been pre-empted (see above). At this time no automatic healing of failed reservations is attempted.

Detection of link failure is done via the topology monitor. The topology monitor polls the router for the OSPF LSA information every "x" seconds (see Annex A for current protocol parameters.) As outlined in [1], the policy system at each node updates the network graph out of the retrieved information.

Although all nodes in the network are eventually notified of link failures via the flooding of the OSPF LSAs, it was decided that only nodes that are adjacent to the failure take action. When a local link failure is detected, the node starts a timer to give a chance to the link to come back (avoid flapping links). If the link comes back before timer expiry, the node takes no further action. However, upon timer expiry, the node verifies which local reservations are affected by the failure. It removes the affected reservations as well as the associated router configuration (if applicable i.e. if the reservation had started) and issues a "Reservation Release" signalling message up the

path (i.e. each node relays towards the source) and if the reservation was bidirectional, it also issues it down the path (i.e. each node relays towards the destination).

The obvious alternative is that when a link failure is detected by a source node that has a reservation through the failed link it acts just as outlined above for nodes on either side of the link. It will wait for x seconds and then immediately remove its local reservation. All other nodes on the path can do the same since they carry the full route of each reservation.

There is another alternative to dealing with link failure that has not been investigated and that is local healing. In this case the nodes at either end of the link failure attempt to route around the failure by creating a new reservation between them. However, local healing doesn't make much sense in the maritime environment where there are a limited number of bearers at each node (sparse connectivity) and the bandwidth for signalling is at a premium.

7.4 Updating Routes (Unimplemented)

Another case when reservations might change is the case that new links become available. In this case a better route may become available and should be taken advantage of. This change would be noticed by the sender and an OSPF change and if it determined that a better route exists it should test the route with an AdmissionControl probe with the same reservation number as its exiting reservation. If the probe comes back successful the flow will be transparently switched to the new route, otherwise the old route will be maintained. The routing update functionality is not currently implemented.

The maintenance algorithm could be expanded to continuously monitor and improve the routing of existing flows. Since links come and go in maritime networks it is desirable to reroute reservations to make use of improved routes or avoid one that are nearing failure (or saturation). Currently when a link fails the reservations that use that link are simply released and associated data traffic continues on the policy-defined diffserv service as explained in Section 7.3.

It is suggested that routes be updated to avoid such failures by proactively rerouting when a reserved route is near failure, either by becoming saturated (no residual bandwidth) or as detected by means outside the scope of this report.

Algorithm: Route-Update

Input: a graph G with link descriptors Li, source S, destination D, routing policies P, and resource requirements RR.

Output: Boolean

1. At some regular interval related to the priority of the reservation send probes as in AdmissionControl phase 1 with the current parameters to see if a different route is available but do not confirm reservations on the return path.
2. If the destination sees a better route that a) does not pre-empt another reservation and b) has the necessary resources available then
3. Switch to that route with a probability X (initially X will be high for lower priority routes and low for higher priority routes.)
4. Otherwise refresh old route.

Figure 15. *Route-Update Algorithm Pseudo Code*

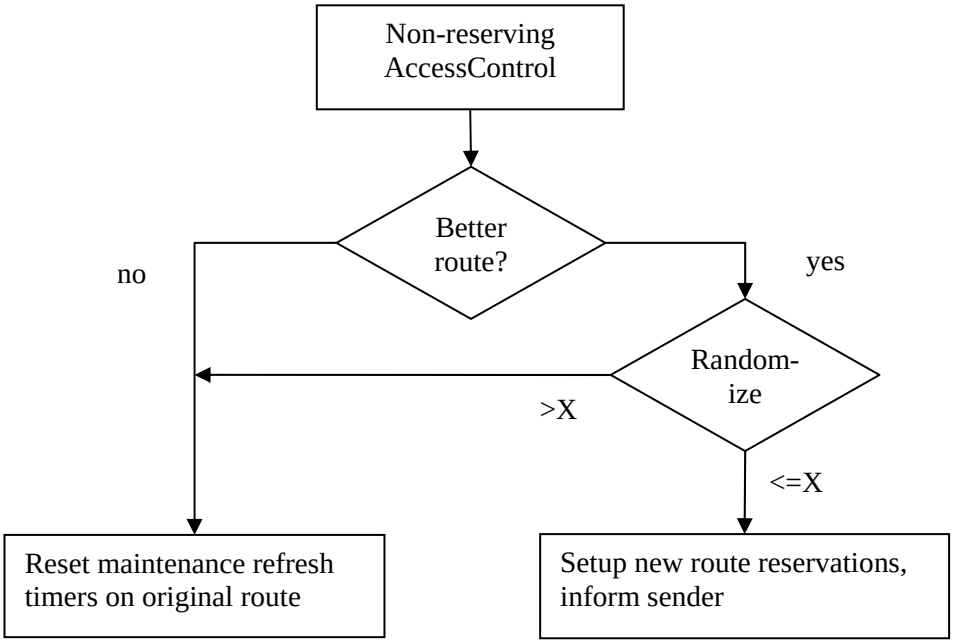


Figure 16. *Flow Diagram of Route-Update Algorithm*

In this algorithm reservations are only switched with a probability of X in order to reduce route flapping. This value can be set to zero if no route improvement is desired, or can be set to a policy defined value. This could be high for lower priority reservations so that they are more likely to be sent on less desirable routings. It could also be very low for high priority routes in order to promote stability. It could also

depend on the type of application since some may handle the reservation handover delays better.

Switching also promotes load balancing since routes that are becoming saturated will have a lower % bandwidth available and thus be less desirable when multiple routes are probed by the maintenance algorithm. This information could also influence the value of X since much better routes should be chosen more frequently than roughly equal routes. The definition of “better” needs some understanding, but initially will be a bandwidth improvement threshold (in %) of an alternate route over the existing route.

Currently this mechanism has not been included in the prototype implementation. There will need to be more investigation on this mechanism to ensure that route flapping is not introduced and to determine the likely significant overhead introduced.

8. Advantages of this Approach

This section discusses various advantages and additional mechanisms that relate to the resource reservation service.

8.1 Load balancing

Load balancing may be achieved in this scheme over both a short and a long time scale. In the short term, the use of multiple probes and the selection of routes that do not pre-empt or have the most residual bandwidth will ensure that when multiple links are available they will be used roughly equally. This is one of the main advantages of this approach over RSVP which probes only a single route.

In the longer term, the Route-Update algorithm may discover routes with lower utilization (with regard to reserve-able bandwidth) and low priority flows have a high probability of switching to the new route, thus distributing the load on the network. High priority flows are more likely to remain on the same route in order to avoid potential churn while redirecting. This algorithm has not been implemented and will need to be tested for stability since route flapping may be introduced if the switch threshold is set too high.

Another method of achieving load balancing in the network would be by rerouting non critical traffic (traffic without a reservation.) This could be accomplished by forcing all non management traffic off a particular low bandwidth link (for instance to ensure good QoS over LOS HF). This may be included as part of the routing system (included as a routing policy) but is beyond the scope of this report.

8.2 Fault-Tolerance: Timeouts

Included in the admission control algorithms are a number of timeout mechanisms to ensure that resources that are no longer required are released for use by future reservations.

First, partial reservations are given a relatively short timeout period. Since these reservations are only placeholders and do not provide utility for resources held it is important that they timeout as soon as possible but not before a reservation would be confirmed. A possible initial value, that needs to be verified, is twice the transit time of the width of the network plus the worst case decision time on all nodes to the receiver and reservation time on the return path. This is likely to be in the order of single digit seconds (due to satellite delay).

Confirmed reservations may also time out. This may happen because the maintenance algorithm has found a better route and this particular node is no longer needed, or the reservation window or application may have finished. There is a trade-off here in network load vs. over-reservation of the link. Since reservations can still be pre-

empted by higher priority flows it is proposed that the balance on reduced link load be followed and thus timeouts be longer. Contributing factors will also be the level of mobility, the average length of reservations and average available bandwidth. Initial values are included in Annex A.

8.3 Fault Tolerance: Acknowledgements

There are currently acknowledgement messages for service request, commit, and release messages. They have been added to deal with the inherently unreliable nature of the wireless medium. Other types of messages have been left un-acknowledged since the timeout mechanism described above will have the desired effect in these cases.

8.4 Bidirectional Reservations

Depending on the application type, it may be necessary to support bidirectional reservations. Reservation may need to be performed in the two directions prior to data flow (source to destination as well as destination to source) in the case for application traffic which requires interactivity between the two participants that is somewhat time critical, such as for VOIP, VTC and chat applications.

In such cases, it is important that the reverse reservation be handled transparently by the reservation system (this can be done based on the traffic type). One inefficient way to do this would be to perform two consecutive reservations: first source S makes its reservation to destination D. Upon success, destination D initiates a reservation to source S, assuming the same resource requirements (bandwidth, priority). This two step process is simple but not attractive since it generates twice the signalling overhead and results in a long time to establish the full 2-way reservation (this is what RSVP does). Also, the route taken by the forward direction may not be available in the reverse direction at the same time.

Assuming the bi-directionality of the links (already currently assumed by the OSPF protocol), a much more efficient way to address the issue is to perform the reverse reservation at the same time as the forward reservation. In other words, as the service request progresses from the source to the destination, each intermediate node will proceed with the admission control decision algorithm in both directions at the same time. It should be noted here that if more than one link exist between two nodes, when performing the admission control for the reverse reservation, an intermediate node may choose a link that differs from the link chosen in the forward direction. For example, in Figure 1, the intermediate node 2 (Router 2) could choose the link that corresponds to the interface if1 of Router 1 for the reverse reservation even though the forward reservation was performed on link with the if2 interface. The link selection follows the admission control algorithm and will choose the best link (i.e. link with the least preemption and greater residual bandwidth).

Each intermediate node will update the probe accordingly i.e. the number of pre-empted flows will be the sum of the two directions while the residual bandwidth will

be the least of the two directions. An example of how bidirectional reservations work is given in Annex B.

A potential enhancement to bidirectional flows would be the ability to specify different resource requirements in each direction. For instance if a video were to be downloaded from a central server it would be useful to reserve a large amount of bandwidth in the reverse direction (from the server) and only a small amount in the forward direction (to the server). This capability is left as future work.

8.5 Reservations Made In Advance

It is debatable whether it makes sense to support the possibility to perform reservation in advance (i.e. with a non-immediate start time) in a maritime environment. Topology changes with time and links are not stable. Therefore, one can argue that reservations made ahead of time do not stand a lot of chance to succeed. Although this last part is true, it is still interesting to be able to support a principle of "first come first serve". In order to support such a principle, reservations need to be processed as they are submitted in the system. It is thus decided that for now, the following will be supported:

As reservations are submitted, the admission process is performed immediately (admitted or rejected decision) while the enforcement process is performed only at start time (just before the reservation becomes active). If while a reservation is "dormant" an event occurs (topology change, link failure, and preemption) such that the reservation cannot be met anymore, the system could:

- notify the user and let the user decide if he wishes to try again later.

or

- transparently to the user, periodically (value to be determined) attempt to re-establish the reservation again. If still unsuccessful when the reservation is about to start, it then notifies the user.

Reservation made in advance implies some sort of time synchronization of all domain nodes i.e. machines must have time information that is not too far apart (say less than 30secs). For example, time differences between machines may lead to wrong behaviour:

- When start time of a reservation has arrived (the reservation needs to be enforced), if the nodes have too much difference in time, timeouts on expected Refresh will occur.

- Similar problems may occur at stop time. If some nodes reach stop time prior to others, they may receive refresh messages after deletion of the reservation. According to the protocol, this will make them issue Release messages when in fact they should not.

In order to avoid a potential problem at stop time, it was decided that the source would stop sending Refresh messages approximately 20 seconds (value smaller than Refresh timeout) prior to the end time of a reservation. Another possible solution would have been to control the ending of reservations via the source node only. Nodes other than the source would not act on the stop time. At the end of a reservation, the source stops sending Refreshes. The Refresh timeout would be used to cancel the reservations. This solution would be acceptable but requires enough time between reservations in order to ensure that a previous reservation is completed prior to starting a new one (to respect the admission module which has based its decision on the stop time of the reservations).

Of course, there is no problem if reservations are not made in advance. In the case of immediate reservation support only, time synchronization is not required. As a reservation is accepted (committed) it is immediately enforced. Stopping of a reservation could be done strictly by the source node (stopping of maintenance) as desired by the user. This could be an explicit cancellation or after a defined period of time.

9. Preliminary Simulation Results

The commercial tool OPNET [5] has been used to simulate the policy-based admission control (PBAC) protocol. The first simulation experiments involved a small network of one NOC and three maritime nodes as shown in Figure 17 below, and a large network with one NOC and 7 simulated maritime nodes as shown in Figure 18.

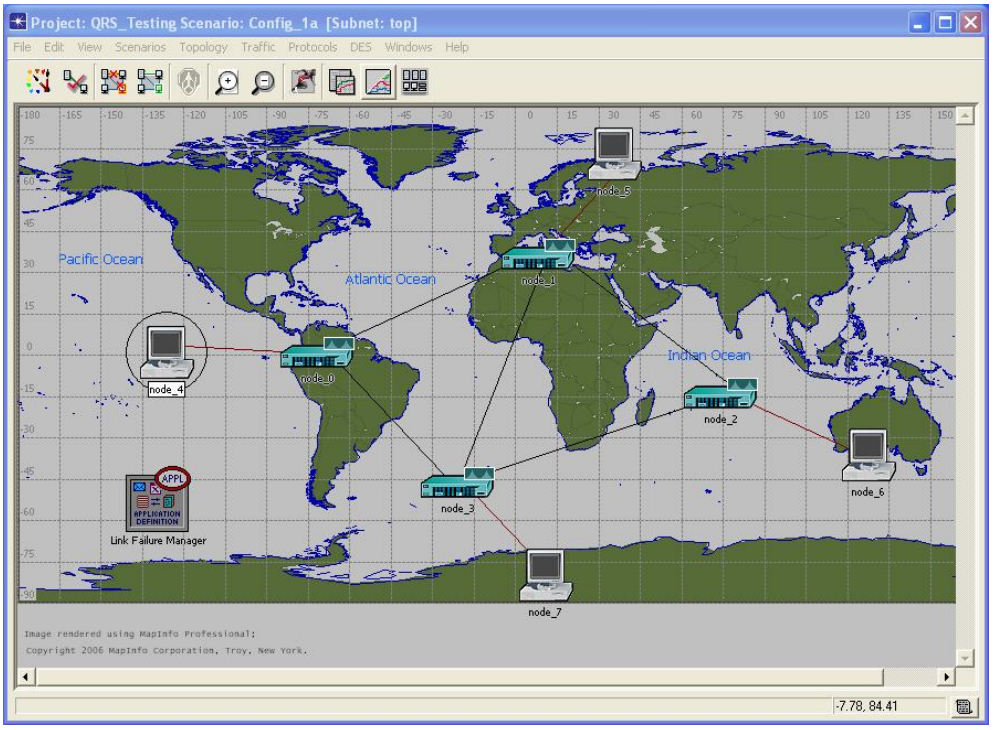


Figure 17. *Simulated Network – Small*

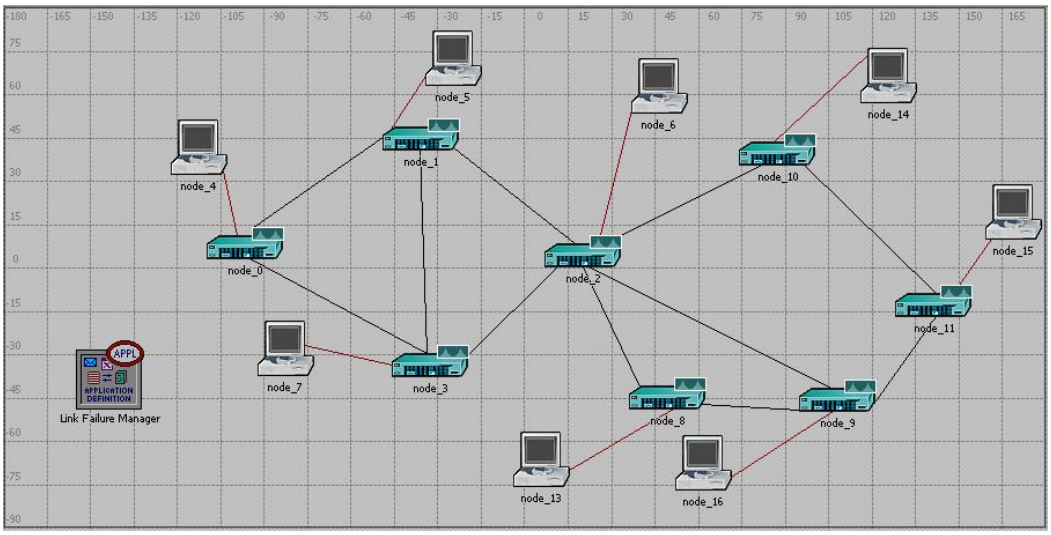


Figure 18. *Simulated Network – Large*

The (simulated) small network has four routers and four workstations and the large network has eight routers and eight workstations. The routers use the base Cisco 3640 model with a modified OSPF stack that forwards the LSA graph to the appropriate PBAC process in its associated workstations. The workstations use the base Intel advanced model with the addition of a PBAC process and packet generation process (used to initiate new requests locally). The routers are connected to their associated workstation with 10BaseT Ethernet and to each other by point to point links with bandwidth between 32 and 128 kbps. In the scenarios pictured above these links have been given the bandwidth and OSPF link costs described in Annex A.

To study the performance of the protocol we measured the overhead introduced by the PBAC service and the call acceptance and preemption rate at both low and high service request load. In order to remove the effects of a particular seed value five runs at each load level are performed with different seeds. Statistics are averaged over those seeds.

Simulation runs of 10000 seconds have been used with 3 priority levels, up to 3 parallel probes, and with a reservation inter-arrival time exponentially distributed and centered on 60s for low service request load and 20s for high request load. Note that this is the rate of reservations originating from each node. All reservations are for 16kbps. Link outages have been disabled, but will be investigated in future simulations.

The lifetimes of reservations are controlled in the simulation when a 20 second refresh interval is reached. 5% of the time a reservation will be terminated at this point. This leads to reservation lifetimes similar to that shown in Figure 20.

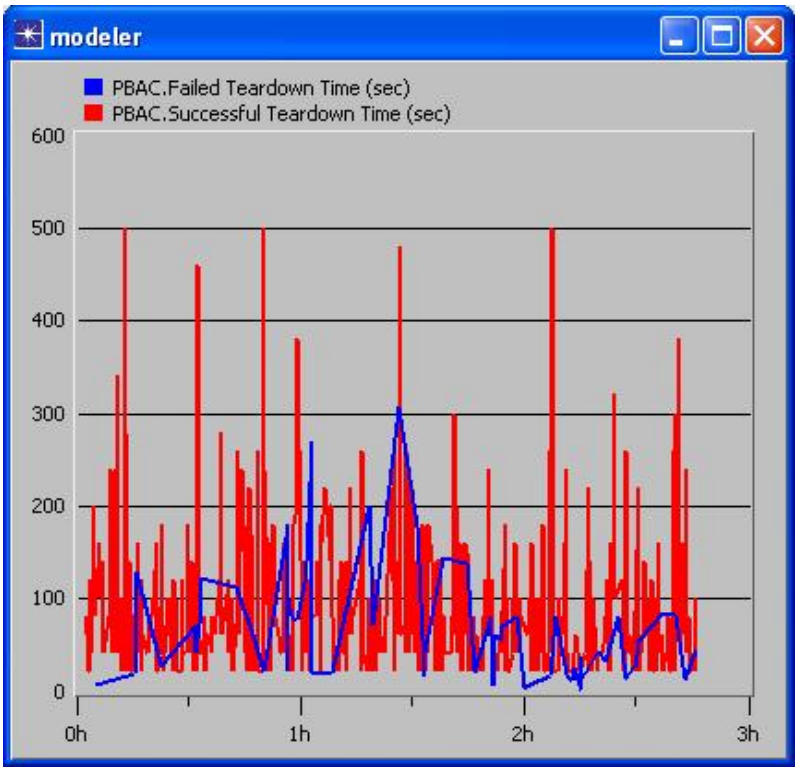


Figure 19. *Reservation Times for Failed and Released Reservations*

The above graph also shows that, as can be expected, the reservations that successfully terminate (are terminated at the refresh interval) last longer than those that have failed (end prematurely) due to preemption or link failures. Note that failed reservation may not be admitted at all (due to a lack of resources) which gives rise to the sub-20s elements on the graph.

The results of the initial simulations are shown in Table 2 below.

Table 2. *Initial Simulation Results*

Network Size	Load	Per Reservation Overhead (bytes)	Call Acceptance Rate	Call Preemption Rate
Small	Low load	13452	89.0 %	7.4 %
	High load	9166	53.3 %	27.0 %
Large	Low load	15538	66.1 %	14.1 %
	High load	10838	34.4 %	20.7 %

Note that current per-reservation overhead is high because it carries the complete XML low level service request which is on average 3015 bytes of data. Overhead is calculated as the sum over each link on which it was sent. This causes the reservation to be split into two UDP packets. We are currently investigating the compression of the service request to improve both overhead and reservation setup time which currently takes 2.5-3.5 seconds on average because of the low bandwidth environment.

Also note the difference in overhead between the low and high load of requests. The lower overhead is likely because when links become congested with requests longer routes are rejected and thus the reservation request does not have to travel over as many links.

As expected, the call acceptance rate is higher and call pre-emption rate lower for lower request loads. It is interesting to note that the large network must handle twice as many total calls as the small network as the same number of reservations is attempted at each node each time period (20s or 60s). If the RSS were not used at the end of the simulation period of 10000s the network would theoretically be loaded at 100% for low load conditions and 260% for high load conditions in the small network and 170% and 430% respectively in the large network. For comparison, with RSS set to allow 100% of the bandwidth to be reserved, it ensures a maximum network utilization of 86% and 100%, in the small network and 81% and 90% for the large network respectively for low and high load conditions. Thus it can be seen that the RRS prevents congestion amongst reserved flows by limiting the number of flows accepted and pre-empting lower priority flows.

These results will be compared with RSVP. It is expected that the multi-probe aspect of PBAC will lead to a higher call acceptance rate and lower call preemption rate but at a higher overhead. This mechanism should provide noticeably better QoS to reserved services than the case where no QoS is implemented, or only the class-based QoS service previously described has been enabled. Though this may be intuitive, qualitative measurements with the implementation will be used to confirm this.

10. Future Work

While many enhancements and modifications to this work have been mentioned in this document, possible future directions for this research are summarised here.

- The maritime links are currently assumed to be bi-directional. The addition of asymmetric links (such as ELOS) would complicate the routing-related services and should be further investigated.

- The route generation algorithms of Section 4.2 currently ignore LOS links since they are inherently unstable. However it may be useful to use paths with LOS links while only making reservations on the more reliable links. This hybrid reservation service should be further investigated.

- Currently resource requests are uni-cast (one source and one destination), a multicast resource request may also be possible if RSVP type mechanisms were to be used to merge multiple reservations.

- Investigate the routing update feature described in Section 7.4 where reservations will adjust their routing to make use of new links or improved routes as they become available. This feature may be useful in particularly large networks.

- Test an alternate link failure mechanism where it is up to each system to index all reservations passing through it and remove the ones that have a failed link (after an appropriate amount of time).

- Study the difference between optimistic and pessimistic resource confirmation where partial reservations are considered or not considered for preemption respectively.

- Enhance bidirectional reservations so that resource requirements (bandwidth) can be separately defined in forward and reverse direction. This could save reserved bandwidth when asymmetric applications are used.

- Another potential enhancement would be the use of MPLS protection mechanisms to allow alternate reserved routes very high priority flows. The alternate routes on “warm standby” would be reserved in advance in anticipation of link outages.

11. Conclusions

The Resource Reservation Service (RRS) is one of four traffic management services designed to improve traffic management in the maritime environment. The RRS and the other traffic services are part of a Policy-Based Traffic Management (PBTM) system that was demonstrated in June 2006.

This document described the RRS and its admission control algorithm. The algorithm makes use of topology information available at every edge router to create routes from the source to the destination of a reservation request. The routes generated are computed so that they do not traverse links with insufficient raw bandwidth and do not use links that have been specified by operational policy as being otherwise unacceptable.

Multiple routes from source to destination are probed in parallel to increase the chance that a route will be found where all intermediate nodes have sufficient resources to admit the resource request. A second advantage of this mechanism is that when multiple acceptable routes are found, the reservations will be balanced across the network.

Two additional advantages of the RRS over similar reservations schemes such as RSVP are the incorporation of fault tolerance schemes and the ability to make simultaneous bidirectional reservations. The use of acknowledgements, timers, and a retransmission scheme are important in error prone environments such as maritime networks. For the same reason, the ability to make reservations for traffic with QoS requirements in both directions at the same time is an advantage (where one alone may fail) as is the savings in overhead and time compared to two sequential reservations.

A preliminary simulation of the protocol has been completed. It showed that the call acceptance rate is higher and the call preemption rate is lower for lower request loads. Also, a lower overhead was observed with higher load, likely because when links become congested with requests, longer routes are rejected and the service request does not travel over as many links.

The RRS was implemented and tested as part of the PBTM research work. It was found to support the features required for an efficient and robust resource reservation scheme to protect mission critical flows in a single domain maritime network. Enhancements, especially as they relate to multi-security and coalition environments are currently under investigation.

12. References

1. I. Labbé, F. St-Onge, D. Kidston and J-F. Roy , “A Policy System for Traffic Management in Maritime Tactical Networks”, DRDC Ottawa TR 2007-005, January 2007.

2. E. W. Dijkstra, “A note on two problems in connexion with graphs” In: Numerische Mathematik. vol. 1, 1959, pgs 269-271.

3. “ACP 200, Maritime Tactical Wide Area Networking (MTWAN)”, AUSCANZUKUS UNCLASSIFIED document, 16 July 2003.

4. R. Braden et al. “Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification”, IETF RFC 2205, September 1997.

5. OPNET web site, last accessed Jul. 11, 2006, http://www.opnet.com/

Annex A: Protocol Details

Details of the proposed protocol for policy based admission control are included here. It includes packet formats, timers and a flowchart outlining the operation of a node upon receipt of a signalling packet.

Packet Formats

This section provides information on the packet formats. Each packet contains a number of fields of given size and description. Additional notes on some fields are provided in the next section.

Request Packet

The **Request** packet is used to forward reservation information on to the next node in the included **Route**. It includes sufficient information for the receiving node to determine if the reservation can be accepted or not.

Field	Size (bits)	Description
Message type	8	Protocol message type. Type = 1
Reservation ID	128	Unique reservation identifier.
Reservation number	16	Probe number
Reservation total	16	Total number of probe sent for this reservation.
Service Request	variable (~7500)	Original XML Service Request submitted by user. This field is serialized as a Java String object.
Number of nodes	16	Number of nodes along the reserved path.
Route	32 * Number of nodes.	List of the IP Addresses for the reserved path.
Forward interface id	32 * Number of nodes.	List of interface number/IP Address.
Reverse interface id	32 * Number of nodes.	List of interface number/IP Address. Only used for full duplex reservations.
Residual Bandwidth	16	Minimum remaining bandwidth on a link after reservation (bottle neck).
Number of pre-empted flows.	16	
Pre-empted Reservation ID	128*N	List of pre-empted Reservation ID.
Bandwidth	16*N	List of bandwidth of the pre-empted flows.

Refresh Packet

The **Refresh** packet is used to confirm the continuation of an existing reservation. If a committed request does not receive a refresh within the **inter-refresh** timeout period, the reservation is considered to have ended and the reservation's resources are released.

Field	Size (bits)	Description
Message Type	8	Protocol message type = 2.
Reservation ID	128	Reservation ID of the one to refresh.
Number of nodes	16	Number of nodes along the reserved path.
Route	32 * Number of nodes.	List of IP Addresses (router ids) for the reserved path.

Commit Packet

The **Commit** message is sent from the destination to the source through all the nodes along the reserved path in order to commit the reservation at each node. It is upon receipt of this packet that resources are set aside for the associated reservation.

Field	Size (bits)	Description
Message Type	8	Protocol message type = 4.
Reservation ID	128	Reservation ID of the one to commit resources.
Probe Number	16	Probe number of reservation to commit.
Number of nodes	16	Number of nodes along the reserved path.
Route	32 * Number of nodes.	List of IP Addresses (router ids) for the reserved path.
"Remote" Forward Interface ids	32 * Number of nodes.	List of interface number/IP addresses of the remote end of the reserved forward interface ids. Built during the commit so the sender can set up the forward tunnel.

Failed Packet

The **Failed** message is sent to the source node from the destination node (or an intermediate note in unicast requests) when there is no path available for the reservation.

Field	Size (bits)	Description
Message Type	8	Protocol message type = 8.
Reservation ID	128	Reservation ID of that was unsuccessful.

Denied Packet

Message sent from a node to destination when the reservation cannot be made along a particular probe's path.

Field	Size (bits)	Description
Message Type	8	Protocol message type = 16.
Reservation ID	128	Reservation ID that was denied.

Reservation number	16	Probe number
Reservation total	16	Total number of probe sent for this reservation.
Source IP Address	32	Source IP address for this reservation.

Release Packet

Message sent from a node to inform that the given reservation is no longer valid and resources should be released.

Field	Size (bits)	Description
Message Type	8	Protocol message type = 32.
Reservation ID	128	Reservation ID that was released.
Route	32 * Number of nodes.	List of IP Addresses for the reserved path.

ACK Packets

This protocol will run on top of UDP/IP. Because UDP offers no guarantee of delivery, some reliability via an acknowledgement-retransmission mechanism is added to the **Request**, **Commit**, and **Release** messages.

Field	Size (bits)	Description
Message Type	8	Protocol message type. Type = 64.
Reservation ID	128	Reservation ID
Reservation number	16	Probe number
Message type	8	Message type for which we send this ACK.

Packet Fields

The protocol runs on top of UDP/IP[6]. The following table show the message type used in the header of each message.

Message Type	Value
Resource Reservation (Partial)	1
Reservation Refresh (Keep Alive)	2
Reservation Commit	4
Reservation Failed	8
Reservation Denied	16
Tear Down	32
Acknowledge	64

[6] UDP port number is configurable. Currently it is configured as port 7227.

Timers

A set of timeouts at each node allow maintaining a consistent state of the system. For instance, if we expect to receive 5 reservation probes, and only 4 have arrived successfully, the system will not block forever waiting for the missing probe.

This section presents the different timer used along with the protocol messages.

Timer	Length (s)	When	Description
Maximum waiting time for reservation probes.	60	The BB that receives a resource reservation message and is the destination node.	This is the time to wait after the destination receives the first probe for the remaining probes (if applicable).
Send periodically refresh messages.	15	The BB is the source node for a reservation.	The timer must periodically trigger the sending of a reservation refresh message to keep it alive.
Inter-refresh arrival monitoring.	30	The BB is a node part of an active reservation path.	This timer is used to make sure we receive periodically refresh messages for a reservation. If we don't receive the message within the given period, the reservation is tear down.
Maximum time for partial reservation commit	80	The BB receives a resource reservation message and is not the destination node.	This is the maximum time to wait for a commit message before cancelling a partial reservation.
ACK receive timer	3	A reliable message is sent.	This timer is used for the guaranteed delivery of messages. If an ACK is not received for the configured maximum wait time, the message is considered lost.

Flowchart

The following diagram (Figure A-1) shows the processing that is done by the system to handle the arrival of a protocol message.

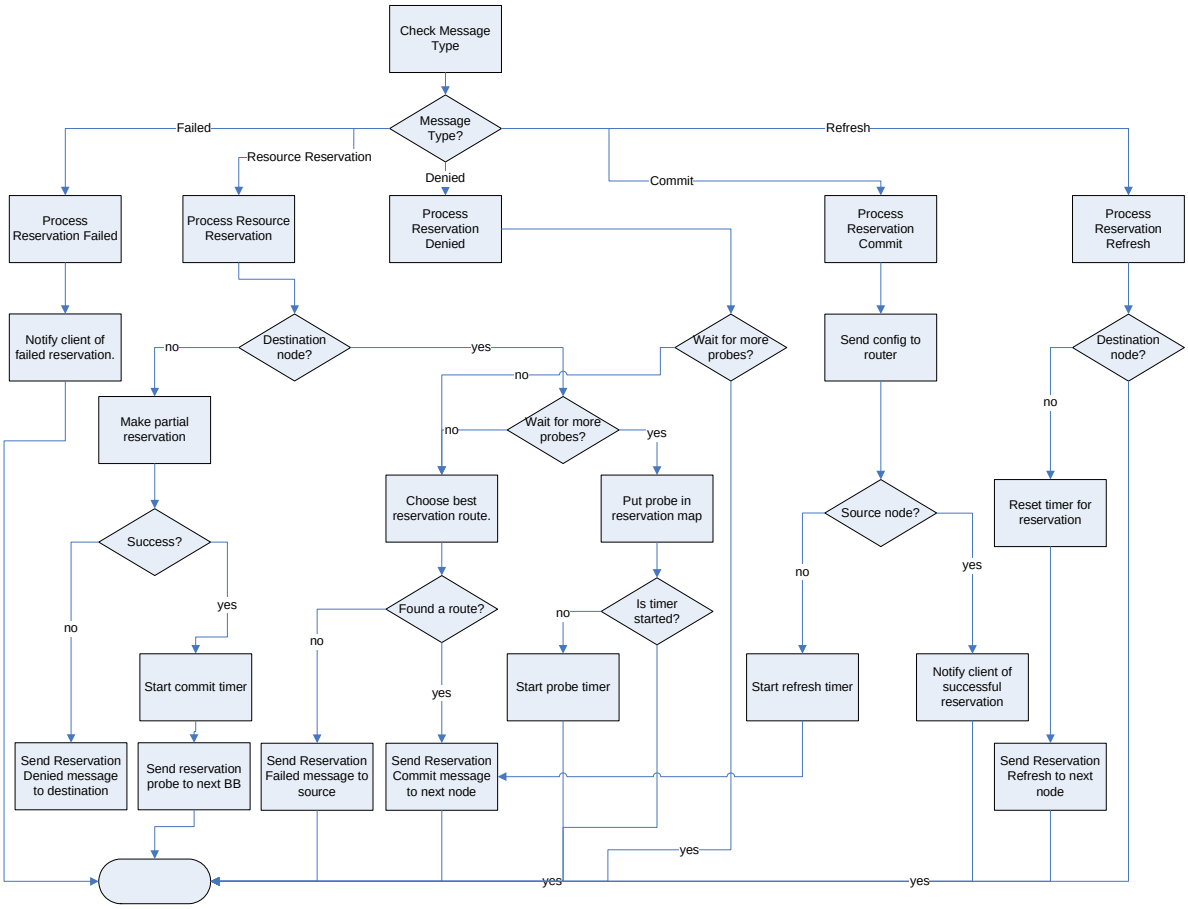


Figure A-1, RSS Packet Processing Flowchart

Annex B: Bidirectional Routing Example

This is an explanation note on the route information that is build up and carried by the Reservation Request message as the message progresses from source to destination. This information is used to set up tunnels for the defined traffic.

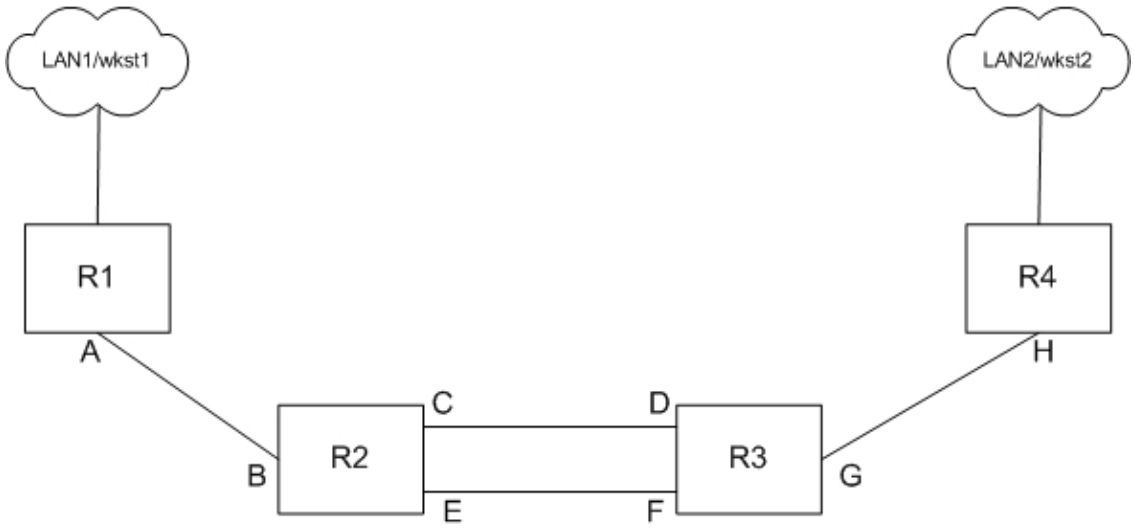
The general format to specify the path is:

{Router node ID/forward path interface}

If the service request is bi-directional, the path information needs to be augmented by the reverse path interface which will be selected at each node as the reservation progresses. The general format for bidirectional reservations becomes:

{Router node ID/forward path interface/reverse path interface}

To illustrate how the route is generated, consider the following network connectivity diagram.



Let us assume the following:

- A service request has been made from LAN1/wkst1 to LAN2/wkst2
- The best route algorithm has selected the following path: {R1/A - R2/C - R3/G - R4/0} (Note that an interface value of 0 implies that no interface is currently specified)
- The service request is bi-directional

As the probe (reservation request message) is sent from source to destination, each node along the selected route updates the information carried by the probe in order to include the reverse tunnel interface while performing access control in both upstream and downstream directions.

At R1:

- The probe is initialized with the selected route: {R1/A/0 - R2/C/0 - R3/G/0 - R4/0/0}
- As router 1 is the first hop along the path, it does not include a reserve interface. It sends the probe to R2.
- The local policy system keeps the following knowledge for this request:

Forward path local interface: A

Reverse path local interface: 0 (N/A since it is a local connection)

At R2:

- R2 receives the probe containing the route: {R1/A/0 - R2/C/0 -R3/G/0 -R4/0/0}
- Since the reservation request is bidirectional, it finds which interface can best accommodate the request. In this case, only interface B is a possible pick. Let's assume that enough bandwidth is available via interface B. It finds that A is the corresponding next hop interface to B and updates the message with appropriate path info: {R1/A/0 - R2/C/A - R3/G/0 - R4/0/0} and sends it to R3
- The local policy system keeps the following knowledge for this request:

Forward path local interface: C

Reverse path local interface: B

At R3:

- R3 receives the probe containing the route: {R1/A/0 - R2/C/A - R3/G/0 - R4/0/0}
- Since the reservation request is bidirectional, it then finds which interface can best accommodate the request. In this case, two interfaces are possible. Let's assume that the best pick is via interface F (refer to Section 7.5 of the thesis for route selection criteria). It finds that E is the corresponding next hop interface to F and updates the message with appropriate path info: {R1/A/0 - R2/C/A - R3/G/E - R4/0/0} and sends it to R4

- The local policy system keeps the following knowledge for this request:

Forward path local interface: G

Reverse path local interface: F

At R4:

- R4 receives the probe containing the route: {R1/A/0 - R2/C/A - R3/G/E - R4/0/0}
- Since the reservation request is bidirectional, it needs to find which interface can best accommodate the request. In this case, interface H is the only possible pick. Let's assume that enough bandwidth is available at interface H. It finds that G is the corresponding next hop interface to H and updates the path info: {R1/A/0 - R2/C/A - R3/G/E - R4/0/G}.
- The local policy system keeps the following knowledge for this request:

Forward path local interface: 0 (N/A since it is a local connection)

Reverse path local interface: H

- Since it is the last node, it is the one that needs to create the reverse tunnel. The node extracts the information from the corresponding fields of the probe to form the reverse tunnel {R4/G – R3/E – R2/A} (last hop).
- Because it is the last node, R4 will issue the commit message. It copies the forward path info in the commit message (no need to send back the reverse path info as it is of no use to the other nodes) and sends it backward towards the source to R3. The commit message thus contains the following route: {R1/A - R2/C - R3/G - R4/0}
- As the commit message is sent from destination back to source, each node along the selected route replaces the local forward interface by the forward tunnel interface (which needs to be expressed with the corresponding next hop interface of the other end of the link):

At R3:

- R3 receives the commit containing the route: {R1/A - R2/C - R3/G - R4/0}
- R3 finds that H is the corresponding next hop interface to G. It updates the path info of the message to: {R1/A - R2/C - R3/H - R4/0} and sends it to R2

At R2:

- R2 receives the commit containing the route: {R1/A - R2/C - R3/H - R4/0}
- R2 finds that D is the corresponding next hop interface to C. It updates the path info of the message to: {R1/A - R2/D - R3/H - R4/0} and sends it to R1

At R1:

- R1 receives the commit containing the route: {R1/A - R2/D - R3/H - R4/0}
- R1 finds that B is the corresponding next hop interface to A. Given that it is the first hop, it now has the information to create the forward tunnel. By extracting the information from the corresponding fields of the message, the forward tunnel is created: {R1/B – R2/D – R3/H} (last hop).

Once the tunnels are created and access control has been completed, the flow will be recognized at each hop and forwarded to the next hop defined by the tunnel. This mechanism is especially useful in maritime environments for its promotion of load balancing when multiple links exist between the same two nodes.

List of Symbols/Abbreviations/Acronyms/Initialisms

BLOS	Beyond Line of Sight
DND	Department of National Defence
ELOS	Extended Line of Sight
HF	High Frequency (radio)
IP	Internet Protocol
LAN	Local Area Network
LOS	Line of Sight
LSA	Link State Advertisement
LSDB	Link State Database
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MIB	Management Information Base
MPLS	Multi Protocol Label Switching
NOC	Network Operating Centre
OSPF	Open Shortest Path First
PBM	Policy-Based Management
PBNM	Policy-Based Network Management
PBTM	Policy-Based Traffic Management
PDP	Policy Decision Point
PEP	Policy Enforcement Point
QoS	Quality of Service
RRS	Resource Reservation Service
RSVP	Reservation Protocol

SOA	Service Oriented Architecture
SNMP	Simple Network Management Protocol
VOIP	Voice over Internet Protocol
VTC	Video TeleConferencing
WAN	Wide Area Network
XML	eXtensible Markup Language

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

   Communications Research Centre
   3701 Carling Av, Ottawa

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

   UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

   A Policy-Based Resource Reservation Service for Maritime Networks (U)

4. AUTHORS (Last name, first name, middle initial)

   Kidston, David , Labbé, Isabelle

5. DATE OF PUBLICATION (month and year of publication of document)

   January 2007

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

   65

6b. NO. OF REFS (total cited in document)

   5

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Technical Memo

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

   Defence R&D Canada – Ottawa
   3701 Carling Avenue
   Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

   15cv

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

   DRDC Ottawa TM 2007-006

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

   CRC TN 2006-002

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

   (x) Unlimited distribution
   ( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
   ( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
   ( ) Distribution limited to government departments and agencies; further distribution only as approved
   ( ) Distribution limited to defence departments; further distribution only as approved
   ( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

   Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

As part of a research effort to provide enhanced communications capabilities in a maritime tactical network, a Policy-Based Traffic Management System was developed to dynamically control link usage and perform end-to-end QoS. A desirable management service in the maritime network is end-to-end guaranteed QoS for critical application flows. This type of service is most commonly useful for real time applications (such as VOIP or video), but can also be used for critical data transfers (such as a specific image transfer or chat session). By entering a service request, an operator can specify the application flow to be considered, the desirable treatment (currently in terms of bandwidth only), as well as any other pertinent information including the priority of the request.
A policy-based resource reservation service has been developed that guarantees bandwidth for an application flow by per-hop admission control. This document describes the resource reservation service (RRS) including the design decisions and the algorithms developed to support it. The RRS is organised around a main admission control algorithm that consists of four phases. In the first phase, global link information is used to generate potential routes between the source and destination of the requesting flow. The second phase probes the potential routes to determine if sufficient resources are available on all links. In the third phase, an acceptable path is selected and committed. Finally in the fourth phase the reservation is maintained until the flow terminates or the network can no longer support its requirements.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Policy-based network management, QoS, traffic management, resource reservations, mobile networks

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca