Performance Enhancement of On-Board Communication Networks Using Outage Prediction

Adeel Baig, *Student Member, IEEE*, Lavy Libman, *Member, IEEE*, and Mahbub Hassan, *Senior Member, IEEE*

***Abstract*—A research area that has become increasingly important in recent years is that of on-board mobile communication, where users on a vehicle are connected to a local network that attaches to the Internet via a mobile router and a wireless link. In this architecture, link disruptions (e.g., due to signal degradation) may have an immediate impact on a potentially large number of connections. We argue that the advance knowledge of public transport routes, and their repetitive nature, allows a certain degree of *prediction* of impending link disruptions, which can be used to offset their catastrophic impact. Focusing on the transmission control protocol (TCP) and its extension known as *Freeze-TCP*, we present a detailed analysis of the performance improvement of TCP connections in the presence of disruption prediction. In particular, we propose a Markov model of Freeze-TCP that captures both the TCP behavior and the prediction+"freezing" feature and, using simulations, show that it accurately predicts the performance of the protocol. Our results demonstrate the significant throughput improvement that can be gained by disruption prediction, even with random packet losses or imperfect timing of the predicted disruptions.**

***Index Terms*—Disruption prediction, freeze-transmission control protocol (TCP), mobile networks, mobile router (MR), performance-enhancing proxy (PEP), public transport, wireless TCP.**

I. INTRODUCTION

RECENT YEARS have evidenced a growing interest in the paradigm of pervasive connectivity, which aims to provide communication and information access anywhere, anytime. As part of that trend, there has been an increasing number of both commercial systems and research projects aiming to provide broadband Internet services to public transport passengers, by deploying high-speed local-area networks (LANs) on-board public transport vehicles. Some examples of commercial systems include [2]–[4], which provide Internet services for land-based transportation systems (i.e., long-distance trains and buses), and [5] and [6], which do the same for air travellers. Research projects, such as [7] and [8] study implementation, integration, deployment, and standardization of IPv6 mobility-related technologies.

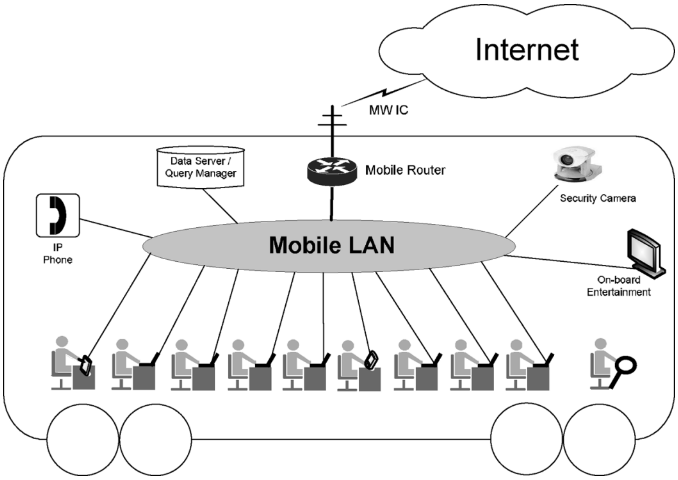


Fig. 1. On-board communication architecture.

The architecture of a typical on-board mobile network is depicted in Fig. 1. It consists of three main components: a high-speed mobile local-area network (MLAN), mobile router (MR), and mobile wireless Internet connection (MWIC) [9]. The MLAN provides a local high-speed connectivity to the on-board passengers (and other devices integrated in the vehicle). The MR facilitates communication between the MLAN and the global communication infrastructure (Internet). It may also be equipped with additional features, such as a data server and query manager to enable proxying/caching of on-board user requests. Finally, the MWIC connects the gateway to the outside world, typically by either a land-based cellular station (in urban areas) or a satellite (in rural areas or in sea- or airborne systems). Thus, passengers need simply connect their devices to the MLAN and start enjoying Internet services. This architecture is advocated by the Internet Engineering Task Force (IETF) network mobility (NEMO) working group, and a basic support protocol, based on an extension of Mobile IPv6 for mobile networks, is currently a proposed standard [10].

The mobile network architecture has several significant advantages over the individual connection approach that is still widely used today, i.e., where each user uses their own private connection to the Internet. First and foremost, it enables the individual passengers to enjoy multiple wireless interfaces. For example, when a long-distance train leaves the coverage area of a metropolitan cellular network, communications can continue to be maintained via satellite, without requiring each passenger to carry their own satellite antenna; furthermore, these

---

Manuscript received May 31, 2005; revised January 30, 2006. National ICT Australia is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the Australian Research Council. This paper was presented in part at the IEEE Global Communications Conference, Dallas TX, November 2004.

A. Baig and M. Hassan are with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia, and also with National ICT Australia, Eveleigh, NSW 1430, Australia (e-mail: abaig@cse.unsw.edu.au; mahbub@cse.unsw.edu.au).

L. Libman is with National ICT Australia, Eveleigh NSW 1430, Australia (e-mail: Lavy.Libman@nicta.com.au).

Digital Object Identifier 10.1109/JSAC.2006.875108

interface transitions can be transparent to the users, who "feel" that they are simply connected to a LAN. In addition, having the users connected via a common local network brings advantages in terms of performance as well. In particular, statistical multiplexing of the MWIC becomes possible when the users' data traffic patterns are bursty; also, popular items (such as news or travel information) that are requested by many users can be cached by the MR on board, eliminating the need to retrieve them several times via the MWIC (in fact, on-board multicasting may even be possible when several users request the same information simultaneously, e.g., tune to the same Internet radio station). Finally, the fact that the entire MLAN is seen as a single mobile entity alleviates many scalability-related issues; for instance, it helps prevent the common problem of disconnection when a large number of users cross a cell boundary simultaneously.

Despite the many advantages listed above, a critical issue with the mobile network architecture is the possibility of temporary outages in the MWIC. An *outage*[1] is a period during which all packets to/from the MR are lost or corrupted, and can be caused by various factors, such as loss of signal (e.g., when the vehicle passes through a tunnel or high-rise building blocking the line-of-sight), interference, or handoff failure between neighboring base stations [11]. The problem is exacerbated in a mobile network setting, where a potentially large number of users are connected to the Internet via a single wireless link, hence, its disruption affects all the active connections and services. Recovering from such unexpected disruptions can be slow and costly. On the flip side, the nature of public transport is such that its routes are known in advance and repetitive. This important feature can be used to *predict* link outages to a certain extent. In particular, many of the typical scenarios that cause wireless link outages, such as going under a tunnel or passing near a high-rise building blocking the line-of-sight, are almost exclusively location-dependent by nature, and therefore occur predictably whenever the vehicle returns to the same location.

It is important to emphasize that, once predicted, not much can usually be done about the wireless link outage itself (unless the MR is equipped with multiple wireless interfaces, in which case it may be able to negotiate a vertical handoff for the affected sessions). Nevertheless, by sending an advance notification to existing connection endpoints before the link goes down (and when it comes up again), the network can avoid the chaotic and uncoordinated attempts by the individual connections (whether at the transport or application layer) to resume their data flows, which would otherwise be subject to contention and excessively long timeouts, and might even leave some applications in an inconsistent state. Ultimately, this leads to a better system performance.

In this paper, we focus particularly on transmission control protocol (TCP), the ubiquitous transport control protocol of the Internet, and study in detail how outage prediction can be harnessed to improve its performance, by using an extension known as *Freeze-TCP*, originally proposed to prevent undesirable timeouts when a mobile host is handed off between wireless base stations [12]. We are not concerned with the details of the disruption prediction mechanism; rather, we merely assume that outages can be predicted sufficiently in advance with a certain probability. Our contribution is twofold. First, we describe a Markovian model for the behavior of Freeze-TCP sources in the presence of partially predicted link outages. This model is an extension of the one described in [1], in that it accounts for random packet losses that occur in addition to complete outages, and for imperfect timing of the predictions. We then apply this model to explore the dependence of the Freeze-TCP throughput on such parameters as the frequency and duration of outages, their prediction probability, random packet loss rate (in addition to the outages), and timing of the prediction. Our results (which are also backed by *ns-2* simulations) demonstrate that, for a given pattern of outages, there is a monotonic dependence between the average throughput attained by Freeze-TCP and the prediction probability, which becomes more convex as the maximum TCP window size increases and/or the link becomes more unstable. Furthermore, our findings show that the throughput improvement attained by Freeze-TCP and outage prediction remains significant as long as the random packet loss rate is below about 1%.

The rest of the paper is organized as follows. Section II presents some of the related work on wireless TCP enhancements and mobility prediction, and discusses particular issues arising for on-board mobile networks. Section III describes the Markov model of Freeze-TCP, as well as the implementation considerations of Freeze-TCP in the *ns-2* simulator. Section IV presents the results of our experiments, obtained by applying the model and simulation runs in several typical scenarios. Finally, concluding remarks are presented in Section V.

II. RELATED WORK

A. *TCP Enhancements for Wireless Environments*

It is generally accepted that, in its classic form, TCP is ill-suited for wireless communications: it takes any packet loss to be an indication of congestion (a reasonable assumption for the majority of fixed communication links), whereas losses in wireless links are mainly caused by random bit errors, e.g., due to signal fading. With the growing acceptance of wireless and mobile computing, the study of techniques to improve TCP performance in wireless environments has become an active area of research [13], [14]. Some of the recent proposals to that end include Snoop [15], where the base station can make local retransmissions to the mobile host; Indirect TCP [16], which splits the communication into two separate TCP connections, between the mobile host to its base station and from there to the remote (fixed) endpoint; M-TCP [17], where the base station "chokes" the remote sender by advertising a small window size; and WTCP [18], which uses rate-based (rather than window-based) flow control altogether. Chakravorty *et al.* [19] go even further and propose to avoid slow-start and congestion avoidance altogether, by clamping the TCP window to an static estimate of the bandwidth-delay product of the link.

All of the above methods, however, require significant support from the wireless base station or the sending host, and are therefore unsuitable for our on-board mobile networking scenario. By contrast, Freeze-TCP, proposed in [12] to prevent

---

[1] In this paper, we use the terms *outage* and *disruption* interchangeably.

undesirable timeouts when a mobile host is handed off to a new wireless base station, is a fully end-to-end approach and therefore well-suited to our context. In Freeze-TCP, once notified of the impending handoff by the MAC layer, the mobile host transmits an acknowledgment carrying a zero window advertisement (ZWA). Upon receiving the ZWA, the remote (fixed) host freezes all data transmission and enters a zero window probing mode, during which it attempts sending one-byte probes at intervals specified by its persistence timer, until the mobile host is reconnected. In addition, once the mobile host is reconnected at the new base station, it transmits a triple acknowledgment (TR-ACK) with a nonzero window advertisement (NZWA), enabling the remote host to resume its data transmission.[2]

It is important to observe that, in an on-board mobile network setting, Freeze-TCP can be implemented transparently by the MR (thus keeping the mobility of the network entirely hidden from the users), by keeping track of the active TCP connections and their current sequence numbers, and "injecting" the ZWA and NZWA packets on behalf of the end hosts. Thus, the MR acts as a performance-enhancing proxy (PEP) [20], and the performance enhancement is achieved without any changes in existing user devices. An alternative approach would be for the MR to merely broadcast a signal notifying all on-board users about the imminent outage, and another signal after the outage is over, and leave the specific transport-layer protocol mechanics to the discretion of the end hosts. This alternative avoids the need to track all active connections at the MR and interfere with their end-to-end semantics, but requires the individual on-board devices to implement Freeze-TCP, and, in particular, be able to respond to such notifications by the MR. The latter approach is likely to be more viable in the long run for other transport- and application-layer protocols that are not as widely used as TCP, for which it may be unreasonable to expect the required protocol extensions to be supported transparently in the MR.

B. Wireless Link Disruption Prediction

The idea of predicting wireless link quality based on location and mobility patterns is not entirely new, and has been proposed before in certain related contexts. We now mention some of the recent proposals on mobility and location predictions, designed to improve network connectivity for stable end-to-end communication. Jiang *et al.* [21] presents a prediction method for future link availability. The calculation of link availability is based on the current node's movement, taking into account the speed and direction of motion between mobile nodes. Song *et al.* [22] present a study on empirical evaluation of a Markov recent history-based prediction scheme, and demonstrate that the users with extensive statistical data history can achieve a prediction accuracy as high as 72%. Su *et al.* [23] propose a predictive route reconstruction mechanism based on mobility patterns of the mobile users. The link expiration time between any two nodes is predicted by making use of the coordinates of the nodes, their speeds, and direction of motion, which are provided by global positioning system (GPS) equipment. Liang and Haas [24] present a predictive distance-based mobility management scheme for personal communication service (PCS) networks. In this scheme, the network uses a stochastic model for the velocity of mobile nodes, and predictions are made about the nodes' whereabouts in advance in the form of a probability density function. Finally, Akyildiz and Wang [25] propose a framework for mobility management and quality-of-service (QoS) in wireless networks, based on historic records and predictive patterns of mobile users.

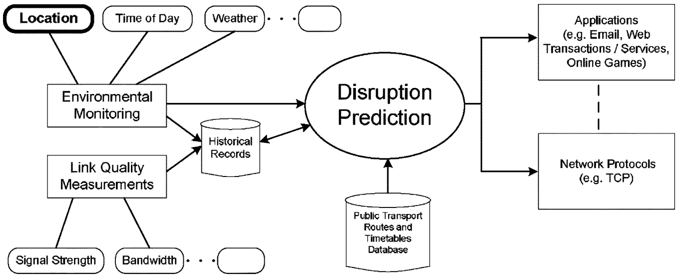


Fig. 2. Disruption prediction process.

However, none of the above works considered the specific issues arising in the context of disruption prediction for on-board mobile networks in public transport vehicles. Fig. 2 presents a high-level view of how the disruption prediction process can be implemented by the MR. First, it needs to track the relevant environmental factors—most importantly, location (which can be obtained by GPS or other, possibly proprietary, means), as well as any additional parameters that may impact the link quality, such as time of day, weather conditions, vehicle velocity, etc.[3] Together with measurements of the quality of the link itself (i.e., signal strength, available bandwidth, etc.), the above are recorded in a historical database, which is then used for the purpose of the disruption prediction itself, e.g., by comparing the current situation with past patterns where disruptions are known to have occurred. Further details about the implementation of the disruption prediction process in the MR, and, in particular, factors affecting the resulting prediction accuracy, are the subject of ongoing research and beyond the scope of this paper. We refer the reader to [27] for an overview of the issues involved.[4]

The warnings generated by the MR about impending disruptions can be utilized to improve the performance of the on-board network in several ways. In this paper, we focus on improvements at the transport layer, achieved transparently by a MR acting as a PEP. Another task that is the responsibility of the MR, and can gain from transparent use of prediction information without involving user devices, is vertical handover of connections, where the MR is equipped with several MWIC interfaces. In particular, in order to allow a MWIC transition (e.g., in a train

---

[2] The purpose of the TR-ACK is to make the remote host perform a fast-retransmit of packets possibly sent (and lost) before the ZWA was received, and, thus, avoid the undesirable consequences of a timeout.

[3] Indeed, the study presented in [26] of the dependence of cellular signal strength on a variety of environmental factors shows location to be the primary factor affecting the signal quality.

[4] While the disruption prediction approach fits naturally into the context of on-board LANs supported by MRs in public transport vehicles, we point out that, in principle, it can be implemented in other scenarios as well, such as in a private car commuting regularly on a fixed route, or by individual users in public transport vehicles where an on-board LAN is not deployed. However, the stability of routes traveled by public transport vehicles and the virtually unlimited memory and computational power available in the MR means that, in general, it can achieve a better prediction accuracy than is possible in other scenarios.

leaving a metropolitan cellular coverage area and switching to a satellite link) to be smooth and transparent to the users, the disruption of the existing link must be predicted well in advance, since the vertical handoff procedure normally entails a significant overhead of several round-trip exchanges. The problems arising in the management of multiple-MWIC networks are discussed elsewhere (see, e.g., [28]–[30]), and we do not consider them here further. Finally, we mention that the warnings generated by the disruption prediction process can also be distributed to the on-board user devices, which can use them to make informed decisions and actions at the application, session, and even transport layer. In particular, we expect this to be the norm with future disruption-tolerant applications, such as those considered by the DTN research group [31].

III. SYSTEM MODEL

Our results on the Freeze-TCP performance are based on a Markov model, which captures the behavior of both the TCP sender and the link. To describe the behavior of the wireless link, we follow the Gilbert–Elliot model, which is a two-state hidden Markov model commonly used for performance analysis over wireless links. In this model, the link alternates between the "up" and "down" states, spending a random amount of time in each state that is distributed exponentially with a mean of t_{up} and t_{down}, respectively. A packet transmitted while the link is in the "up" or "down" state is lost with a probability of p^l_{up} and p^l_{down}, respectively, such that $p^l_{\mathrm{up}} \ll p^l_{\mathrm{down}}$. For simplicity, in this paper we assume $p^l_{\mathrm{down}} = 1$, i.e., all packets are lost while the link is in the "down" state. In our setting, the "down" periods correspond to link outages, which can be predicted independently with a certain probability. On the other hand, packet losses incurred during the "up" state are considered to be random events unrelated to the vehicle location (e.g., caused by noise), and are therefore inherently unpredictable.

For clarity of presentation, we begin the description of the Freeze-TCP Markov model in the first subsection for the special case of $p^l_{\mathrm{up}} = 0$, i.e., assuming the "up" state is entirely error-free and all losses occur only during outages. The following subsection extends the model to account for random losses in the "up" state as well, i.e., for $p^l_{\mathrm{up}} > 0$. Finally, the third subsection describes some implementation considerations of the *ns-2* simulation platform.

A. *Freeze-TCP Model for a Link Without Random Losses*

We now proceed to describe the Markov chain structure in detail. A state is defined by a triplet $\langle W, C_{\mathrm{th}}, f\rangle$, where W is the size of the last successfully transmitted window, C_{th} is the current congestion window threshold between slow-start and congestion avoidance phases, and f is a boolean value denoting "freezing." Every state must have $0 \le W \le C_{\mathrm{th}} \le W_m$, where W_m is an external parameter of the model, denoting the maximum TCP window size (in packets).[5] States with $W < C_{\mathrm{th}}$ model TCP's "slow-start" phase, whereas $W = C_{\mathrm{th}}$ correspond to the "congestion avoidance" phase. A value of $f = 0$ corresponds to normal operation, whereas $f = 1$ is a "frozen" (persistent) state, which the protocol enters upon reception of a ZWA packet and leaves upon reception of a NZWA packet.

The states in our model capture the internal status of TCP with a granularity of a window; that is, the transitions among the states do not occur after every packet. This reduces the total number of states considerably, thereby facilitating the subsequent analysis. There is an approximation involved with this approach, however, in that the effect of a link failure in the middle of a window transmission cannot be captured with precision. Instead, below we take a conservative view, deeming any link failure to have occurred at the start of the corresponding window transmission.

There are four kinds of transitions possible among the states; these are outlined below. The first three types are from regular ($f = 0$) states; the last kind applies to "frozen" ($f = 1$) states.

1) If everything is normal, and the link stays up during the round-trip time (RTT), TCP advances normally. Thus, from any state in the slow-start phase ($\langle W, C_{\mathrm{th}}, 0\rangle$ where $0 < W < C_{\mathrm{th}}$), it will advance to $\langle \min(2W, C_{\mathrm{th}}), C_{\mathrm{th}}, 0\rangle$. From $0 = W < C_{\mathrm{th}}$ (a slow-start state just after a timeout in the previous window), it will proceed to $\langle W = 1, C_{\mathrm{th}}, 0\rangle$. Finally, from a congestion-avoidance state ($\langle W, C_{\mathrm{th}}, 0\rangle$ where $W = C_{\mathrm{th}}$), the transition is to $\langle W + 1, C_{\mathrm{th}} + 1, 0\rangle$, unless $W = W_m$, in which case it stays in the same state (the window is already at the maximum size).
2) If the link goes down and its failure is not predicted in time, a timeout occurs and TCP sets the congestion window threshold value to be half the current window size, before restarting in a slow-start phase. This corresponds to a transition from $\langle W, C_{\mathrm{th}}, 0\rangle$ to $\langle 0, \lfloor W/2\rfloor, 0\rangle$.
3) If the link goes down but its failure is predicted in time, TCP freezes its current state instead of falling back into slow-start. This corresponds to a transition from $\langle W, C_{\mathrm{th}}, 0\rangle$ to $\langle W, C_{\mathrm{th}}, 1\rangle$, for all states where $W \neq 0$. Observe that "freezing" is not possible from a state with $W = 0$, as that means that a timeout has already occurred.
4) From a "frozen" state ($\langle W, C_{\mathrm{th}}, 1\rangle$), there is a transition to the corresponding regular state $\langle W, C_{\mathrm{th}}, 0\rangle$ when the link resumes operation.

The corresponding transition rates depend on the transition type as well as the state from which they occur; indeed, states with $W = 0$, which denote that a timeout has just been suffered, require special consideration. The transition rates are detailed below.

- For all transitions of type 1 occurring from a state with $W > 0$, the rate is $1/\max(W \cdot t_x, T + t_x)$, where T is the connection's RTT, t_x is the transmission time of a single packet, and W is the window size at the state *to* which the transition is made.[6]
- For all transitions of type 1 from a state with $W = 0$ (i.e., after the link is down), the rate is the reciprocal of t_{down}, where t_{down} is the average link "down" time.

---

[5] We assume that the maximum window size is too small to cause congestion on the connection's path; therefore, TCP will not back off before reaching this window size.

[6] Recall that the interpretation of W is the size of the last window successfully transmitted; hence, the number of packets transmitted in a transition between states is given by W at the transition's destination state.

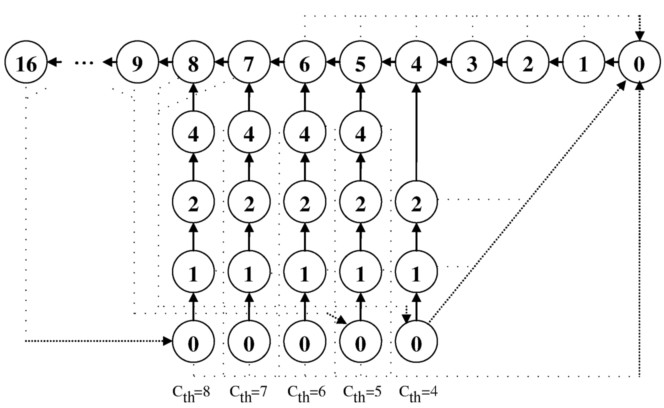


Fig. 3. The Freeze-TCP Markov chain model (excluding the "frozen" states), for $W_m = 16$. The solid arrows correspond to transitions of type 1; the dotted arrows are transitions of type 2. Transitions of types 3 and 4 (to and from the "frozen" states) are not shown.

- For all transitions of type 2 occurring from a state with $W > 0$, the rate is the misprediction probability (i.e., one minus the prediction probability) times the reciprocal of t_{up}, where t_{up} is the average link "up" time.
- For all transitions of type 2 from a state with $W = 0$ (all of which are to the state $\langle 0,0,0\rangle$), the rate is $1/(2T + t_x)$. Indeed, if the link is down during a second consecutive timeout (which is twice the RTT, due to Karn's exponential backoff), TCP will eventually resume operation in a congestion avoidance mode.
- For all transitions of type 3 (which, by definition, are from states with $W > 0$), the rate is the prediction probability times the reciprocal of $(t_{\mathrm{up}} - t_w)$, where t_w is the average warning period ($t_w = 0.5T$ for a perfectly timed successful prediction). In other words, the link "up" time is effectively reduced by the warning period. Note that the above implicitly assumes the average warning period to be small compared to the average "up" time, so that the "effective" up time can still be considered exponentially distributed (otherwise, obviously, it cannot be modeled by a Markov chain).
- For all transitions of type 4, the rate is the reciprocal of t_{down}, the average link "down" time.

Observe that, as per the transitions described above, many of the states are unreachable. These include, for example, all slow-start states where W is not 0 or a whole power of 2, or where $C_{\mathrm{th}} > W_m/2$, as well as all "frozen" states with $W = 0$. Furthermore, many functional equivalences among states can be noted; for instance, it makes no difference whether C_{th} is 0, 1, 2, or 3. Therefore, the analysis of the chain actually involves much fewer states than may seem at first.

Fig. 3 presents a diagram of the functional part of the chain, with the exception of the "frozen" states and transitions involving them (of types 3 and 4). Thus, it is to be kept in mind that, for any state in the diagram with $W > 0$, there is a corresponding "frozen" state, with transitions back and forth between them only (at the rates outlined above); there are no other transitions within the "frozen" states.

B. Model Extension for Random Losses in the "Up" State

We now proceed to extend the model to account for random packet losses that may occur while the link is in the "up" state, i.e., with $p_{\mathrm{up}}^l > 0$ in the Gilbert–Elliot link model. At first glance, incorporating random packet losses requires only the modification of the transition rates—specifically, increasing the rates of the timeout transitions to reflect the possibility of timeouts to occur due to random losses, not just (unpredicted) outages. However, this causes the model to result in an excessively low throughput, since it does not account for the fact that TCP does not always time out after a lost packet; indeed, if a lost packet is followed by three correct ones, the *fast retransmit* procedure is triggered, and the congestion window is merely halved rather than reset back to 1 packet altogether.

To account for the fast retransmit behavior of TCP, we introduce a set of auxiliary states, denoted by $\langle \mathrm{FR}, C_{\mathrm{th}}, f\rangle$. As for "regular" states, C_{th} denotes the congestion window threshold and f is a Boolean value denoting "freezing." However, these states do not have a current window size $W \geq 0$; instead, "FR" is used to denote that these states are introduced to model the fast retransmit feature. Specifically, a state $\langle \mathrm{FR}, C_{\mathrm{th}}, f = 0\rangle$ denotes that the TCP sender has attempted to recover after a loss of a packet followed by a TR-ACK, with the congestion window having been reduced to C_{th}. Therefore, the range of C_{th} in these auxiliary states is between 2 to $W_m/2$, where W_m is the maximum window size of the TCP connection. (Note that fast retransmit is not possible with $C_{\mathrm{th}} = 1$; for a TR-ACK to follow a lost packet, the previous window size must have been large enough for at least four segments.)

The new auxiliary states are interwoven with the rest of the chain by the same four types of transitions described above. In case of a successful transmission, TCP "recovers" from the state $\langle \mathrm{FR}, C_{\mathrm{th}} = w, f = 0\rangle$ to the regular state $\langle W = w, C_{\mathrm{th}} = w, f = 0\rangle$, and proceeds with congestion avoidance from there. A timeout (e.g., due to an unpredicted outage) occurring in any of the auxiliary states causes a transition to $\langle W = 1, C_{th} = 1, f = 0\rangle$. A predicted outage causes a "freeze" transition from $\langle \mathrm{FR}, C_{\mathrm{th}}, f = 0\rangle$ to its corresponding state $\langle \mathrm{FR}, C_{\mathrm{th}}, f = 1\rangle$, and the transition in the opposite direction occurs when the link becomes operational again. The rates of the above four transition types are determined identically to the rates of similar transitions from the regular states.

In addition, a fifth transition type is required for the fast retransmit occurrences, where the chain advances from a regular state $\langle W = w, C_{\mathrm{th}} = v, f = 0\rangle$ to the auxiliary state $\langle \mathrm{FR}, C_{\mathrm{th}} = u, f = 0\rangle$, where $u = \lfloor (1/2)\min(2w, v+1)\rfloor$.[7] The rate associated with this type of transition is $(1/\max(u \cdot t_x, T + t_x)) \cdot P_{\mathrm{FR}}(u)$, where T is the connection RTT, t_x is the transmission time of a packet (so that $\max(u \cdot t_x, T + t_x)$ is the time between the start of a transmission of a window of size u and the start of the transmission of the next window), and $P_{\mathrm{FR}}(u)$ is the probability of a fast retransmit event to occur due to a transmission attempt of a window of size u, derived in a straightforward manner from the packet loss rate of the link.

---

[7] Recall that a state $\langle W, C_{\mathrm{th}}, f\rangle$ denotes a situation that a window of size W has been *successfully* transmitted; thus, the fast retransmit occurs during the transmission attempt of the next window, which is of size $\min(2W, C_{\mathrm{th}} + 1)$.

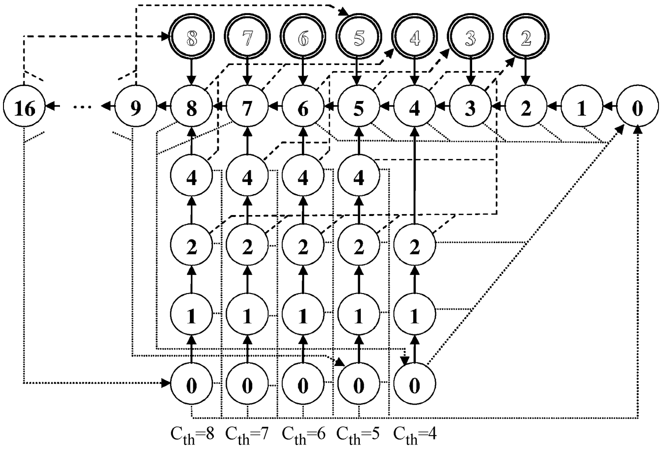


Fig. 4. The Freeze-TCP Markov chain model (excluding the "frozen" states), extended for random packet losses, for $W_m = 16$. The new auxiliary states corresponding to TCP's fast retransmit feature are denoted by double circles.

Fig. 4 presents the extended diagram of the functional part of the chain (again, without the "frozen" states and transitions involving them). The new auxiliary states are shown as double circles, and the numbers therein denote the respective congestion window thresholds (and, for this reason, appear in a different font to distinguish them from the other states, where the number inside the circle is the size of a successfully transmitted window).

Finally, the model is solved by finding the steady-state probabilities of the chain states (e.g., by the standard technique of inverting the transition rate matrix), and, subsequently, using them to compute the expected throughput. The protocol throughput is composed of two separate components. One is the fully successful window transmissions; in a state $\langle W, C_{\mathrm{th}}, f = 0\rangle$, this contributes $W/(T+t_x)$ packets per time unit (unless $W \cdot t_x > T$, in which case the throughput is limited by $1/t_x$). The other component consists of partial successes, i.e., portions of the window successfully received before a random loss. The contribution of this component is calculated by considering the transitions corresponding to timeouts and fast retransmit occurrences, and, for each such transition, multiplying its steady-state rate by the *conditional* expected value of the successful portion of the window, given that a loss occurred (again, this is obtained in a straightforward manner for every window size from the packet loss rate of the link). Obviously, the throughput in all the "frozen" states is 0. Averaging the throughputs according to the steady-state probabilities yields the throughput performance of the protocol.

To conclude, we note that our Markov TCP model bears a certain similarity to the one presented in [32]; indeed, we follow a similar approach in modeling the slow-start and congestion-avoidance phases, and the auxiliary states introduced to capture the fast retransmit TCP behavior are similar to the states with $l = 1$ in [32]. There are also some differences—for example, we provide separate states for all values of C_{th} up to $W_m/2$, and not just those which are whole powers of 2, as in [32]; this makes the agreement between our model and simulations somewhat more precise, for only a modest cost in complexity. However, the most obvious difference from [32], and indeed the motivation behind this work, is the inclusion of "frozen" states, to model the behavior of Freeze-TCP. We emphasize that our method of duplicating the Markov chain states by introducing a extra Boolean parameter f, so that the states corresponding to $f = 1$ are used to "freeze" their counterparts with $f = 0$, can be used on any Markov model of TCP. Thus, whilst we have focused on modeling the TCP-Reno flavor (as has [32]), our method can be readily used with Markov models proposed for other TCP flavors (e.g., SACK and Vegas [33]).

C. Simulation Model

To verify the validity of the Freeze-TCP Markov model, we have implemented a simulation platform, using the *ns-2* simulator [34]. The implementation of the Freeze-TCP extensions, coupled with disruption prediction, required significant portions of code to be added/edited in various modules of the standard *ns-2* distribution. Following are some details on the challenges we faced and our implementation decisions.

First, the standard *ns-2* does not provide support for the TCP receiver to detect link states and send ZWA packets. There is also no support for the sender to process ZWAs, i.e., freeze the current operating variables (including the congestion window, *ssthresh*, timers, etc.). We have addressed the sender-side limitation by writing a new agent that can detect and process ZWAs. This agent was executed with the fast retransmit/recovery option enabled. The modification of the receiver was more challenging, because the receiver needed to generate ZWAs and NZWAs according to the predictions of the link outages. This required us to develop an interface that allows the receiver to communicate with a link-status module and generate ZWA/NZWA packets with a given prediction probability, with the durations of time within and between disruptions sampled from the exponential distribution.

Second, although the standard *ns-2* loss model allows to configure a link with particular loss rates, it does not allow complete disruptions (setting loss rate to 100% merely cuts off communication from sender to receiver, but receiver can still send ACKs to the sender). To implement full disruptions, we used the network dynamics interface to completely disconnect the link between sender and receiver at the start of a disruption, save the current loss parameters, and eventually reconnect the link with these saved parameters.

Finally, the standard *ns-2* does not have any automatic mechanism to control the simulation durations so as to achieve a desired confidence level with required precisions. We have used a publicly available simulation management tool, Akaroa [35], that was interfaced with our simulator to adaptively control the duration of all our simulation experiments. All simulations were automatically stopped when a 95% confidence level was reached with 5% relative precision.[8] The actual simulation run durations varied between 50,000–200,000 s, depending on the simulation parameters.

IV. ANALYSIS AND SIMULATION RESULTS

To verify the validity of the Freeze-TCP Markov model and evaluate the protocol performance, we have conducted several

---

[8] In addition to computing confidence levels online, the Akaroa tool uses sophisticated methods (based on spectral analysis) to detect the end of the transient phase and any possible correlations in the output data. All data in the transient phase are discarded before computing the confidence level.

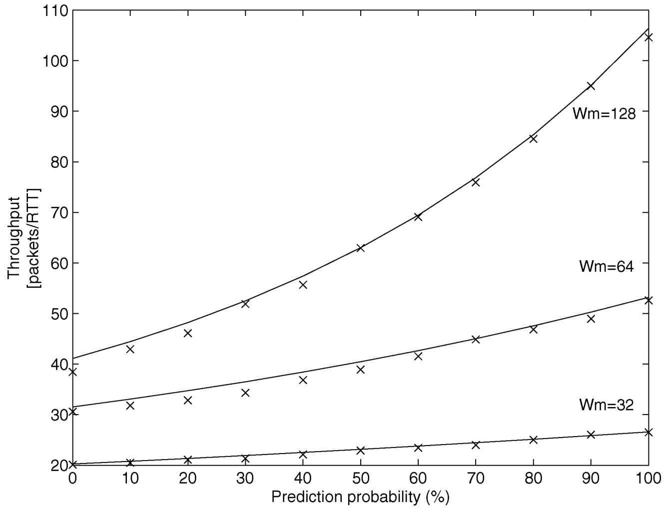


Fig. 5. Average throughput: "up" time of $50 \cdot \mathrm{RTT}$ and "down" of $10 \cdot \mathrm{RTT}$.

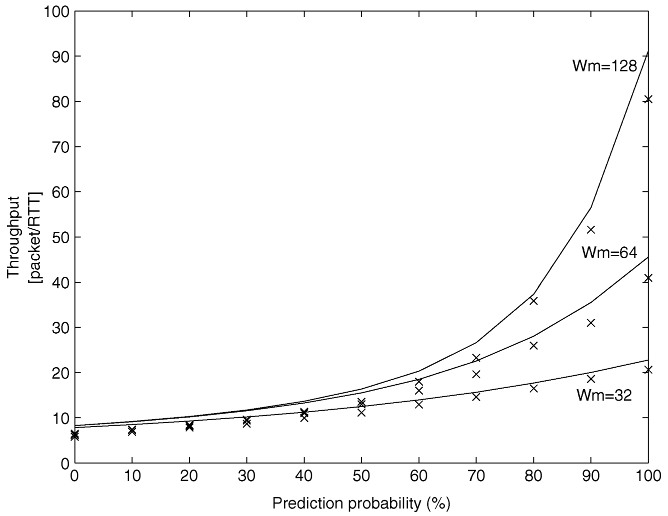


Fig. 6. Average throughput: "up" time of $10 \cdot \mathrm{RTT}$ and "down" of $4 \cdot \mathrm{RTT}$.

sets of experiments and compared the throughput predicted by the model with that achieved in simulation for a variety of scenarios. All these cases assume a TCP connection between endpoints engaged in a long file transfer, using segments (packets) of 1000 bytes and a maximum TCP window size of either 32, 64, or 128 segments. The bandwidth of the connection is taken to be 1 Mb/s, and the one-way delay (in each direction) is set to 250 ms. Finally, we vary the probability of successful prediction between 0% and 100%. The results are presented below.

A. Throughput Dependence on Window Size and Link Stability

For our first set of experiments, we take a link with an average "up" time of $50 \cdot \mathrm{RTT}$ and an average "down" time of $10 \cdot \mathrm{RTT}$, and compare it with a less stable link, with an average "up" time of $10 \cdot \mathrm{RTT}$ and an average "down" time of $4 \cdot \mathrm{RTT}$. These values were selected to best illustrate the benefits made possible by outage prediction. Indeed, if the average "up" time is higher than the RTT by many orders of magnitude, outage prediction becomes insignificant, since the slow-start and congestion avoidance phases take a negligible portion of time anyway. Conversely, if the link is so unstable that it frequently changes states within a single RTT, prediction becomes hardly relevant, since even the ZWA and NZWA notifications cannot be properly communicated. The results are shown in Figs. 5 and 6. In both figures, the plots show the TCP throughput, as predicted by the respective Markov model solution, as solid curves; the throughput achieved in the actual simulation runs is shown with "x"-marks.

We observe that our model achieves a good accuracy in predicting the Freeze-TCP throughput. The fact that the predicted throughput is consistently higher, by a slight margin, than the value actually attained, can be explained by observing that, in our model (and similarly in [32] and [33]), the TCP window in the congestion-avoidance phase is assumed to grow by one packet per round-trip period, whereas in reality, TCP increases the window size by the reciprocal of its current value upon reception of every acknowledgment. This effect causes the window to grow slightly more slowly (e.g., from a window of two packets, reception of two acknowledgments causes it to grow to $2 + (1/2) + (1/2.5) = 2.9$ after a RTT, which is slightly less than 3); thus, it takes a slightly higher number of RTT periods than predicted by the model to attain the maximum window size.

Another important conclusion from the plots is that an investment in a good prediction method, correctly predicting a high percentage of outages, becomes more worthwhile as the window sizes are larger or the link is less stable. Indeed, we observe that for $W_m = 32$ the lines are nearly linear, while for $W_m = 64$ and $W_m = 128$ the curves become increasingly quadratic; furthermore, the curves in Fig. 6 are more convex than those in Fig. 5. This phenomenon can be explained as follows. The time it takes TCP to complete the slow-start and congestion avoidance phases is constant. Therefore, if most of the outages happen after TCP has already reached the maximum window size, then each unpredicted outage inflicts a fixed penalty on the TCP performance, namely, the difference between TCP's maximum throughput and that achieved during the slow-start and congestion avoidance phases. In that case, the throughput simply increases linearly in the percentage of outages that are predicted. However, if most of the outages happen while TCP is still in the congestion avoidance stage, then each additional correct prediction has an increasing contribution to the throughput, as it happens at a higher window size. The dependence of the throughput on the percentage of correctly predicted disruptions then becomes more convex. This effect is graphically illustrated in Fig. 7. Therefore, the more likely outages are to happen during the congestion avoidance phase—in other words, the larger the maximum window size and/or the less stable the link—the more convex is the throughput dependence on the prediction probability.

B. Throughput Dependence on Random Link Losses

We now introduce random packet losses in the link and consider their impact on the performance achieved by Freeze-TCP. The random packet loss rate while the link is "up" is varied between 10^{-5} and 10^{-2}. Figs. 8 and 9 plot the resulting throughput as a function of the prediction probability, for several values of the random loss rate for a maximum window size of 64 and 128,

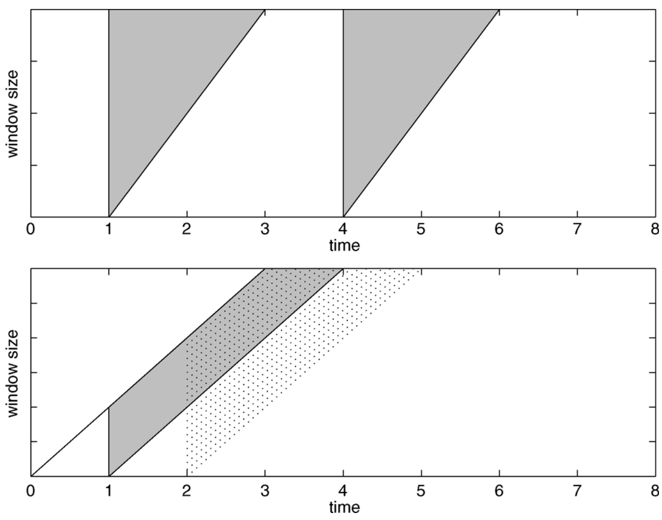


Fig. 7. Illustration of outages during maximum window size (top, $t = 1, 4$) versus outages during congestion avoidance phase (bottom, $t = 1, 2$). The shaded areas represent the difference (in transmitted packets) between a predicted and an unpredicted outage. In the bottom figure, misprediction of the outage at $t = 2$ (dashed area) loses more than that at $t = 1$ (solid area).

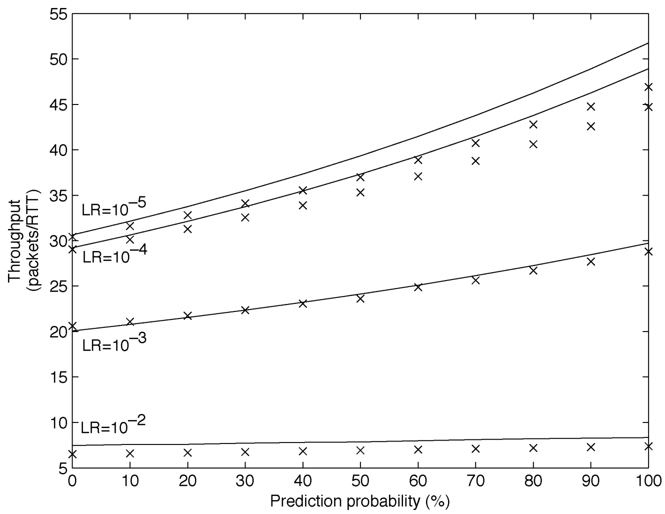


Fig. 8. Average throughput as a function of prediction probability, for several packet loss rates, $\mathrm{maximum\ window} = 64\ \mathrm{packets}$.

respectively. (Again, the throughput predicted by the Markov model is shown as solid curves, and that achieved in actual simulation runs is shown with "x"-marks.) These figures show that as the link loss increases, the advantage gained by a high outage prediction probability becomes less significant. This can be explained by the fact that, as the random loss rate increases, outages account for a comparatively lesser proportion of lost packets. Consequently, the throughput that would have been achieved by TCP even without outages is lower. Nevertheless, it can be seen that, even in the presence of random losses (as long as the random packet loss rate remains reasonable, i.e., lower than about 1%), correct prediction of a high percentage of outages brings about a significant improvement in the TCP throughput.

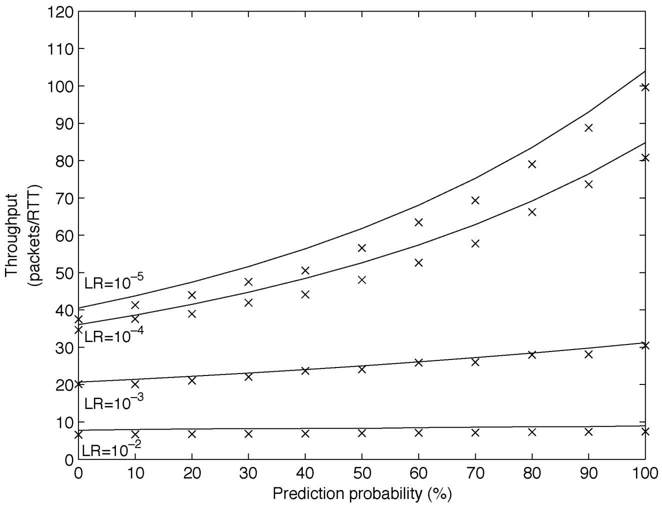


Fig. 9. Average throughput as a function of prediction probability, for several packet loss rates, $\mathrm{maximum\ window} = 128\ \mathrm{packets}$.

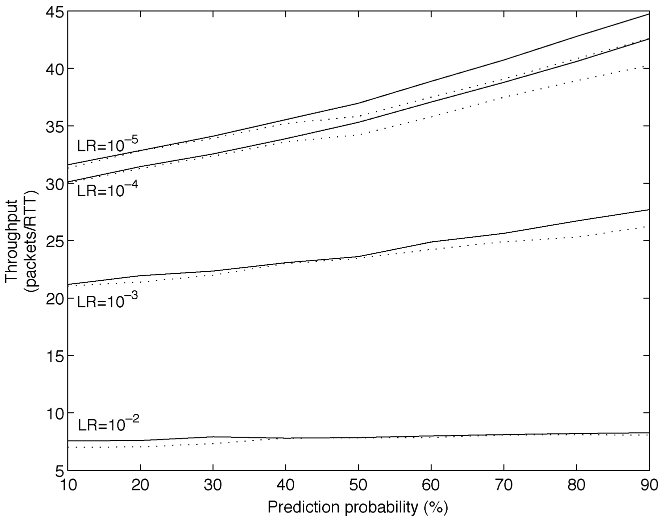


Fig. 10. Average throughput: perfect versus imperfect warning period, $\mathrm{maximum\ window} = 64\ \mathrm{packets}$.

C. Throughput Dependence on the Warning Period

We now explore how the throughput is affected by the distribution of the warning period. All the results so far were obtained with a warning period set deterministically to a "perfect" value, namely, the one-way delay of the connection (250 ms), regardless of the prediction probability. We compare this with the case that the warning period is exponentially distributed with a mean of 250 ms$/-\ln P_{pr}$, where P_{pr} is the prediction probability. This way, the prediction probability is equivalent to the probability of the warning period to be above 250 ms; indeed, a warning given less than a one-way delay before the outage is worthless, as it does not reach the remote endpoint in time to prevent it from sending a packet that is doomed to be lost.

Our results are shown in Figs. 10 and 11, where the solid curves correspond to a "perfect" deterministic warning period as before, and the dotted curves are obtained with warning periods that are exponentially distributed with the respective prediction probabilities, as described above (incidentally, note that

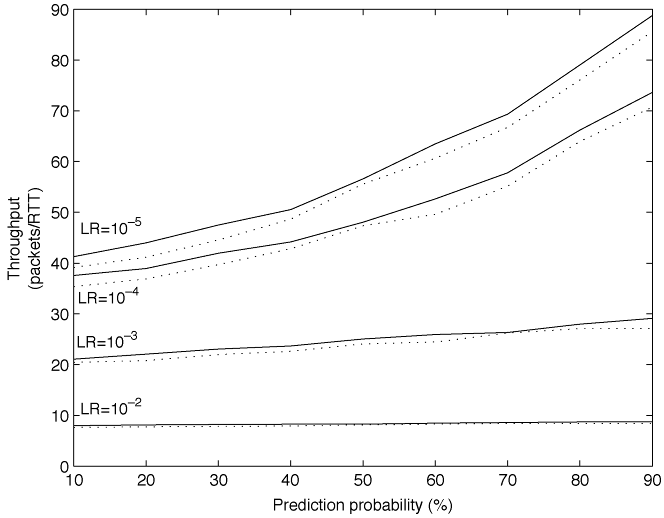


Fig. 11. Average throughput: perfect versus imperfect warning period, $\text{maximum window} = 128\ \text{packets}$.

prediction probabilities of 0% and 100% had to be omitted for this reason). It can be seen that the "imperfect" (exponentially distributed) warning period has a very limited impact on the throughput attained by Freeze-TCP, compared with a deterministic perfect one, irrespective of the maximum window size. This is an encouraging finding; in particular, it suggests that, even when the vehicle location cannot be determined with high accuracy, a conservative (i.e., early) timing of the warning may still achieve most of the potential performance improvement.

V. CONCLUSION

Integration of mobile communication infrastructure in public transport vehicles is recently becoming an important trend in wireless networking. In this paper, we have introduced the concept of *disruption prediction* of wireless communication links for on-board mobile networks in public transport vehicles, based on the predictable and repetitive nature of their routes. Specifically, we have advocated the use of the Freeze-TCP extension for on-board mobile networks. We proposed a Markovian model of Freeze-TCP to analyze the benefits of such outage predictions on TCP throughput, and studied its dependence on the prediction probability, random link loss rate, and prediction warning period. In particular, we have demonstrated the extent of the performance gains that are made possible by outage prediction, showing that these gains can be significant even if not all outages are predicted, the timing of the outage warnings is not perfect, or the outages are not the only cause of losses.

In this paper, we did not consider the implementation details of disruption prediction in the MR, and made no assumptions about any particular method to obtain warnings about impending outages. For tractability of the mathematical model, we only assumed outage durations and interoutage times to be exponentially distributed, and predictions to be successful with a probability independently of each other. Further research is called for to gain insight into the factors that affect the prediction accuracy and the true outage and prediction dynamics that can be expected in realistic on-board mobile communication networks. This research direction is being undertaken in our ongoing work.

REFERENCES

[1] A. Baig, M. Hassan, and L. Libman, "Prediction-based recovery from link outages in on-board mobile communication networks," in *Proc. IEEE GLOBECOM*, Dallas, TX, Nov. 2004, pp. 1575–1579.

[2] PointShot Wireless. [Online]. Available: http://www.pointshotwireless.com

[3] icomera. [Online]. Available: http://www.icomera.com

[4] 21Net. [Online]. Available: http://www.21net.com

[5] Inmarsat's Swift64. [Online]. Available: http://www.inmarsat.com/swift64

[6] Connexion by Boeing. [Online]. Available: http://www.connexionbyboeing.com

[7] Nautilus 6 Working Group. [Online]. Available: http://www.nautilus6.org

[8] PerMoCom [Online]. Available: http://www.cemnet.ntu.edu.sg/Permocom/permocom.shtml

[9] K.-D. Lin and J.-F. Chang, "Communications and entertainment onboard a high-speed public transport system," *IEEE Wireless Commun.*, vol. 9, no. 1, pp. 84–89, Feb. 2002.

[10] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," RFC 3963, Jan. 2005.

[11] A. Annamalai, C. Tellambura, and V. K. Bhargava, "Simple and accurate methods for outage analysis in cellular mobile radio systems—A unified approach," *IEEE Trans. Commun.*, vol. 49, no. 2, pp. 303–316, Feb. 2001.

[12] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 1537–1545.

[13] G. Xylomenos, G. C. Polyzos, P. Mähönen, and M. Saaranen, "TCP performance issues over wireless links," *IEEE Commun. Mag.*, vol. 39, no. 4, pp. 52–58, Apr. 2001.

[14] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. H. Katz, "Comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[15] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP performance over wireless networks," in *Proc. ACM MobiCom*, Berkeley, CA, Nov. 1995, pp. 2–11.

[16] A. V. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Vancouver, BC, Canada, May 1995, pp. 136–143.

[17] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 5, pp. 19–43, Oct. 1997.

[18] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: a reliable transport protocol for wireless wide-area networks," in *Proc. ACM MobiCom*, Seattle, WA, Aug. 1999, pp. 231–241.

[19] R. Chakravorty, S. Katti, J. Crowcroft, and I. Pratt, "Flow aggregation for enhanced TCP over wide-area wireless," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 2003, pp. 1754–1764.

[20] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," RFC 3135, Jun. 2001.

[21] S. Jiang, D. He, and J. Rao, "A prediction-based link availability estimation for mobile ad hoc networks," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 1745–1752.

[22] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 1414–1424.

[23] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Manage.*, vol. 11, no. 1, pp. 3–30, Jan. 2001.

[24] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional PCS networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 718–732, Oct. 2003.

[25] I. F. Akyildiz and W. Wang, "The predictive user mobility profile framework for wireless multimedia networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1021–1035, Dec. 2004.

[26] I. Chan, A. Chung, M. Hassan, K.-C. Lan, and L. Libman, "Understanding the effect of environmental factors on link quality for on-board communications," in *Proc. IEEE Veh. Technol. Conf.*, Dallas, TX, Sep. 2005, pp. 1877–1881.

[27] B. Benatallah, M. Hassan, L. Libman, and A. Sun, "OCEAN: Scalable and adaptive infrastructure for on-board information access," School of Comput. Sci. Eng., Univ. New South Wales, Tech. Rep. 601, Jan. 2006. [Online]. Available: ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/0601.pdf

[28] C. Ng, E. Paik, and T. Ernst, "Analysis of multihoming in network mobility support," Internet draft: draft-ietf-nemo-multihoming-issues-02.txt (work in progress), Feb. 2005.

[29] A. Chung and M. Hassan, "Optimizing profit and performance for multi-homed mobile hotspots," in *Proc. IEEE ICC*, Seoul, Korea, May 2005, pp. 924–928.

[30] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee, "MAR: A commuter router infrastructure for the mobile Internet," in *Proc. ACM MobiSys*, Boston, MA, Jun. 2004, pp. 217–230.

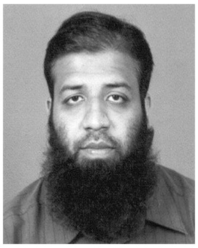
[31] DTN Research Group. [Online]. Available: http://www.dtnrg.org

[32] C. Casetti and M. Meo, "An analytical framework for the performance evaluation of TCP Reno connections," *Comput. Netw.*, vol. 37, no. 5, pp. 669–682, Nov. 2001.

[33] A. Wierman, T. Osogami, and J. Olsén, "A unified framework for modeling TCP-Vegas, TCP-SACK, and TCP-Reno," in *Proc. IEEE/ACM Int. Symp. Modeling, Analysis and Simulation of Comput. Telecommun. Syst.*, Orlando, FL, Oct. 2003, pp. 269–278.

[34] The network simulator—ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns

[35] Project Akaroa. [Online]. Available: http://www.cosc.canterbury.ac.nz/research/RG/net_sim/simulation_group/akaroa



Adeel Baig (S'02) received the B.E. degree in electronics engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 1998 and the M.Eng.Sc. degree in computer engineering from the University of New South Wales, Sydney, Australia, in 2001. He is currently working towards the Ph.D. degree at the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia.

His current research interests are in the protocols and applications for on-board mobile communication networks, network optimization, and QoS provisioning.

Mr. Baig is a student member of the Australian Computer Society and a lifetime member of the Pakistan Engineering Council.

Lavy Libman (S'97–M'04) received the B.Sc. degrees in electrical engineering and in computer engineering, and the M.Sc. and Ph.D. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1992, 1997, and 2003, respectively.

Since September 2003, he has been a Researcher at the Networks and Pervasive Computing Research Program, National ICT Australia, Sydney. He is also a Conjoint Lecturer in the School of Electrical Engineering and Telecommunications, University of New South Wales. Previously, he held several visiting and consulting summer positions, including Bell Laboratories, Murray Hill, NJ, in 2002, and Millimetrix Broadband Networks, Givat Hashlosha, Israel, in 2000. From 1993 and 1999, he served as a Computer Engineer in the Israel Defence Forces. His research interests include mobile and disruption-tolerant network architectures, cross-layer performance optimization in wireless networks, and game-theoretic aspects of network architecture and protocols.

Dr. Libman has been a member of technical program committees of several international conferences, including IEEE INFOCOM, IEEE Conference on Local Computer Networks (LCN), IEEE Vehicular Technology Conference, QShine, WiOpt, as well as ACM Sigcomm Asia Workshop. He was also a member of the executive committee of IEEE INFOCOM 2002.

Mahbub Hassan (M'91–SM'00) received the M.Sc. degree from the University of Victoria, Victoria, BC, Canada and the Ph.D. degree from Monash University, Melbourne, Australia.

He is Associate Professor and Head of Networking Research in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. From April to May 2005, he was an Invited Professor at the University of Nantes, France, working in INRIA's ATLAS project. He is affiliated with National ICT Australia (NICTA), where he leads a project on large-scale wireless mesh networks as part of the Smart Transport and Roads (STAR) mega project. He has coauthored numerous research articles, as well as one IETF draft and three books, including the widely adopted graduate textbook *High Performance TCP/IP Networking* (Prentice Hall, 2004). He serves on the Editorial Advisory Board of the *Computer Communications Journal* (Elsevier Science). He organized and guest edited special issues for the *Journal of Supercomputing* and *Real Time Imaging Journal*. He has worked closely with large industrial research laboratories and developed industry short courses on leading edge networking topics.

Dr. Hassan was the recipient of the Teaching Excellence Award from the School of Computer Science, Monash University, in 2000, and was nominated for the Research Excellence Award from the Faculty of Engineering, University of New South Wales, in 2004. He received several Australian Research Council (ARC) research grants in the area of mobile and wireless networking. He served as an Associate Technical Editor of the *IEEE Communications Magazine* from 1999 to 2001, and was a Guest Editor of the magazine's Feature Topic on TCP Performance (April 2001 issue). He was organizer and Chair of the IEEE ICON Wireless Workshop, Sydney, 2003, and Co-chair for the SPIE Conference on Internet Quality-of-Service from 2001 to 2003.