RSVP Performance Evaluation using Multi-Objective Evolutionary Optimisation

Olufemi Komolafe and Joseph Sventek
Dept. of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.
Email: {femi, joe}@dcs.gla.ac.uk

Abstract— **The proposed uses of the resource reservation protocol (RSVP) now extend beyond reserving resources in Internet Protocol (IP) networks to being a generic signaling protocol for generalised multi-protocol label switching (GMPLS). In any implementation of RSVP, there are a number of discretionary timing parameters, the values of which affect the efficacy of RSVP in establishing and maintaining reservations/connections. This work frames the interactions between key RSVP timing parameters and performance metrics as a multi-objective optimisation problem which, due to its intractable nature, is tackled using a reputable multi-objective evolutionary algorithm. It is shown that this approach is a feasible means of exploring many of the innate tradeoffs in soft-state protocols such as RSVP. This approach facilitates an extensive comparison of a number of variants of RSVP: standard RSVP, RSVP featuring the recently standardised retransmission algorithm and two subsequent variants of this algorithm, supporting the asymmetric delivery of RSVP control messages. These RSVP variants are compared in terms of multiple performance metrics under a number of different exemplar network conditions, giving insight into their relative merits. Furthermore, the relative significance of the different timing parameters is investigated and their most expedient values determined.**

I. Introduction

The resource reservation protocol (RSVP) [28], [8] was originally proposed for resource reservation in Internet Protocol (IP) integrated services networks. However, RSVP has been attracting considerable attention for use in multi-protocol label switching (MPLS) networks [23], with the intention being to use RSVP as a signaling protocol to establish label-switched paths (LSPs) [2]. Recently, MPLS signaling protocols, along with IP routing protocols, are being standardised as a control plane for packet, time-division, wavelength and spatial switching networks. Generalised MPLS (GMPLS) [3], as this control plane is known, will therefore potentially use a suitably enhanced form of RSVP. Another signaling protocol that may be used for (G)MPLS signaling is the constraint-based routing label distribution protocol (CR-LDP) [13]. An implementation study found that the uses of RSVP far outnumbered that of CR-LDP for GMPLS signaling [6]. These findings dissuaded the IETF from carrying out further work on CR-LDP, focusing exclusively on RSVP for (G)MPLS signaling [1].

Hence, it is reasonable to conclude that RSVP will have a key role to play in future IP-centric networks, albeit in a different form and role than originally envisaged. Therefore, given the increase in the variety and importance of uses for RSVP, it is becoming increasingly important to study the behaviour of RSVP thoroughly. In any implementation of RSVP, there are a number of discretionary timing parameters, the values of which affect the efficacy of RSVP in establishing and maintaining reservations or connections. This work frames the interactions between RSVP timing parameters and network performance metrics as a multi-objective optimisation problem which, due to its intractable nature, is tackled using a reputable multi-objective evolutionary algorithm. The adopted approach provides insight into the most expedient RSVP timing values for a number of different scenarios and allows alternative mechanisms for delivering RSVP messages to be compared.

Section II provides an overview of RSVP, discussing related work and relevant, recently-proposed enhancements to the protocol. The problem being tackled is defined in Section III, and multi-objective evolutionary optimisation is described in Section IV. Section V presents the design and implementation used. Section VI justifies the evolutionary algorithm parameters chosen. Sections VII and VIII present the results obtained. Section IX concludes the paper and suggests avenues for future work.

II. RSVP Overview & Related Work

A. RSVP Description

The primary control messages in RSVP, Path and Resv messages, originate from the senders and receivers, respectively. Path messages follow the route computed by the routing protocol and provide receivers with the description of the sender and traffic flow. Upon receipt of a valid Path message, each intermediate RSVP-capable router updates or creates a Path state entry for the sender before forwarding the appropriately updated Path message towards the receiver. After receiving a Path message, the receiver can make a reservation by sending a Resv message back to the source. RSVP is a soft-state protocol. Hence, Path and Resv messages must be exchanged regularly between the routers in order to maintain the reservation. The frequency with which Path and Resv messages are sent is determined by the soft-state refresh period. Although the Path state and the Resv state will eventually timeout if not refreshed, PathTear or ResvTear messages may be used to tear down state promptly.

B. Related Work on Soft-state Protocols

Some previous studies on soft-state protocols are applicable to RSVP and are relevant to the work in this paper. Scalable

0-7803-8968-9/05/$20.00 (C)2005 IEEE

timers, where the refresh period increases proportionally with the amount of state to be refreshed, have been studied and an algorithm for dynamically adjusting the sender's refresh rate and estimating the sender's refresh rate at the receiver was proposed [26]. The first significant analytical model for soft-state protocols was developed in [22]. The main performance metric considered was the probability of the sender and receiver having consistent state. A key finding was that incorporating feedback into soft-state protocols improves the network consistency without incurring excessive network resource consumption. Another analytical model for soft-state signaling protocols was presented in [14], allowing the comparison of a spectrum of signaling protocols from "pure" soft-state to soft-state augmented with explicit state removal and/or reliable signaling to "pure" hard-state protocols. Many interesting results are presented in [14], including the observation that reliable, explicit, state establishment/update/removal improves the performance of soft-state protocols significantly.

C. Related Work on RSVP Performance Evaluation & Improvement

The performance of RSVP has been evaluated using an industrial-strength RSVP implementation on a commercial IP router [17]. Performance metrics considered included the connection set up time, soft-state refresh overhead and the impact of real-time packet scheduling. Emphasis was placed on designing the RSVP protocol engine optimally in [15] and, using the resulting RSVP implementation, the performance of RSVP was studied. Interestingly, the results suggested that the scalability of RSVP is better than is traditionally assumed [15]. An exponential back-off retransmission algorithm to deal with the loss of initial, or trigger, control messages was proposed [21]. The performance of the retransmission algorithm has been subsequently evaluated [19], [16]. In addition to discussing the retransmission algorithm, [18] suggests that RSVP neighbours exchange heartbeat messages to allow prompt failure detection and potentially reduce the protocol overhead by maintaining soft-state on a per-neighbour rather than a per-flow basis. Attention was also paid to reducing the protocol overhead in [27], where the compression of state (that would normally be refreshed via multiple messages) into a single digest message was studied.

D. Key Stages in the Evolution of RSVP Standards

RSVP standards have evolved from the original specifications [8] to reflect some of the developments previously discussed. An early, significant modification to RSVP was to introduce bundle and summary refresh messages in an attempt to make RSVP more scalable [5]. Based on earlier findings [21], RFC 2961 introduced a retransmission algorithm to improve the reliability of control message delivery. Essentially, when a node sends a trigger Path or Resv message, it will indicate that it requires an acknowledgement by setting a flag in the message. The node retransmits the message, at an interval determined by an exponential back-off timer, while awaiting the Acknowledgement message or until a retry limit has been reached. Thereafter, refresh RSVP messages are sent as normal. The desire to use RSVP as a generic signaling protocol for MPLS led to the definition of RSVP with traffic engineering extensions (RSVP-TE) [2]. Most recently, extensions to RSVP-TE have been defined for GMPLS signaling [7].

III. Problem Definition

A number of discretionary timing parameters are pivotal to the performance of RSVP. The key timing parameters considered in this work were defined in RFCs 2205 and 2961 and are described in Table I. As shown in Table I, certain default values are suggested for these parameters. Most work on RSVP, for example [15] - [19], [21], [27] typically use these default timing parameters in their studies. Hence, it is fair to say that little attention has been paid to exploring the interactions between the timing parameters and establishing the values that optimise the performance of RSVP. This paper proposes and implements a method for finding the optimal values of the timing parameters. Essentially, these timing parameters are the inputs into an optimisation problem, the aim of which is to optimise the performance of RSVP in terms of a number of appropriate performance metrics. It is known that a tradeoff typically exists between the protocol overhead and most other performance metrics in a soft-state protocol such as RSVP [14]. However, to date, there has been no significant effort to explore the nature of this tradeoff. A multi-objective optimisation technique that allows the simultaneous optimisation of conflicting objectives is used in this paper, allowing the tradeoff between protocol overhead and a number of performance metrics to be studied.

TABLE I
KEY RSVP TIMING PARAMETERS

Parameter	Symbol	Explanation	Default value
Refresh period	R	Interval between successive refreshes (timer randomly set in range $[0.5R, 1.5R]$)	30s
Lifetime factor	K	Number of successive refreshes that must be lost before state timeout	3
Rapid retransmission interval	R_f	Initial retransmission interval for unacknowledged messages	0.5s or RTT (if known)
Rapid retransmission limit	R_l	Number of retransmissions permissable without acknowledgement	3
Increment value	Δ	Ratio between successive retransmission intervals = $\Delta + 1$	1

If RSVP is being used to establish a reservation for a flow in a congested network, it is desirable to optimise a number of network performance metrics. Four such metrics are considered in this work. The first performance metric considered is the *reservation establishment time*, the interval between when the source sends the first Path message and when it receives the first Resv message. Since an often cited criticism of RSVP relates to the traffic overhead, the *protocol overhead*, approximated using the total number of RSVP bytes exchanged, is the second performance metric considered. The reservation should be maintained for the required duration, rather than being torn down prematurely due to the occurrence of state timeout. Therefore, the last two performance metrics considered are the mean *end-to-end delay* and *loss* experienced by packets in the flow; these metrics reflect the ability of RSVP

to maintain reservations. These objectives are all indicative of the efficacy of RSVP in establishing and maintaining reservations. Hence, any findings are applicable to the use of RSVP to establish reservations in IP networks or, since the operation of RSVP will remain largely unchanged, to the use of RSVP as a generic signaling protocol in a GMPLS-based control plane.

Formally, the problem statement is: *Find the values of the RSVP timing parameters (refresh period, lifetime factor, rapid retransmission interval, rapid retransmission limit and increment value) that simultaneously optimise a number of network performance metrics (reservation establishment time, protocol overhead, mean flow end-to-end delay and loss) and study the interactions between the inputs and objectives for variants of the proposed retransmission algorithm under a number of different network conditions, comparing the results with the performance of standard RSVP.*

IV. Multi-Objective Evolutionary Optimisation

Many real-life optimisation problems involve multiple, often conflicting, objectives. The conflicting nature of the objectives typically gives rise to a set of compromised solutions. Figure 1 shows all possible solutions to an exemplar minimisation problem with two conflicting objectives. In such cases, Solution *A* is said to *dominate* Solution *B* if *A* is better than *B* in at least one objective and is no worse in all other objectives. If the entire solution space is considered, a set of solutions that are not dominated by any other solutions will typically exist. Such solutions, highlighted in Figure 1, are known as Pareto-optimal (PO) solutions. The goal of multi-objective optimisation is to find as many solutions in the Pareto-optimal set as possible.

The desire to find a set of Pareto-optimal solutions means that multi-objective optimisation lends itself naturally to genetic algorithms (GAs), since GAs consider a population of solutions. Merits of applying GAs to multi-objective optimisation include:

- The possibility of finding multiple solutions in a single optimisation run.
- The ability to optimise multiple (conflicting) costs simultaneously, avoiding the non-trivial problem of formulating a representative aggregated cost function.
- Minimal prior problem knowledge is required.
- Performance is not constrained by the shape and continuity of the search space.
- Mechanisms to ensure the emergence of a diverse final population may be easily incorporated.
- GAs perform well when faced with noisy or stochastic objective functions.

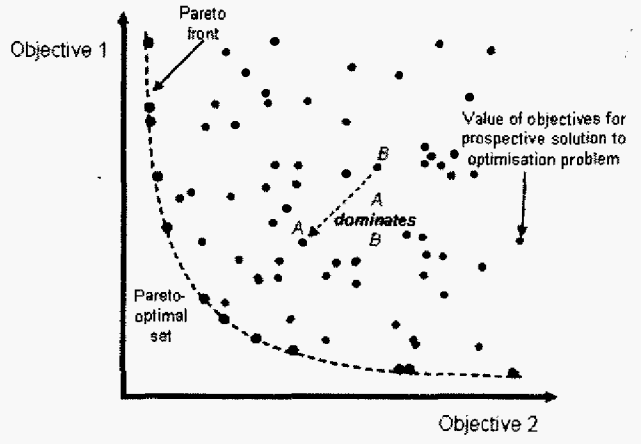

Fig. 1. All possible solutions to exemplar bi-objective minimisation problem.

Consequently, significant research effort has been devoted to the application of GAs to multi-objective optimisation problems and a number of alternative algorithms have been proposed and evaluated [9], [29]. The particular multi-objective evolutionary algorithm used in this work is the non-dominated sorting genetic algorithm II (NSGA-II) [10]. NSGA-II, as its name suggests, is an enhancement of NSGA [25], a multi-objective evolutionary algorithm that is known to outperform many other such algorithms [29]. The NSGA-II algorithm, illustrated in Figure 2, is described in greater detail in [10]. The approach used to apply NSGA-II to RSVP performance evaluation is explained in Section V.

V. Design and Implementation

A. Problem Encoding

In genetic algorithms, a set of problem input values corresponding to a prospective solution is typically coded as a binary string. When considering multiple inputs, the convention is to code each input as a sub-string in the overall binary string. The binary string used is illustrated in Figure 2. The range of values for the inputs, the timing parameters, are chosen to give a relatively good spread and included the suggested default values. (It should be noted that the recommendation that the refresh period should be randomly set around the nominated value [8] cannot be followed in this work; such randomisation will make it difficult to study the impact of the refresh period, hence refresh messages are sent exactly every refresh period.) The fitness associated with an individual is the result obtained when the objective functions are evaluated for this set of input values.

B. Objective Evaluation

Objective evaluation is carried out using simulation. Network Simulator (NS-2) [20] is a widely used tool for studying data networks. An RSVP simulator, RSVP/ns [24], has been implemented in NS-2. For this work, RSVP/ns was enhanced to implement the exponential back-off retransmission algorithm of unacknowledged trigger Path and/or Resv messages recommended in RFC 2961. The timing parameters used for RSVP in the simulation are obtained by decoding the corresponding binary string. Figure 2 illustrates the exemplar topology used for objective evaluation in this work.

In the simulation, the sender sends a Path message to the receiver and simultaneously starts sending it a constant bit rate (CBR) flow. The five intermediate RSVP-capable routers

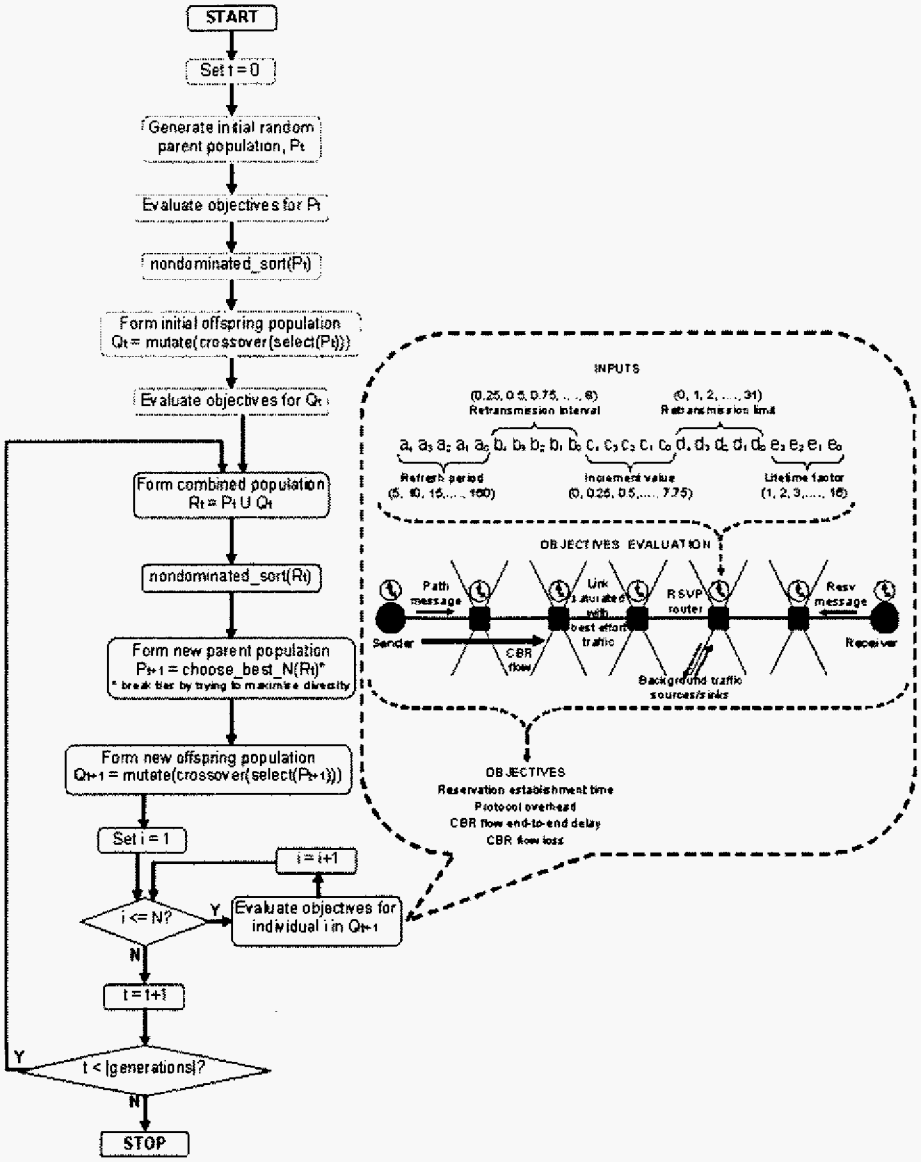

Fig. 2. NSGA-II algorithm and methodology adopted.

establish the Path state specified by the Path message. Upon receipt of the Path message, the receiver generates a corresponding Resv message which is sent hop-by-hop back to the sender, with the intermediate routers reserving the appropriate resources for the flow. The reservation establishment time can be calculated once the sender receives this first Resv message. Thereafter, refresh Path and Resv messages are sent regularly in order to maintain the reservation. The protocol overhead is determined by aggregating the total number of RSVP bytes exchanged throughout the entire simulation. The links between the routers are all saturated with background best-effort traffic. Hence, if no reservation is made for the CBR flow, it will contest for resources with the background traffic. The link capacities were set such that the aggregate traffic exceeds the link capacities, therefore a proportion of the CBR and best effort traffic must be buffered and possibly lost. However, if a reservation is successfully made for the CBR traffic, only the background traffic should be buffered and potentially dropped. Consequently, determining the mean delay and loss of packets in the CBR flow gives insight into the ability of RSVP to establish and maintain reservations for a given set of timing parameters values.

RSVP/ns implements WFQ [11] and WF^2Q [4] to enforce bandwidth guarantees. The CBR source emitted 500byte packets at a rate of 400kbit/s. The buffer size was nominally set to 1000 packets. The default background best effort traffic was produced by multiple exponentially distributed On/Off sources that generate 500byte packets at a rate of 500kbits/s during On periods. The mean On and Off periods were both 1s. There are four background traffic sources/sinks connected to each of the five intermediate RSVP-capable routers, with each router exchanging a stream of the background traffic

with every other router. The capacities of the outer pair and the inner pair of the four links between the five intermediate routers are set to 0.5Mb/s and 0.75Mb/s. (The ratio between these values corresponds to the relative volume of background traffic carried by the links.) Additionally, link capacities of 1Mb/s and 1.5Mb/s are also considered to observe the impact of decreasing congestion. For brevity, only the capacity of the outer pair of links (i.e. the lower of the two link capacities in each pair) is quoted when referring to either these two scenarios henceforth.

The exemplar network topology and scenarios used for objective evaluation in this work were chosen as a relatively simple and yet interesting and realistic configuration to explore the interaction between RSVP timing parameters and certain performance metrics, using multi-objective evolutionary optimisation. Hence, any insights gained from this configuration may be leveraged when considering different and/or more complex configurations.

C. RSVP Variants

In this work, the performance of RSVP using the retransmission algorithm proposed in RFC 2961 is compared against three other possible implementations of RSVP. It is interesting to ascertain if and when the retransmission algorithm outperforms standard RSVP. The two key discretionary timing parameters in standard RSVP are the refresh period and the lifetime factor. Since there are only two inputs for this optimisation problem, an exhaustive exploration of the search space was conducted; the refresh period was varied from 5s to 160s, in 5s increments, and the lifetime factor increased from 1 to 16. The objectives corresponding to each of the 512 combinations can be evaluated through simulation. Another motivation behind using an exhaustive search is to verify the GA results. Essentially, obtaining comparable results from the exhaustive search and GA, two techniques that seek optimal solutions in radically different ways, ought to engender confidence in the findings presented in this paper.

Two further variants of the basic retransmission algorithm proposed in RFC 2961 are considered. This step is motivated by the earlier observation that the loss of Path messages was found to be typically more detrimental than the loss of Resv messages [16]. In order to verify this hypothesis further, the two variants of the algorithm implemented are the retransmission of only unacknowledged trigger Path messages or only unacknowledged trigger Resv messages. NSGA-II is also used to seek the optimal solutions when the retransmission algorithm is implemented asymmetrically and the encoding introduced in Section V-A is used. Note that when only Path (Resv) messages are retransmitted, Path states and Resv states have the same lifetime factor and refresh rate for each individual being considered but obviously the retransmission interval, retransmission, retransmission limit and increment value only apply to Path (Resv) messages.

VI. Selection of Optimisation Parameters

In the encoding described in Section V-A, there are 2^{24} possible combinations of the input timing parameters. The computation time to evaluate the objectives for each set of inputs by simulation was found to be approximately 7 minutes. Each time the simulation was conducted, different random generation seeds are used to produce the background traffic; this approach appears to be more credible and generic than the alternative of using the exact same background traffic for different simulations. By investigating the impact of increasing the number of simulation runs on the variance of the mean of the simulation results, it was found that up to 10 different simulation runs were needed before objectives converged to a steady average. Consequently, the total time needed for an exhaustive search of all possible timing parameters values was approximately $1.17 * 10^9$ minutes or 2,234 years.

Genetic algorithms are known to be well-suited to this kind of intractable problem. However, the need for multiple simulation runs to evaluate the objective function accurately poses the question that, given a fixed computation time, should more effort be expended in evaluating the objectives accurately (necessitating a large number of simulation runs for each individual in the GA population) or should a larger number of roughly evaluated solutions be considered by the genetic algorithm (resulting in a larger population and/or number of generations)?

Fortunately, the tradeoffs implicit in the application of GAs to noisy or stochastic objective functions have been previously considered and it has been found that the overall efficiency of genetic algorithms may be improved by reducing the time spent on accurate objective evaluation and increasing the population size and/or number of generations [12]. It was also suggested that, although objectives are approximately evaluated during the GA run, once the GA finishes, the objective functions of individuals in the final population should be accurately evaluated.

Hence, the NSGA-II parameters used to study the impact of the timing parameters on the performance of RSVP were determined empirically, using the approach suggested in [12]. In this work, carrying out a total of 1280 objective evaluations during a GA run was considered to be practical. The results obtained by NSGA-II when this total is reached using different combinations of population size, number of generations and simulation runs per individual was investigated, with the link capacity set to 0.5Mb/s. The objectives in this exemplar optimisation were the protocol overhead and the loss in the CBR flow for which RSVP is attempting to reserve bandwidth. As suggested in [12], the objectives for individuals in the final population were accurately evaluated by repeating the simulation 10 times and finding the mean results. The best (i.e. non-dominated) solutions in the final populations for each different combination are shown in Figure 3.

It is evident from Figure 3 that improvements are obtained over the exemplar initial randomly generated population, regardless of the population size, number of generations and

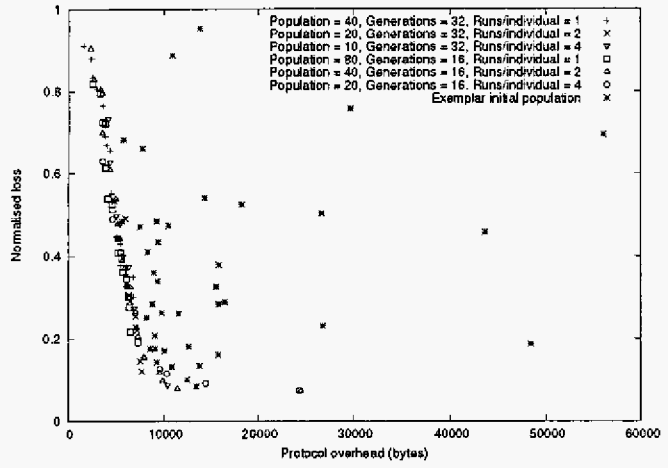

Fig. 3. Impact of different combinations of population size, number of generations and simulation runs/individual.

number of runs/individual. Secondly, it can be seen that all the different cases converge on a similar set of final non-dominated solutions. The fact that these different GA runs, with different randomly generated initial populations and featuring different population sizes, number of generations and number of runs/individual, all converge on a similar set of solutions suggest that these solutions may indeed be very close to the Pareto-optimal set. Hence, it was felt that any of the cases considered could be justifiably used for the optimisation problem tackled in this work. A population size of 40, 16 generations and 2 simulation runs/individual is chosen since, from Figure 3, it can be seen that the solutions for these values exhibit the greatest diversity in the final population.

VII. Multi-objective Optimisation Results

A. Single Reservation with Background Traffic from Exponential Distribution

Having selected a suitable population size, number of generations and number of runs/individual, the simultaneous optimisation of the protocol overhead, the reservation establishment time (RET) and the CBR flow loss and delay was carried out. Note that if the reservation is not established by the end of the simulation, a fixed arbitrary high value is assigned to the reservation establishment time. In order to obtain the results presented in this section and the rest of the paper, two GAs runs are carried out, each with a different randomly generated initial population. The final populations are combined and the simulations repeated 10 times for each solution. The average objective values are calculated and the non-dominated solutions identified and presented.

As an example, Figure 4 shows the non-dominated solutions obtained for each of the four methods of delivering RSVP control messages when the link capacity is 0.5Mb/s. Although the minimisation of delay was also carried out, the delay axis is omitted from Figure 4. Figure 4 shows the interaction between three of the objectives for an exemplar case. However, in order to explore the relationship between all four objectives, the RET, loss and delay of the non-dominated solutions are all plotted alongside the corresponding protocol overhead. Figures 5, 6 and 7 show the protocol overhead and RET, loss and delay respectively for a link capacity of 0.5Mb/s and 1Mb/s. Similar trends exist in Figures 5, 6 and 7 and a number of interesting observations may be made.

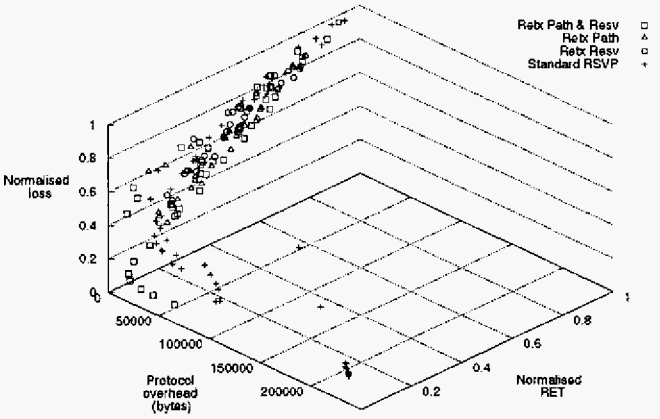

Fig. 4. Non-dominated solutions for single reservation with link capacity = 0.5Mb/s (background traffic from exponentially distributed On/Off sources).

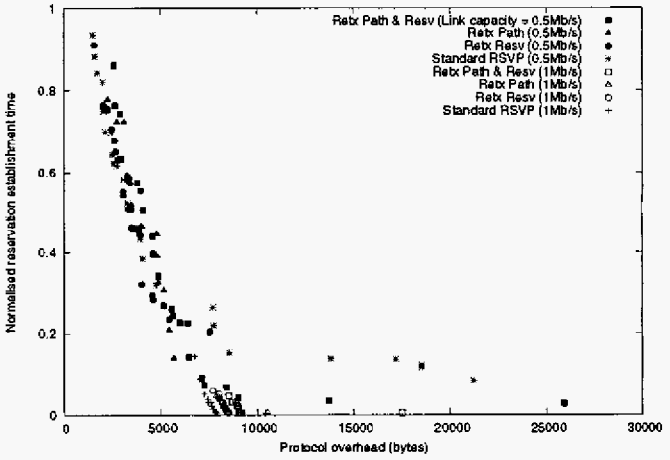

Fig. 5. Reservation establishment time and protocol overhead for non-dominated solutions (single reservation with background traffic from exponentially distributed On/Off sources).

When the link capacity is 0.5Mb/s, it can be seen that standard RSVP and retransmitting only Resv messages obtain the lowest protocol overhead, at the expense of high RET, loss and delay. This observation may be understood by realising that the increased congestion in the downstream direction (travelled by Path messages) due to the CBR flow means a significant number of Path messages would be excessively delayed or dropped. Consequently, control message delivery mechanisms that have the ability to retransmit Path messages would send more messages to combat the high delay and loss, pushing the lowest overall protocol overhead obtainable

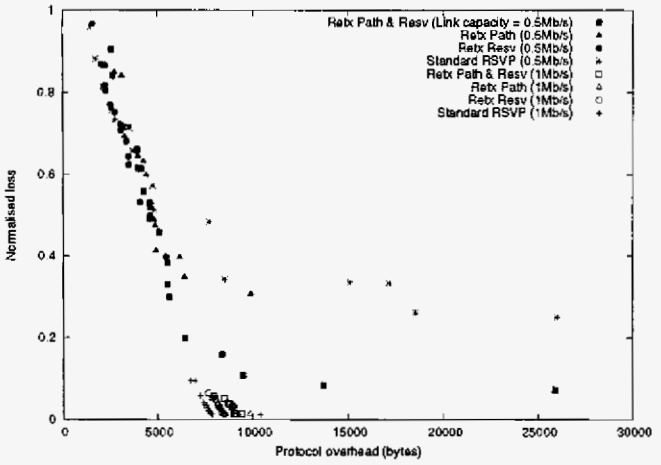

Fig. 6. Loss and protocol overhead for non-dominated solutions (single reservation with background traffic from exponentially distributed On/Off sources).

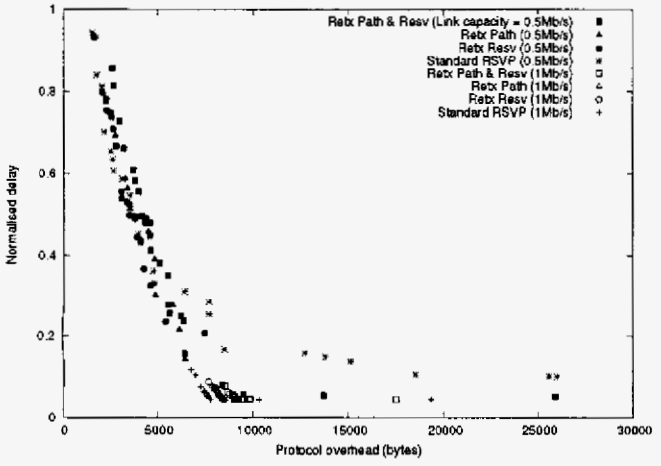

Fig. 7. Delay and protocol overhead for non-dominated solutions (single reservation with background traffic from exponentially distributed On/Off sources).

higher. Somewhat paradoxically, this inability to retransmit Path messages limits the performance of standard RSVP and the scheme that retransmits only Resv messages as the protocol overhead rises and the RET, loss and delay fall. Therefore, it can be seen that retransmitting only Path messages obtains a smaller lowest RET, loss and delay than retransmitting only Resv messages. When comparing the results for the four different implementations of RSVP, it is evident that retransmitting both Path and Resv messages appears to be the best at minimising the RET, loss and delay. However, standard RSVP does eventually obtain comparable values of these objectives but the corresponding protocol overhead is significantly higher. This high protocol overhead suggests that the refresh period must be low for standard RSVP to obtain these results whereas the ability of the retransmission algorithm to decrease the rate at which subsequent messages are sent allows it to obtain similar RET, loss and delay values with a lower overall protocol overhead.

Doubling the link capacity to 1Mb/s has a profound impact on the results. The RET, loss and delay obtained by the four RSVP variants are similar. However, interestingly, standard RSVP obtains these values with the lowest protocol overhead. The next highest protocol overhead is achieved when the retransmission of only Path or Resv messages is allowed. Retransmitting both Path and Resv messages produces the highest protocol overhead. This increasing overhead is correlated to the volume of retransmitted messages and Acknowledgment messages each scheme generates. Due to the lower network congestion, however, negligible benefit is derived from this additional overhead, unlike the case when the link capacity is 0.5Mb/s. It can be seen that the lowest protocol overhead found when the link capacity is 1Mb/s is higher than the corresponding value for a link capacity of 0.5Mb/s. This observation is due to the fact that the lower network congestion means that more Path and Resv messages will be successfully delivered to RSVP routers and so Path and Resv states are more likely to be maintained, resulting in the generation of more subsequent RSVP messages, yielding a higher lowest protocol overhead.

B. Single Reservation with Background Traffic from Pareto Distribution

In order to verify whether the findings in Section VII-A hold for a different type of background traffic, the evaluation was repeated with the background traffic obtained from Pareto On/Off sources, with a Pareto shape parameter of 1.5. The same mean On period, Off period and traffic generation rate as the exponentially-distributed On/Off sources in Section VII-A are used.

As an example, Figure 8 shows the protocol overhead and loss for the non-dominated solutions. It can be readily seen that these results are very similar to the corresponding results when the background traffic is derived from exponentially distributed On/Off sources, shown in Figure 6. This observation, allied with other results and analysis conducted when the background traffic is derived from Pareto On/Off sources (omitted for brevity), suggests that the methods used to evaluate the performance of RSVP and the findings in this work are valid, even when the background traffic is varied.

C. Multiple Reservations with Background Traffic from Exponential Distribution

RSVP may be used for signaling in an environment in which connections/reservations are established and torn down rapidly, contrasting to the static environments considered thus far. In this section, attention is paid to the performance of RSVP in such dynamic environments by considering an exemplar case in which reservations are to be made for 3 consecutive flows. Sources *A*, *B* and *C* are all identical to the CBR flow source used earlier. Source *A* sends the CBR flow to the receiver for only the first third of the simulation duration, after attempting to make a reservation for the flow. After this period, Source *A* stops sending packets and sends a PathTear message towards

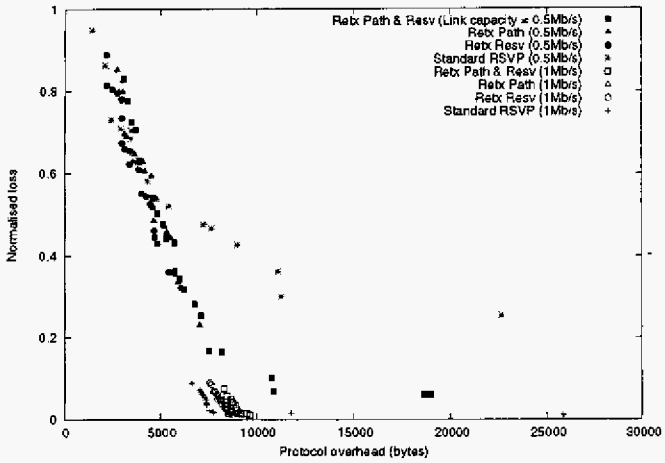

Fig. 8. Loss and protocol overhead for non-dominated solutions (single reservation with background traffic from Pareto distributed On/Off sources).

the receiver. At the start of the second third, Source *B* attempts to make a reservation for its CBR flow, which it starts sending immediately. After this interval, Source *B* stops transmitting, sending a PathTear message towards the receiver. Finally, at the start of the last third, Source *C* attempts to make a reservation for its flow and starts sending traffic to the receiver. The reservation establishment time (RET), protocol overhead, loss and delay are computed by aggregating the metrics for the three flows. The background traffic is once again derived from exponentially distributed sources connected to the intermediate routers.

When the link bandwidth is 0.5Mb/s, only one reservation can be accepted at any instant. Hence, the subsequent reservation cannot be made until the preceding reservation has been removed, either explicitly by the PathTear message or natural state timeout. The latter occurs when the PathTear messages, which are sent as best effort packets, are excessively delayed or dropped due to congestion. The motivation behind this approach is to explore the tradeoff between the requirement to avoid premature state timeout and to minimise the wasteful allocation of resources to orphaned reservations. The RET, loss and delay for the final non-dominated solutions are presented in Figures 9, 10 and 11 respectively. These figures show that when the link capacity is 1Mb/s, retransmitting only Path messages and retransmitting both Path and Resv messages obtain the best results.

Figures 9, 10 and 11 also show that, when the link capacity is 0.5Mb/s, the non-dominated solutions tend to lie in two distinct regions. Firstly, when the protocol overhead is low/moderate, the meritocratic ordering of the RSVP implementations are retransmitting both Path and Resv messages, retransmitting only Path messages, retransmitting only Resv messages and, lastly, standard RSVP. In the second region, the protocol overhead is very high and the meritocratic ordering is reversed, hence standard RSVP obtains the best result and the scheme that retransmits both Path and Resv messages the worst, in terms of the protocol overhead. The different performance in these two regions may be explained by realising that the speed with which orphaned reservations may be removed when PathTear messages are subject to delay and/or loss, depends on the refresh period and lifetime factor. Lending more emphasis to the lifetime factor than the refresh period (i.e. having a low lifetime factor and a moderate refresh period) favours schemes that retransmit unacknowledged messages since, as has been shown earlier, such schemes are better at establishing reservations quickly and preventing premature state timeout, without necessitating significant increases in the protocol overhead. Hence in the first region, retransmitting both Path and Resv messages obtains the best results. On the other hand, depending more on the refresh period to detect orphaned reservations (i.e having a low refresh period and moderate lifetime factor) favours schemes that minimise the retransmission of messages, since the low refresh period means that reservations will be established quickly anyway and premature state timeouts are unlikely, therefore retransmitting messages merely adds to the overhead, with negligible performance gains and so standard RSVP is seen to perform best.

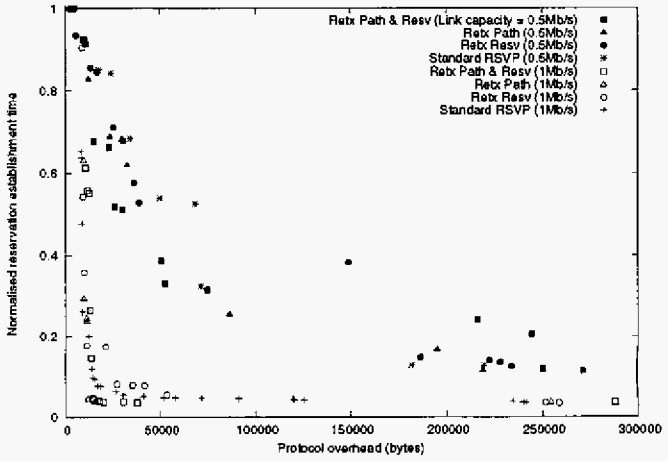

Fig. 9. Reservation establishment time and protocol overhead for non-dominated solutions (multiple reservations with background traffic from exponentially distributed On/Off sources).

VIII. Expedient Values for Timing Parameters

The populations of non-dominated solutions were analysed in order to find the most expedient values for the timing parameters for standard RSVP and RSVP featuring the retransmission algorithm. The corresponding results for the retransmission of only Path and Resv messages are omitted for brevity. The frequency of occurrence of different values of the timing parameters in the non-dominated populations was determined and are illustrated in Figures 12 to 18.

For RSVP featuring the exponential back-off retransmission algorithm applied to unacknowledged trigger Path and Resv messages,

- Figure 12 suggests that the non-dominated populations are replete with solutions with high *refresh period* values,

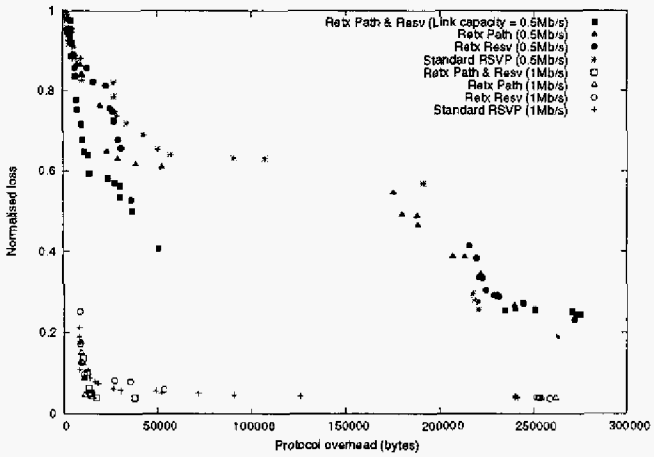

Fig. 10. Loss and protocol overhead for non-dominated solutions (multiple reservations with background traffic from exponentially distributed On/Off sources).

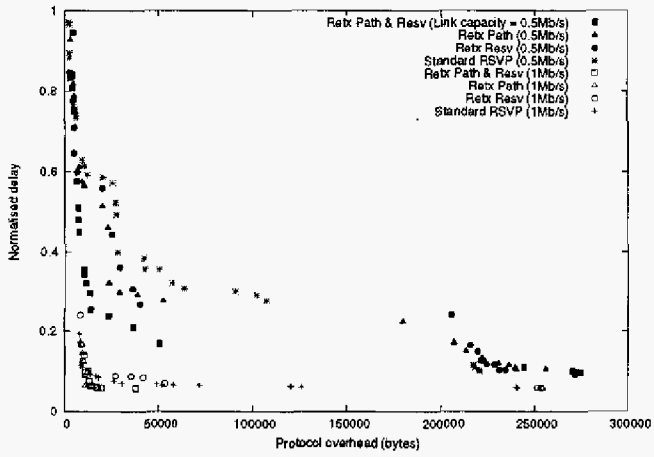

Fig. 11. Delay and protocol overhead for non-dominated solutions (multiple reservations with background traffic from exponentially distributed On/Off sources).

with the only exception being when attempting to establish multiple consecutive mutually exclusive reservations over a highly congested network, when low refresh period values occur frequently among the non-dominated solutions.

- According to Figure 13, a low *lifetime factor* should be adopted for multiple consecutive reservations in a highly congested network. A wide range of values occur among the non-dominated populations in the other scenarios considered.
- Higher values of the *retransmission interval* are preferable for a single reservation, as shown in Figure 14. When establishing multiple reservations under the lower congestion level, the retransmission interval values in the non-dominated populations occur at either end of the range of allowable values, whereas low/moderate values dominate the higher congestion case.
- Figure 15 shows that, although a range of values of the *increment value* occur among the non-dominated solutions, lower values are preferable when establishing multiple reservations and, conversely, higher values are better suited for single reservations.
- A range of values for the *retransmission limit* occur in the non-dominated solutions. However, a clear trend in Figure 16 is that low values produce many of the non-dominated solutions for a single reservation under the lower congestion regime.

For standard RSVP,

- In Figure 17, it is evident that, when establishing a single reservation in the less congested network, high values of the *refresh period* occur frequently among the non-dominated solutions. For all the other cases, however, both low and high values occur frequently.
- Figure 18 shows that a diverse set of values of the *lifetime factor* occur among the non-dominated solutions. Lower values tend to be popular when establishing multiple reservations, whereas no clear trend emerges for single reservations.

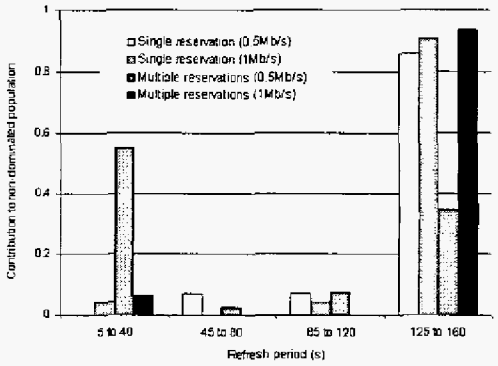

Fig. 12. Values of refresh period in non-dominated populations for retransmission algorithm.

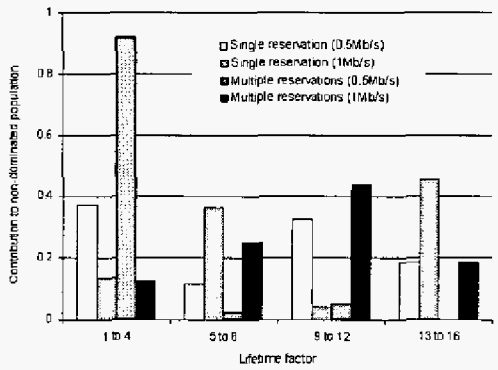

Fig. 13. Values of lifetime factor in non-dominated populations for retransmission algorithm.

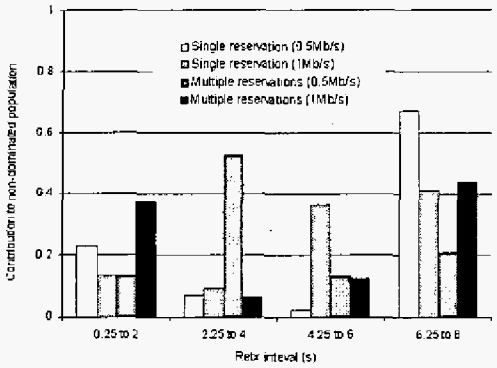

Fig. 14. Values of retransmission interval in non-dominated populations for retransmission algorithm.

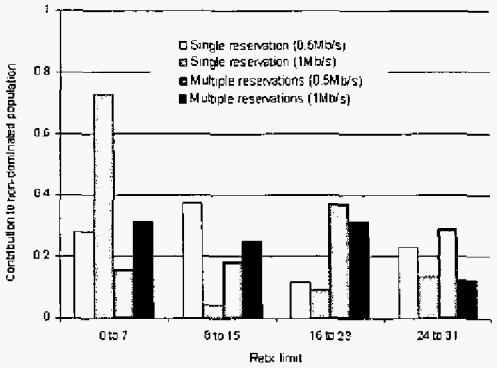

Fig. 16. Values of retransmission limit in non-dominated populations for retransmission algorithm.

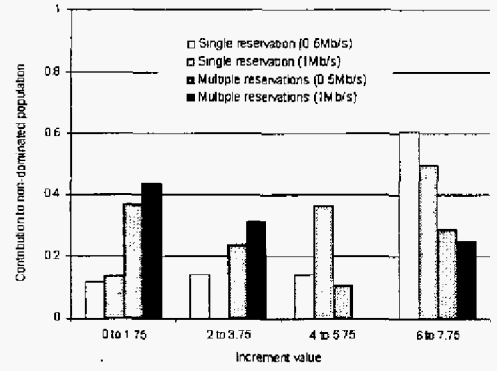

Fig. 15. Values of increment value in non-dominated populations for retransmission algorithm.

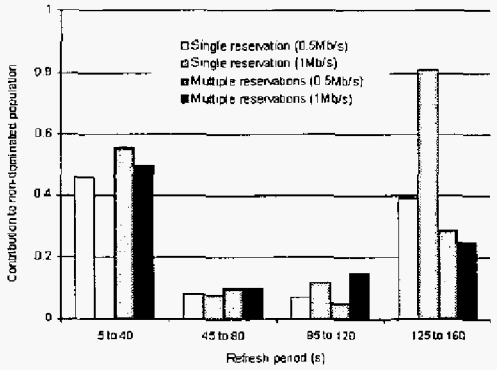

Fig. 17. Values of refresh period in non-dominated populations for standard RSVP.

IX. Conclusions and Future Work

As the proposed uses of the resource reservation protocol (RSVP) extend beyond reserving resources in IP networks to being a generic signaling protocol for generalised multi-protocol label switching (GMPLS), it is becoming increasingly important to evaluate the performance of RSVP thoroughly. In any implementation of RSVP, there are a number of discretionary timing parameters and this work has shown that, as expected, the choice of values for these timing parameters is pivotal to the performance of RSVP. Hence, the determination of the optimal values in different network conditions is worthy of significant effort and the default values, typically suggested in the corresponding standards, do not necessarily optimise performance.

A number of appropriate performance metrics were considered in this paper and the fact that, as expected, the protocol overhead conflicts with the other performance metrics (a phenomenon likely to hold for many other soft-state protocols) means that multi-objective optimisation techniques should ideally be used to evaluate the performance of RSVP. It was found that the selected multi-objective evolutionary algorithm obtained credible results, producing a population of non-dominated final solutions. Hence, the user is able to choose the solution most suited to the specific situation and priorities from this final population, rather than being restricted to a single solution, as would typically be the case were classical optimisation techniques used.

It has been shown in this paper that multi-objective optimisation techniques are a feasible and appropriate means of studying soft-state protocols, complementing the analytical models previously proposed. Hence, if RSVP is being used as a signaling protocol in a given network, provided an accurate simulation model of the network exists, multi-objective evolutionary techniques may be readily used to pre-compute the expedient values for RSVP timing parameters in a number of likely scenarios, with the corresponding values being selected when the scenario arises. However, an alternative approach will be to find the optimal timing parameters in real time. The viability of this approach is clearly contingent upon the speed and ease with which the optimisation may be conducted. Hence, investigating methods to conduct the multi-evolutionary optimisation more quickly represent an interesting area of future work. It will be also interesting to extend the approach used in this paper to conduct a more extensive

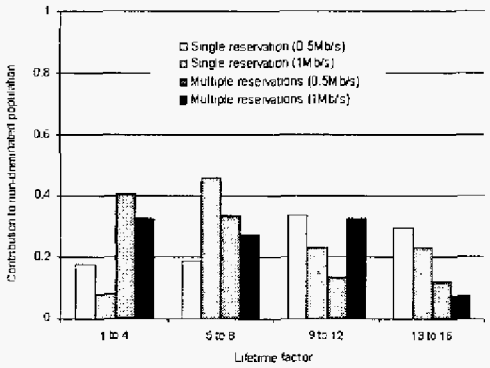

Fig. 18. Values of lifetime factor in non-dominated populations for standard RSVP.

exploration of the impact of factors such as the network topology, network parameters and background traffic on the optimal timing values and performance of RSVP. Another interesting avenue for future work is to explore how the findings presented here apply when RSVP messages are sent over a separate network/channel than the data messages. In this scenario, any degradation in the performance of RSVP will likely be caused by control plane faults and not by congestion. The impact of such faults is likely to be compounded by the likelihood that the control plane fault will also adversely affect the routing protocol, upon which RSVP is dependent. Studying the interactions between routing and signaling protocols when control plane faults occur is an interesting and important open problem.

REFERENCES

[1] L. Andersson, G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group Decision on MPLS Signaling Protocols", RFC 3468, Feb. 2003.

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels', RFC 3209, Dec. 2001.

[3] A. Banerjee, J. Drake, J. Lang, B. Turner, K. Kompella, Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements", IEEE Communications Magazine, pp. 144-150, January 2001.

[4] J. Bennett, H. Zhang, "WFQ: Worst-case Fair Weighted Fair Queueing", Proc. of IEEE Infocom, 1996.

[5] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, S. Molendini, "RSVP Refresh Overhead Reduction Extensions" RFC 2961, April 2001.

[6] L. Berger, Y. Rekhter, "Generalized MPLS Signaling - Implementation Survey", http://www.ietf.org/IESG/Implementations/, June 2002.

[7] L. Berger (Ed.), "Generalised Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions" RFC 3473, Jan. 2003.

[8] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSer-Vation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, Sept. 1997.

[9] C. Coello, "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques", Int. Journal of Knowledge & Systems, Vol.1, No. 3, pp. 269-308, August 1999.

[10] K.Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, Vol.6, No. 2, pp. 182-197, April 2002.

[11] A. Demers, S. Keshav, S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", Proc. of ACM SIGCOMM, 1989.

[12] J. Fitzpatrick, J. Grefenstette, "Genetic Algorithms in Noisy Environments", Machine Learning, Vol. 3, pp. 101-120, 1988.

[13] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, A. Malis, "Constraint-Based LSP Setup using LDP", RFC 3212, Jan. 2002.

[14] P. Ji, Z. Ge, J. Kurose, D. Towsley, "A Comparison of Hard-state and Soft-state Signaling Protocols", Proc. of ACM SIGCOMM, 2003.

[15] M. Karsten, J. Schmitt, R. Steinmetz, "Implementation and Evaluation of the KOM RSVP Engine", Proc. IEEE Infocom, 2001.

[16] O. Komolafe, J. Sventek, "An Evaluation of RSVP Control Message Delivery Mechanisms", Proc. IEEE Workshop on High Performance Switching & Routing, 2004.

[17] A. Neogi, T. Chiueh, P. Stirpe, "Performance Analysis of an RSVP-Capable Router", IEEE Network, pp.56-63, Sept./Oct. 1999.

[18] L. Mathy, D. Hutchison, S. Schmid, G. Coulson, "Improving RSVP for Better Support of Multimedia Communication", Proc. IEEE Int. Conf. on Multimedia Computing & Systems, 1999.

[19] M. Menth, R. Martin, "Performance Evaluation of the Extensions for Control Message Retransmission in RSVP", Proc. 7th IFIP/IEEE Workshop on Protocols for High Speed Networks in Lecture Notes in Computer Science 2334, pp. 35-49, Springer-Verlag, 2002.

[20] "The Network Simulator - ns-2", http://www.isi.edu/nsnam/ns, Dec. 2004.

[21] P. Pan, H. Schulzrinne, "Staged Refresh Timers for RSVP", Proc. IEEE Globecom, 1997.

[22] S. Raman, S. McCanne, "A Model, Analysis, and Protocol Framework for Soft State-based Communication", Proc. of ACM SIGCOMM, 1999.

[23] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001

[24] "The Network Simulator ns-2: Contributed Code", http://www.isi.edu/nsnam/ns/ns-contributed.html, Dec. 2004.

[25] N. Srinivas, K. Deb, "Multiobjective Function Optimization using Non-dominated Sorting in Genetic Algorithms", Evolutionary Computation, Vol.2, No. 3, pp. 221-248, 1995.

[26] P. Sharma, D.v Estrin, S. Floyd, V. Jacobson, "Scalable Timers for Soft State Protocols", Proc. IEEE Infocom 1997.

[27] L. Wang, A. Terzis, L. Zhang, "A New Proposal for RSVP Refreshes", Proc. IEEE Int. Conference on Network Protocols, 1999.

[28] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource Reservation Protocol", IEEE Network, Vol. 7, pp. 8-18, Sept. 1993.

[29] E. Zitzler, L. Thiele, K. Deb, "Comparisons of Multiobjective Evolutionary Algorithms: Empirical Results", Evolutionary Computation, Vol. 8, No. 2, pp.173-195, 2000.