Adaptability Aware Signalling Mechanisms for Large-Scale Networks

Yangcheng HUANG
Dept of Computer Science
University College London, Gower Street
London WC1E 6BT, UK

Saleem N Bhatti
Dept of Computer Science
University College London, Gower Street
London WC1E 6BT, UK

Abstract — **Existing signaling protocols and mechanisms have proven to be very well designed. We contend in this paper, however, that many properties of these systems are not yet fully understood in the context of wide deployment in large-scale network, and a number of fundamental protocols design issues require further more careful re-evaluation. In this paper, we make several observations regarding large-scale signaling with existing mechanisms. We introduce *adaptability* as an important metric and use it to evaluate the performance of two signaling mechanisms: hard state and soft state. We show the dumbness of these signaling mechanisms in large-scale networks, which would yield improvements in wide-area scalability and performance. Specifically, we develop an analytic model that allows us to quantify the relationship between state consistency and the system parameters under different scenarios. Through analysis we find that, although reducing refresh interval could help improve resilience, it is not efficient under certain circumstances; therefore the value of the refresh interval should be determined by both consistency requirements and system parameters such as internal failure ratio and link loss rate.**

Keywords-QoS;signaling; soft state; scalability; adaptability

I. INTRODUCTION

In order to support real-time multimedia communications across large-scale networks, one essential feature for the future Internet architecture is to provide scalable QoS signaling mechanisms. Existing signaling protocols, such as RSVP_TE [3], and mechanisms, have proven to be very well designed. We contend in this paper, however, that many properties of these systems are not yet fully understood in the context of wide deployment in large-scale network, and a number of fundamental protocols design issues require further more careful re-evaluation.

For example, it is commonly believed that a smaller refresh interval in soft state mechanism could speed up adaptation to changes at the expense of increased overhead. However, there is no solid proof on *how much* it could improve the consistency by reducing a certain amount of refresh interval. Given requirements on system consistency, how could we determine the value of the refresh interval in order to achieve the best balance between performance and overhead? Does the default value of time-out timer interval work well with our specific scenarios? More generally, could soft state mechanisms work efficiently against all types of failures?

Previous work on tuning signalling performance has mainly focused on three aspects: signalling overhead reduction [2] [3], reliable message delivery [6] and performance analysis [4] [5] and comparison [7]. With respect to adaptability, due to the lack of sufficiently accurate knowledge and understanding of signalling mechanisms, heterogeneity of the Internet makes adaptability analysis difficult, especially for cross-network signalling. In addition, scalability with respect to network size has been a big issue for wide deployment of QoS signalling protocols such as RSVP; therefore there is still no *live* instance of wide-deployed QoS signalling protocols in real world, which could facilitate adaptability analysis.

In this paper, we make several observations regarding QoS signaling with existing mechanisms. We introduce *adaptability* as an important metric and use it to evaluate the performance of two signaling mechanisms: hard state and soft state. Through the analysis, we contend that, most of the existing signalling mechanisms perform in dumb way, neither aware of network real-time conditions (in terms of failure rate) nor aware of network heterogeneity in message delivery, which would yield improvements in wide-area scalability and performance.

Specifically, we present a theoretical analysis on resilience of soft state signaling protocols based on probability theory. We develop analytic models that allow us to quantify the relationship between state inconsistency and the system parameters under different scenarios. Through analysis on resilience of existing soft state mechanisms we find that, in the presence of link packet loss and internal state corruption, the inconsistency ratio is not linear with the value of refresh interval; while reducing refresh interval could improve the consistency, the degree of improvements is not only determined by the value of refresh interval, but also depends on network parameters such link loss ratio and internal state corruption rate.

We also find that under certain scenarios, refresh interval has a threshold value on its effect on consistency. When the refresh interval is beyond the threshold value, its effect on consistency is very limited. From this we illustrate that it is not always efficient to improve system consistency by reducing refresh interval, unless the refresh interval is smaller than the threshold.

Moreover, the performance analysis results indicate that soft state does not perform well in case of internal state corruption; among the system parameters, internal corruption rate has a larger impact on consistency; as a result, it is necessary for routers to do extra measures to recover from internal state corruption, instead of just relying on soft state.

II. SIGNALING IN LARGE-SCALE NETWORKS

A. *Signaling Protocols*

The primary purpose of signaling is to manage state information along the data path of traffic flows; QoS signaling protocols typically have to handle additional state related to resource reservation to be made. According to their roles in signaling process, the signaling nodes could be classified into three types: signaling initiator, signaling forwarder and signaling receiver [8].

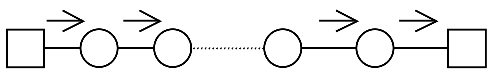


Figure 1. Generic Signaling Process

Note that, signaling forwarder here doesn't have to be a router; it can be any networked entities which are capable of message forwarding.

For a typical signaling protocol like RSVP, there are usually two types of signaling messages: trigger messages and refresh messages [6].

Signaling initiator use trigger messages to

- install new state onto signaling receiver;
- remove state from signaling receiver;
- update state changes in signaling receiver.

For example, PATH, RESV, PATHTEAR and RESVTEAR are trigger messages in RSVP.

Refresh messages contain replicated state information to update state for robustness. Signaling initiator uses refresh messages to maintain state stored in signaling receiver. Without receiving such refresh messages after a time-out interval, state would be removed from receiver. PATH and RESV messages sent after setting up RSVP sessions are refresh messages.

A generic signaling process usually includes two sub processes: trigger process, which delivers trigger messages, and refresh process, which delivers refresh messages. Although these two sub processes may share similar message format and message data, they differ significantly in their control mechanisms and objectives.

- Essentially, the trigger process is based on use of best effort traffic messages with hard state mechanism effectively; an acknowledgement is used to achieve the reliability in message delivery; state refresh is based on a soft-state mechanism.
- Overhead is the main concern of the refresh process, while reliability is the main concern for the trigger process.

Consequently, when we carry out performance evaluation of a signaling protocol, it is critical to analyze these two processes separately.

In this paper, we focus on refresh process only.

B. *Challenges*

As multimedia applications are scaled to wide-area networks, corresponding signaling protocols are supposed to work under the scenario of large-scale networks.

Here a large-scale network, like today's Internet, is characterized by the following factors: uncertainty and heterogeneity, these factors put forwards new challenges for signaling protocols.

The status of signaling participants and network connections in the path of signaling message delivery are uncertain; especially for inter-domain signaling, node and links in message delivery may be unavailable without notice; further, signaling protocols are supposed to use unreliable IP traffic [1]. Therefore, the signaling protocol should be robust against failure and other conditions that imply that the stored signaling state has to be removed.

In addition, with the wide deployment of wireless infrastructures, the increasing heterogeneity is another consideration for large-scale signaling. Different types of networks provide diverse network capacity in message delivery in terms of bandwidth and reliability; different platforms (hardware devices and software systems) need different signaling protocol suites; different multimedia applications may have different requirements on signaling performance. Therefore, performance issues of signaling protocols are heavily dependent on the scenarios and are normally a trade-off between delay, reliability, complexity and scalability. The trade-off varies in different parts of the network. For example, in radio access networks low bandwidth consumption will outweigh the low latency requirement, while in core networks it may be reverse.

C. *Scalability Requirement*

Scalability has always been an essential requirement for protocols. However, with above challenges, scalability means more important for signaling protocols.

Signaling protocols should be scalable in the amount of (state) data exchanged between signaling nodes. The amount of data in signaling traffic should be neglectable compared with that in other traffics. Also, signaling protocols should be adaptive to various scenarios, especially overhead sensitive environments such as radio access networks.

Signaling protocols should be scalable in the number of messages received by signaling communication participants (Signaling Initiating Point, Signaling Forwarder, and Signaling Sink Point) [8] and also the corresponding processing overhead on these participants. The major concern lies in the core of the network, where large numbers of messages arrive.

Signaling protocols should be scalable in the amount of signaling set-up interactions. End systems may set up several signaling sessions at the same time, and too many trigger (state set-up) messages may degrade system performance.

Basically, the performance of the signaling should degrade gracefully rather than catastrophically under overload conditions [9].

III. ADAPTABILITY AWARENESS IN SIGNALING

The essential requirement for enabling scalability is to provide signaling sessions with the ability to have direct and immediate access to network conditions and real-time network status during the signaling message delivery; furthermore, heterogeneity requires the ability to treat each signaling session *specially* and determine its parameter configurations by its own performance requirements, available resources and capabilities. The first ability could be achieved essentially by network measurements and monitoring techniques. In this paper, we focus on how to determine signaling configurations; that is, how to qualify the parameters in signaling protocols, by performance requirements, available resources and capability.

Here we define adaptability, or self-adaptability, as the follows:

Adaptability refers to protocols' capacity in determining its performance parameters according to the running network environments, and changing their behavior when the network circumstances change, in order to achieve better performance against overhead.

In the following paragraphs, we explain the dumbness of existing signaling mechanisms and explain why it is necessary to improve it.

A. Dumbness in Hard State Signaling

Basically, hard-state signaling uses best-effort state installation and removal, which is *neat* in terms of signaling. However, hard state has to rely on reliability mechanism to guarantee the delivery of the message because "state will remain installed unless explicitly removed by the receipt of a teardown message from the state-installer" [7], and therefore hard-state signaling requires reliability mechanism (1) to confirm state installation/removal operations; (2) to remove orphaned state in case that the state-installer crashes or departs without removing state [7]. For example, acknowledgement is introduced in RSVP to guarantee the delivery of trigger messages; the acknowledgement message from the receiver is required to confirm the message delivery. The node sends trigger message and then waits for acknowledgement from the receiver until retransmission timer is fired.

Such a reliable delivery process is not well designed because of the repeated re-transmission cycle. If a message were lost on the wire, the next re-transmit cycle by the network would be one soft-state refresh interval later. By default, a soft-state refresh interval is 30 seconds. Image that if one link or node between the sender and the receiver is down, all the other nodes have to process the re-transmitted trigger message endlessly. In fact, improvements such as [6] on this has been proposed and implemented in protocols; however, it just reduces the overhead, with no improvements in terms of signaling: the nodes still have to carry out the re-transmission process.

B. Dumbness in Soft State Signaling

Rather than relying on explicit (hard-state) failure detection or notification, soft state advocates the use of periodic message to refresh state in other nodes, no matter that there is error or not in the nodes. Unlike hard state, which requires a separate fault recovery protocol, soft state uses the same refresh mechanism to recover from failure. Nodes listening to the refreshing in silent mode verify state consistency and recover from inconsistency after receiving the refresh message [6].

While soft state is robust and resilient, it does not share the *adaptability* property, which facilitates scalable wide-area operation. A naive application of soft state can substantially increase bandwidth utilization and lead to non-scalable operation. We explain its dumbness from the following aspects.

1) Standard soft state mechanism, while simple in implementation, increases complexity in configuration and maintenance.

Timer interval in soft-state signalling has to be set manually by administrator. The value is mainly determined based on recommendations of original protocol designers or past experience. Usually there is no careful calculation or solid theoretical proof for the configuration. For example, in RSVP, each RSVP-enabled node chooses the refresh time locally. By default, the value of refresh timer is set to be 30 seconds, as suggested in [10]. However, such a value is not suitable for QoS signalling in wireless networks (which is *usually* set to be 5 seconds).

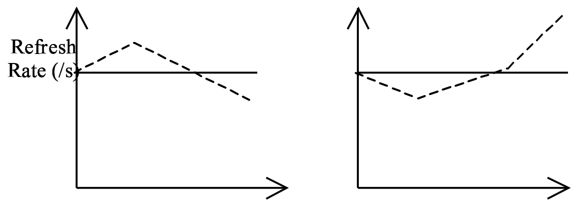
Soft state is usually per-session based. It is common that several timers run simultaneously in one process. Different applications or protocols may have correlations and conflicts in timer configuration. For example, the timer interval of a Srefresh (summary refresh) message should be longer than that of standard refresh message; when a standard refresh message is sent, a corresponding summary refresh should not be sent during the same refresh period [2].

As a result, in implementation, a lot of effort has to be spent on per-session timer maintenance, message retransmission (e.g., avoid message bursts) and message sequencing.

2) With current soft state scheme, it is hard to balance signalling performance and cost and achieve certain QoS guarantee.

A smaller refresh interval in RSVP speeds up adaptation to changes, while the communication overhead can be excessive; if the refresh interval is large, it will take longer to detect and recover state corruption [2] and message loss, which may not guarantee the quality of traffic delivery service.

With existing signalling mechanisms, it is always a trade-off between performance and overhead. It is true but general. One prerequisite is missing from above observation: scenarios. It is obvious that with the same consistency requirements, signalling overhead in fixed network could be much lower than that in mobile network. Accordingly, given fixed requirements on overall overhead, it is reasonable to allow more overheads (when messages are delivered) in wireless networks than in wired networks in order to react to message loss (in wireless networks) more quickly and improve overall performance when signalling messages traverse across heterogeneous network. Similarly, it is reasonable to speed up refresh paces only when failures are detected and lower down refresh paces when there is no failure, rather than keeping a uniform refresh pace. As shown intuitively in Figure 2, dashed line donates a dynamic refresh timer interval mechanism, while the other represents a fix refresh timer interval mechanism. The signalling refresh rate is on y-axis.



(a) Dynamic refresh rate adaptive to network environments (from wireless to wired network)

(b) Dynamic refresh rate adaptive to real-time network conditions

Figure 2. Overhead Distribution

C. Existing Methods in Tuning Soft State Performance

There have been several proposed methods in reducing refresh overheads in RSVP extensions [2][3].

RFC3209 [3] defines a new Hello message (for rapid node failure detection). HELLO extension facilitate node failure detection by exchanging heartbeat refresh message between neighboring nodes, instead of an end-to-end approach, which could localize refresh traffic and adapt to local network conditions. The RSVP Hello extension enables RSVP nodes to detect when a neighboring node is not reachable.

RFC2961 [2] describes mechanisms to reduce processing overhead requirements of refresh messages, eliminate the state synchronization latency incurred when an RSVP message is lost, and refresh state without the transmission of whole refresh messages. Three new RSVP message types are defined: 1) *Bundle messages* consist of a bundle header followed by a body consisting one or more standard RSVP messages. Bundle messages help in scaling RSVP to reduce processing overhead and bandwidth consumption. 2) ACK messages carry one or more MESSAGE_ID_ACK or MESSAGE_ID_NACK objects. ACK messages are sent between neighboring RSVP nodes to detect message loss and to support reliable RSVP message delivery on a per-hop basis. 3) Srefresh messages carry one or more MESSAGE_ID LIST, MESSAGE_ID SRC_LIST, and MESSAGE_ID MCAST_LIST objects. They correspond to Path and Resv messages that establish the states. Srefresh messages are used to refresh RSVP states without transmitting standard Path or Resv messages.

Paper [6] introduces a staged refresh timer mechanism to reduce trigger message re-transmission, which has been also defined as a RSVP extension in [2]. The staged refresh timer mechanism retransmits RSVP messages until the receiving node acknowledges. It can address the reliability problem in RSVP.

D. A Stateless Adaptive Signaling Framework

Usually feedback mechanism is utilized to achieve adaptation to changes.

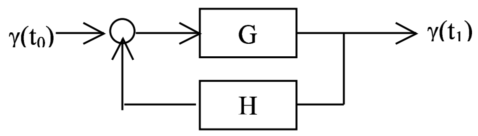


Figure 3. A Typical Feedback Loop

In this paper, we utilize a stateless mechanism to achieve adaptability in soft state mechanism. In our design refresh timer interval γ is determined by network parameters (failure rate τ) and requirements on consistency δ.

$$\gamma = \Phi(\tau,\delta) = \gamma_0 - f(\delta) - \theta(\tau) \qquad (1)$$

(γ_0 donates the default value of refresh interval defined by signaling protocols)

Obviously, with the increase of failure rate τ and consistency requirement δ, refresh interval γ should be reduced.

Such an approach is *stateless* since there is no per-session state in calculating the refresh timer interval. Therefore, the next step toward adaptive refresh signaling is to quantify the relationship between consistency and related factors and calculate the function f and θ in equation 1.

IV. Probabilistic Resilience Model

In this section, we present a theoretical analysis on resilience of soft state signaling protocols based on probability theory. We develop an analytic model that allows us to quantify the relationship between state inconsistency and the system parameters under different scenarios. Further analysis is given in order to present a quantitative understanding on the resilience performance of soft state mechanism.

A. Failure Events

We categorize the failure events into two types: node-level failure and link-level failure. In terms of signaling process, node-level failure, such as node restarting or leaving, causes state loss, which is referred as "internal state corruption" in RSVP_TE [2], while link-loss failure means message loss by transport layer or link layer. Soft state is used to recover from these two types of failure. Internal state corruption could be detected and recovered by state in refresh message, and lost signaling message would be re-transmitted by periodic message refresh process.

B. *Metrics & Assumptions*

Assume refresh messages are sent with period (refresh timer interval) *r*.

Two important factors on resilience are *link loss ratio* p and *internal failure rate* λ. Link loss ratio describes the loss possibility of a message during the delivery, while internal failure rate represents the occurrence frequency of internal state corruption. These two network characteristics could be measured with many existing techniques.

The *failure recovery time* (FRT) in one refresh period *r* is defined as the time from the occurrence of first failure until the end of the period. Note that, the failure recovery time is also the state inconsistency time for the system.

The *inconsistency ratio* of one period *r* is defined as the fraction of inconsistency time in the period.

We calculate the communication overhead as follows.

$$\gamma * \theta \quad (2)$$

(γ donates refresh interval, while θ donates average refresh message length)

In order to simplify our model, we assume all messages' packet length is identical, without losing generality. Therefore, we analyze the performance on overhead only through the value of refresh interval.

Assume that state corruption events occur according to a Poisson process of rate λ (λ>0); this assumption has been made by statistical methods for reliability theory [11]; this model comes about when the inter-arrival times between failures are independent and identically distributed according to the exponential distribution, with parameter λ (λ>0).

C. *Probabilic Soft State Model Without Channel Loss*

Consider an arbitrary period, starting at t_0. Let X be the time of first failure occurrence after t_0.

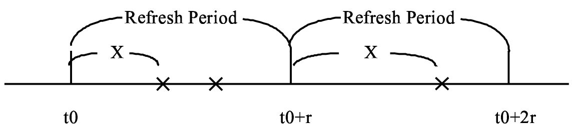


Figure 4. Soft State Model Without Channel Loss

According to above definition,

$$FRT= (t0+r-X)^{+} \quad (3)$$

Also, according to the assumption,

$$X-t_0 \sim Exp(\lambda) \quad (4)$$

Therefore, the expected failure recovery time is

$$E(t0+r-X)^{+}$$

$$= E(r-\gamma)^{+} \text{ (where } \gamma = X-t_0 \sim Exp(\lambda))$$

$$= \int_0^{\infty} (r-\gamma)^{+} \lambda e^{-\lambda\gamma} d\gamma = \int_0^{r} (r-\gamma)\, \lambda e^{-\lambda\gamma} d\gamma$$

$$= r + \frac{-1 + e^{-r\lambda}}{\lambda} = \varphi(r) \quad (5)$$

The expected inconsistency ratio without channel loss is a

$$1 - \frac{1 - e^{-\lambda r}}{\lambda r} \quad (6)$$

D. *Probabilic Soft State Model With Channel Loss*

Let Y be the time of first failure occurrence after last state refresh.

$$Y \sim EXP(\lambda) \quad (7)$$

For a refresh interval with length S, the expected failure recovery time (or expected time under inconsistent state) is $E(S-Y)^{+} = g(s)$; among *n* refresh intervals, the total time spent under inconsistent state is n g(s);

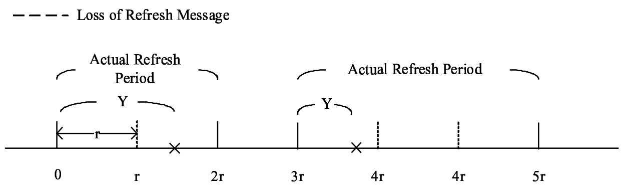


Figure 5. Soft State Model With Channel Loss

Let *p* be the channel loss rate. With channel loss, the length of the refresh interval observed at one node can be r, *2r … k r*, subject to certain possibility. Let S be the length of a typical refresh interval. S is a random variable.

$$P(S=r) = 1-p$$

$$P(S=2r) = p\,(1-p)$$

$$\dots$$

$$P(S=k\,r) = p^{k-1}(1-p) \quad (8)$$

According to the Geometric distribution density function,

$$S \sim r\ Geom(1-p) \quad (9)$$

Therefore,

$$E(S) = r/(1-p)$$

$$E(S-Y)^{+} = E(\varphi(S)) = \sum (\varphi(k\,r)\, p^{k-1}(1-p))$$

$$= \varphi(r)(1-p) + \varphi(2r)\, p\,(1-p) + \ldots + \varphi(k\,r)\, p^{k-1}(1-p) + \ldots$$

$$= \frac{r}{1-p} - \frac{e^{r\lambda}-1}{e^{r\lambda}-p}\,\frac{1}{\lambda} \qquad (10)$$

Then the expected inconsistency ratio is

$$(n\, g(s))/(n\, E(S)) = g(s)/E(S) \qquad (11)$$

Therefore, the expected inconsistency ratio with channel loss is

$$\frac{\frac{r}{1-p} - \frac{e^{r\lambda}-1}{e^{r\lambda}-p}\,\frac{1}{\lambda}}{\frac{r}{1-p}} = 1 - \frac{e^{r\lambda}-1}{e^{r\lambda}-p}\,\frac{1-p}{\lambda r} \qquad (12)$$

Given the consistency ratio δ, link loss ratio p and node churn rate λ, we could get the value of refresh interval easily with above equation.

Figure 6 presents a 3d view of soft state performance on resilience.

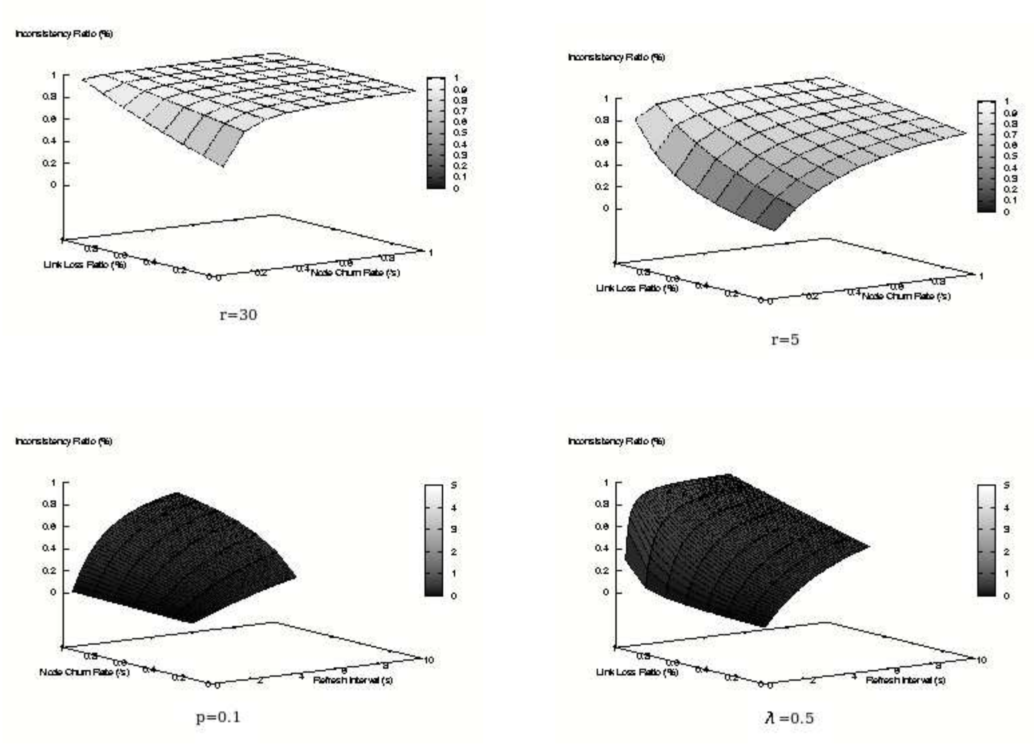


Figure 6. Performance of Soft State in the Presence of Failure (A 3D View)

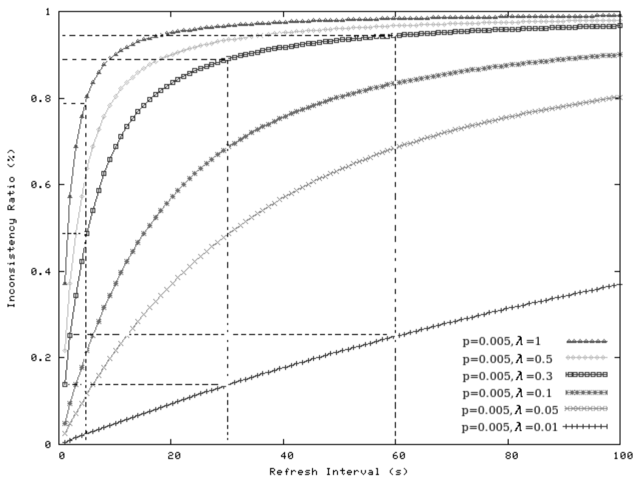


Figure 7. Consistency vs. Refresh

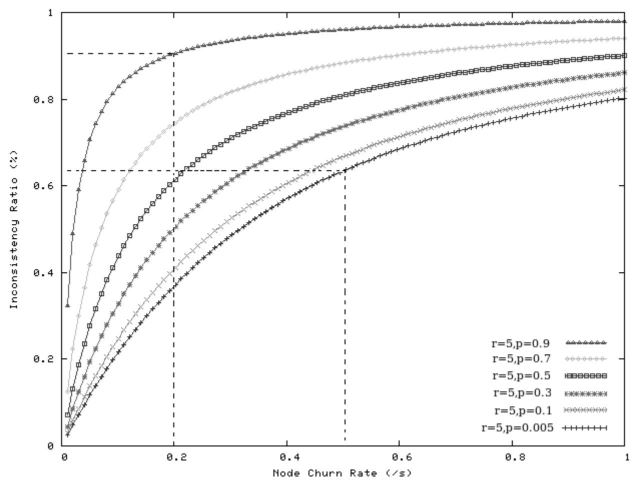


Figure 8. Consistency vs. Node Churn Rate

V. ANALYSIS

A. Scenario

Proactive wireless routing protocols such as OLSR [12] employ periodic exchange of messages to maintain topology information of the network at each node. In OLSR, each node periodically broadcasts HELLO messages, Topology Control messages to maintain network topology information, MID (Multiple Interface Declaration) and HNA (Host Network Association) messages. The default Hello timer interval is set to be 2 seconds in [13], while all the other timer interval is set to be 5 seconds.

To simulate the node dynamics, we assume poison processes with rates λ be used to generate node join and failure events. We take such *node churn rate* as internal failure rate in proposed resilience model. Without losing generality, the value of λ is set to be between 0 and 5. In addition, the *link loss rate* (probability) is between 0 and 100%.

B. Performance Analysis

1) Consistency vs. Refresh Interval

The configuration for parameters λ and p is as TABLE I.

TABLE I. PARAMETERS (λ, P)

Node Churn Rate λ	Link Loss Ratio p
0.01	0.005
0.05	0.005
0.1	0.005
0.5	0.005
1	0.005

From Figure 7 we can see,

When node churn rate is small (λ=0.01), inconsistency ratio is almost linear with refresh interval; under such a scenario, reducing refresh interval could improve consistency.

When nodes join or leave frequently (λ=0.5 or 1), refresh interval has a threshold value R on its impact on consistency. When refresh interval r is smaller than R, the inconsistency ratio increases very fast when increasing refresh interval. When refresh interval r is larger than R, the inconsistency ratio gets steady when increasing refresh interval. Under such a scenario, reducing refresh timer interval would has very limited improvements on consistency when it is larger than the threshold; reducing refresh timer interval would improve the consistency dramatically when the refresh interval value is between 0 and the threshold.

For example, when the refresh interval decreases from 60s to 30s, there is only around 5% improvement on consistency; however, when the refresh interval decreases from 5s to 2s, the inconsistency ratio drops from 80% to 40%.

In our case, the value R is around 30s.

When node churn rate is between 0.01 and 0.5, the inconsistency ratio increases moderately when increasing refresh interval.

2) Consistency vs. Node Churn Rate

The configuration for parameters r and p is as TABLE II.

TABLE II. PARAMETERS (P, R)

Link Loss Ratio p	Refresh Timer Interval r
0.005	5
0.1	5
0.3	5
0.5	5
0.7	5
0.9	5

From Figure 8 we can see,

With the increase of node churn rate, the inconsistency ratio increases rapidly; a small increase of node churn rate would cause a large decrease in system consistency, even when the link loss ratio is very low. For example, with and 0.5% loss of refresh messages, when churn rate λ increases from 0.2 (one node churn event per 5 seconds) to 0.5 (one node churn event per 2 seconds), the consistency suffers from nearly 30% decrease.

In case of internal state corruption, performance of existing soft state mechanism is not satisfactory. For example, when there is one node joining or leaving event per second on average, the inconsistency ratio would be above 70%. From this we conclude that, under existing soft state mechanism, internal failure ratio (e.g. node churn rate) has a large effect on consistency.

3) Consistency vs. Link Loss Ratio

The configuration for parameters r and p is as TABLE III.

TABLE III. PARAMETERS (λ, R)

Node Churn Rate λ	Refresh Timer Interval r
0.01	5
0.05	5
0.1	5
0.3	5
0.5	5

From Figure 9, we could see that,

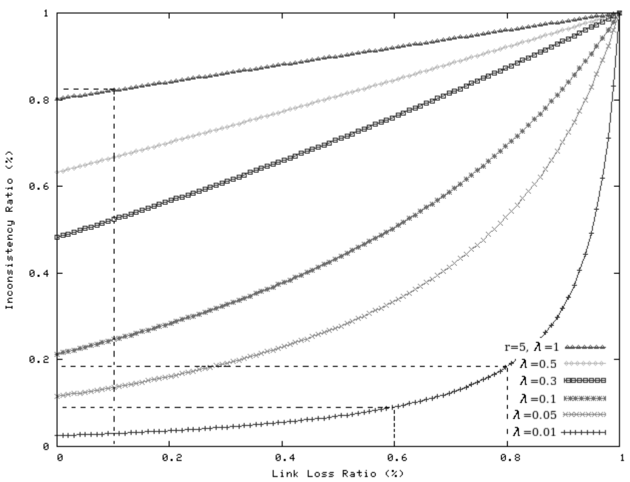


Figure 9. Consistency vs. Link Loss Ratio

When node churn rate is small (λ=0.01), increase of link loss ratio doesn't have an obvious effect on consistency. For example, even with 80% message loss in delivery, the inconsistency ratio is still less than 20%.

When node churn rate is larger (λ>=0.3), the link loss rate is almost linear with inconsistency ratio. The slope decreases with the increase of node churn rate. For instance, the slope decreases from 0.5 to 0.2 when node churn rate increases from 0.3 to 1.

C. Results

From above analysis, we could get the following implications.

- Reducing refresh timer interval could improve the consistency *more or less*; the degree of the improvement depends not only on the value of refresh interval, but also on system parameters such as link loss ratio and node churn rate.
- Although increasing link loss ratio has the similar effect with increasing node churn rate, but not with the same degree. By comparing Figure 8 and Figure 9 we can see, the node churn rate has a larger impact on consistency than link loss ratio.
- From above discussion, soft state performs well in terms of message loss in delivery; however its performance degrades largely with a small node churn rate. Existing soft state is resilient in recovering inconsistency when node churn rate and link loss ratio are both low; how, its performance is not satisfactory with a moderate node churn rate. This explains why it is important for routers to do state verification by itself, not just relying on refresh messages to recover from state loss.

VI. RELATED WORK

In addition to research effects (on tuning signaling performance) listed in section III, we give a very brief overview on model based signaling performance analysis as follows.

Komolafe [5] frames the interactions between RSVP timer interval and performance metrics as a multi-objective optimisation problem, "which, due to its intractable nature, is tackled using a reputable multi-objective evolutionary algorithm". This research also compares RSVP extensions in terms of multiple performance metrics under difference network conditions, including standard RSVP, acknowledgement based retransmission algorithm and staged timer mechanism and RSVP_TE HELLO extensions.

Ping [7] uses a continuous time Markov model to quantify state inconsistency and cost in single and multi-hop signaling scenarios, to compare and contrast a variety of signaling approaches. The study shows that a simple soft state approach does not compete with a mixed hard/soft state approach in performance: "a soft-state approach coupled with explicit removal substantially improves the degree of state consistency while introducing little additional signalling message overhead".

Raman [14] develops an open-loop multi-class queuing model for soft-state based communication to analyze the consistency behavior and bandwidth consumption behavior given different data arrival rates, loss rates and session expiration rates. The transmission channel between sender and receiver acts as a "service"; consistent state and inconsistent state are inputs of the queuing system.

Fu [16] presents models on soft state mechanism with formal modeling methods SDL.

VII. SUMMARY AND CONCLUSIONS

One essential problem of existing signaling protocols is scalability. In this paper, we introduce *adaptability* as an important metric for large-scale deployment of signaling protocols and use it to evaluate the performance of signaling mechanisms. We find that, however, existing signaling mechanisms, namely soft state and hard state, show the dumbness in disseminating messages, which would yield improvements in wide-area scalability and performance. Overall, we believe that adaptability could help improve scalability.

Based on such a principle, we propose a simple stateless adaptive signaling framework, which could determine refresh interval by existing network parameters (such as failure rate) and requirements on consistency. We believe such a mechanism is more straightforward and simple than feed-back based adaptability mechanism.

Furthermore, we develop an analytic model that allows us to quantify the relationship between state consistency and parameters including refresh interval, link loss ratio and internal failure rate. Through analysis we find that, although reducing refresh interval could help improve resilience, it is not efficient under certain scenarios; also, existing soft state mechanism doesn't perform satisfactorily against internal state corruption, and extra failure detection measures such as router's self verification are quite necessary

VIII. ACKNOWLEDGEMENTS

We are grateful to members of the Networks Research Group at UCL for valuable comments.

REFERENCES

[1] D. Clark: The design philosophy of the DARPA Internet protocols, Symposium proceedings on Communications architectures and protocols, p.106-114, Stanford, USA, August 16-18, 1988.

[2] L. Berger, et al. RSVP Refresh Overhead Reduction Extensions. RFC 2961, April 2001.

[3] D. Awduche, et al. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209, Dec 2001.

[4] O. Komolafe, J. Sventek. An Evaluation of RSVP Control Message Delivery Mechanisms. Proc. IEEE HPSR, Phoenix, Arizona, USA, April 2004.

[5] Komolafe, O. Sventek,J.S. RSVP Performance Optimization using Multi-Objective Evolutionary Optimization, Proceedings of IEEE Infocom 2005.

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

[7] Ping Ji, Zihui Ge, Jim Kurose, Don Towsley. A Comparison of Hard-state and Soft-state Signaling Protocols. Proc. ACM SIGCOMM, 2003.

[8] Y Huang, S Bhatti. Scaleable Signalling Underlay for Overlay Networks, Proc. IEEE International Conference on Networking (ICON), 16-19 November 2004.

[9] M. Brunner, Requirements for Signalling Protocols, RFC 3726, April 2004.

[10] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin. Resource ReserVation Protocol -- Version 1 Functional Specification", RFC 2205, September 1997.

[11] Steven E. Rigdon, Asit P. Basu, Statistical Methods for the Reliability of Repairable Systems, John Wiley & Sons, May 2000.

[12] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, a. Qayyum et L. Viennot, Optimized Link State Routing Protocol, IEEE INMIC Pakistan 2001.

[13] Engelstad, P.E., Tonnesen, A., Hafslund, A., Egeland, G. Internet Connectivity for Multi-Homed Proactive Ad Hoc Networks, Proc. IEEE International Conference on Communication (ICC'2004), Paris, June 20-24, 2004.

[14] Raman, S., Mccanne, S: A Model, Analysis, and Protocol Framework for Soft State-based Communication. Proc. SIGCOMM 1999, Cambridge, MA, Sept. 1999

[15] Baskett, F., Chandy, M., Muntz, R., and Palacios, F. Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. Journal of the Association for Computing Machinery 22, 2 (1975), 248-260

[16] Xiaoming Fu, Dieter Hogrefe. Modelling Soft State Protocols with SDL. NETWORKING 2005, p289-302.

[17] Hancock, R., Karagiannis, G., Loughney, J., and S. van den Bosch. Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005.

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