

## RCP-Planet: A rate control protocol for InterPlaNetary Internet

Jian Fang<sup>1,\*,\dagger</sup> and Ian F. Akyildiz<sup>2,\ddagger</sup>

<sup>1</sup>*Jewelry Television, Knoxville, TN 37922, U.S.A.*

<sup>2</sup>*Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.*

### SUMMARY

UDP traffic is a part of the aggregate traffic over InterPlaNetary Internet backbone links, which includes planet images and multimedia data from some scientific observations. Existing rate control schemes cannot solve the rate control problem in InterPlaNetary Internet which is characterized by *extremely long propagation delays, high link errors, asymmetrical bandwidth, and blackouts*. In this paper, a rate control protocol, RCP-Planet, is proposed to address all the above challenges. RCP-Planet consists of two novel algorithms, i.e. *Begin State* and *Operational State*. The protocol is based on a novel rate probing mechanism, i.e. sending probing sequences to capture the available bandwidth. A new rate control scheme is designed to update the transmission rate based on the observed rate for the probing sequence. Tornado codes are used for packet-level FEC because of their fast encoding and decoding speed. Bandwidth asymmetry problem is addressed by FEC block-level ACKs. Moreover, RCP-Planet incorporates Blackout State into the protocol to improve performance in blackout conditions. Simulation results show that RCP-Planet achieves high throughput performance, fairness, and is delay tolerant. Copyright © 2007 John Wiley & Sons, Ltd.

Received 8 April 2006; Accepted 7 August 2006

KEY WORDS: InterPlaNetary Internet; rate control; extremely long propagation delay; blackout

### 1. INTRODUCTION

Recently, the research interests in deep space are arising rapidly, which include scientific spacecraft travelling, Mars exploration, radio and radar astronomy observations of the solar system and the universe. The future space missions to deep space require communication among planets, moons, satellites, asteroids, robotics spacecrafts, and crewed vehicles. The scientific

\*Correspondence to: Jian Fang, Jewelry Television, Knoxville, TN 37922, U.S.A.

<sup>\dagger</sup> E-mail: jxF6483@yahoo.com, jian.fang@acntv.com

<sup>\ddagger</sup> Professor.

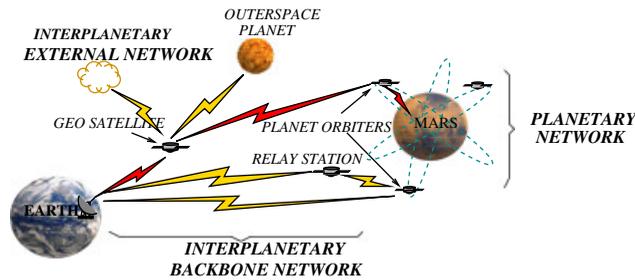


Figure 1. The InterPlaNetary Internet architecture.

data from these missions need to be delivered to the Earth successfully. In order to achieve this goal, InterPlaNetary Internet is proposed to be the Internet of the deep space planetary networks [1, 2].

A typical InterPlaNetary Internet architecture shown in Figure 1 is proposed for the Mars Exploration Mission [3], which includes *InterPlaNetary Backbone Network*, *InterPlaNetary External Network*, and *Planetary Network*.

- *InterPlaNetary Backbone Network:* It provides a common infrastructure for communications among the Earth, outer-space planets, moons, satellite relays, etc. It includes the data links (direct link or multihop paths) between elements with long-haul capabilities.
- *InterPlaNetary External Network:* It consists of spacecrafts flying in groups in deep space between planets, clusters of sensor nodes, and groups of space stations, etc. Some nodes also have long-haul communication capabilities.
- *Planetary Network:* This architecture can be implemented at any outer-space planet, providing interconnection and co-operation among the satellites and surface elements on a planet.

We focus on the Mars–Earth communication. The data are first transmitted from the ground station on the Mars surface, which is the source end-point, to Mars orbiters, Mars orbiters then send the data to the satellites orbiting Earth either directly or through the relay station near Mars. The Earth satellites then deliver the multimedia data to the ground station on Earth, which is the sink end-point. Along the communication path as shown in Figure 1, the InterPlaNetary Backbone Network plays a significant role for the performance of the entire deep space communication. The characteristics of InterPlaNetary Backbone Network can be summarized as follows [4]:

- *Extremely long propagation delays:* The InterPlaNetary backbone links usually have extremely long propagation delays. For example, the end-to-end round trip time for the Mars–Earth communication network varies from 8.5 to 40 min according to the orbital location of the planets [5].
- *High link error rates:* The bit error rates on the deep space links are very high usually in the order of  $10^{-1}$  [5].
- *Blackout:* Periodic link outages may occur due to orbital obscuration with the loss of line of sight because of moving planetary bodies, the interference of an asteroid or a spacecraft [3]

- *Bandwidth asymmetry*: The asymmetry in the bandwidth capacity of forward and reverse channels is typically in the order of 1000:1 in space missions [5].

UDP traffic is a part of the aggregate traffic over the deep space communication links [3]. Some audio and visual information including planet images and data from scientific observations will be transmitted *via* these links. This type of traffic does not require 100% reliability and mostly has strict requirements on jitter bound, minimum bandwidth, and smooth traffic [6]. The multimedia applications are usually classified into two classes: streaming of stored or live multimedia and real-time interactive multimedia. Obviously, real-time interactive multimedia is not applicable over InterPlaNetary Internet backbone links because of the extremely long propagation delays and due to the same reason, live or stored media streaming is not feasible, either. However, the multimedia data can be sent to the receiver on the Earth and buffered to replay them later on. Retransmissions cause high overhead considering the huge propagation delay in the deep space and they may be unnecessary for the UDP traffic since these types of traffic can tolerate the loss to a certain degree. The control for the UDP traffic is an important problem, because uncontrolled UDP traffic can not only congest the network, but can also cause unfairness and starvation for other types of data traffic.

Rate control protocols need to address the challenges posed by the InterPlaNetary Internet Backbone Network, i.e. *extremely long propagation delays, high link errors, asymmetrical bandwidth, and blackouts*. Furthermore, UDP traffic over InterPlaNetary Internet can be coded in MPEG, motion JPEG, or H.26x. Even though error resilience techniques are adopted in coded video [7], compressed video is still highly sensitive to data loss. The quality of other types of multimedia can also be degraded dramatically if the packet loss rate is high. Without retransmissions, the rate control protocols should be able to deal with the packet losses due to link errors or congestions in the InterPlaNetary Internet. Apart from that, InterPlaNetary Internet also requires smooth traffic. The source cannot get the feedback until one RTT later. Due to the very long propagation delay, the congestion control decision based on such past information might not lead to proper actions. As a result, rapid changes in the transmission rate may lead to serious congestions and packet losses.

As discussed in Section 2, the existing rate control schemes cannot address the challenges posed by the InterPlaNetary Internet Backbone Network. In this paper, we introduce a rate-control protocol, RCP-Planet, for the UDP traffic in the InterPlaNetary Internet. The objective of RCP-Planet is to address the challenges in the InterPlaNetary Internet to achieve high throughput and to address the error control problem. To address the extremely long propagation delays, RCP-Planet deploys a Begin State algorithm in the first RTT and an Operational State algorithm after one RTT to control the multimedia traffic. A novel rate-probing mechanism is proposed to capture the available network bandwidth. Based on the observed rate for a probing sequence, a new rate control scheme is designed to update the transmission rate smoothly and conservatively to reduce the chances of congestion. In order to address the error control problem for UDP applications, packet-level FEC is adopted to recover lost packets. Moreover, FEC block-level ACKs are used to solve the bandwidth asymmetry problem. In order to reduce the performance degradation due to blackout conditions, a Blackout State is incorporated into RCP-Planet. Performance evaluation reveals that RCP-Planet achieves high throughput and is delay tolerant by addressing the challenges in the InterPlaNetary Internet and the error control problem.

The remainder of the paper is organized as follows. First, related works are addressed in Section 2. RCP-Planet protocol overview and the Begin State algorithm are presented in Section 3, which includes packet-level FEC, the rate-probing mechanism, and the Begin State algorithm. The Operational State algorithms including the new rate control scheme, the Blackout State behaviour, and bandwidth asymmetry are explained in Section 4. Performance evaluation is presented in Section 5 and is followed by the conclusions in Section 6.

## 2. RELATED WORK

Many rate-control protocols are proposed to control the flow of UDP traffic in terrestrial networks [8–15]. These proposed protocols can be mainly categorized into two types of rate control schemes, i.e. AIMD based (additive increase multiplicative decrease) and equation based.

AIMD-based rate control schemes are TCP-compatible, i.e. they compete fairly with existing TCP by changing the sending rate in such a way similar to that of TCP [8, 12, 13]. The existing AIMD-based rate control schemes [8–10, 12, 13] are developed based on the assumption that the propagation delay is relatively short. SCTP (Stream Control Transmission Protocol) [9] implements TCP-like mechanisms, such as slowstart, fast retransmit, and fast recovery. SCP (Streaming Control Protocol) [8] is a modified version of TCP that performs TCP-Vegas-like rate adjustment. TEAR (TCP emulation at receiver) [10] determines the receiving rates at the receiver based on signals, such as packet arrivals, packet losses, and timeouts. Using these signals, TEAR emulates the TCP flow-control functions at the receiver including slow start, fast recovery, and congestion avoidance. RAP (Rate Adaptation Protocol) [13] is a rate-based congestion control mechanism for wired and short distance networks. RCS (rate control scheme) [12] is a rate control scheme for real-time traffic in networks with high bandwidth-delay products and lossy links. The traditional AIMD-based rate control schemes perform rate increase additively at each RTT and halve the transmission rate in case of packet losses. Since the space links have very long propagation delays, the link may not be fully utilized during additive transmission rate increase with RTT-granularity.

The equation-based rate control schemes [11, 14, 15] are proposed in order to provide relatively smooth congestion control for UDP traffic in the terrestrial networks. The idea of the equation-based congestion control is to adjust the transmission rate no more than the throughput of the corresponding TCP counterpart with the same packet loss rate, round-trip time, and packet size. TFRC (TCP Friendly Rate Control) [14] is an equation-based rate control scheme which adopts a simple TCP throughput model in its congestion control mechanism. MPEG-TFRC (TCP Friendly Rate Control Protocol for MPEG-2 Video Transfer) [15] is another equation-based rate control scheme designed for transporting MPEG-2 video in a TCP-friendly manner. Unlike TFRC, TFRC takes video characteristics into consideration while adjusting its media rate. Although the use of TCP response function ensures that equation-based control schemes competes fairly with TCP over long-time scales, the steady-state throughput model of TCP source is highly sensitive to RTT values. Therefore, the equation-based rate control schemes cannot achieve high-link utilization and hence are not promising solutions for InterPlaNetary Internet with extremely long propagation delay links.

The throughput performance of SCTP [9], RAP [13], TFRC [14], and RCS [12] are shown in Figure 2 over a 10 Mb/s InterPlaNetary backbone link. The RTT value ranges from 100 to

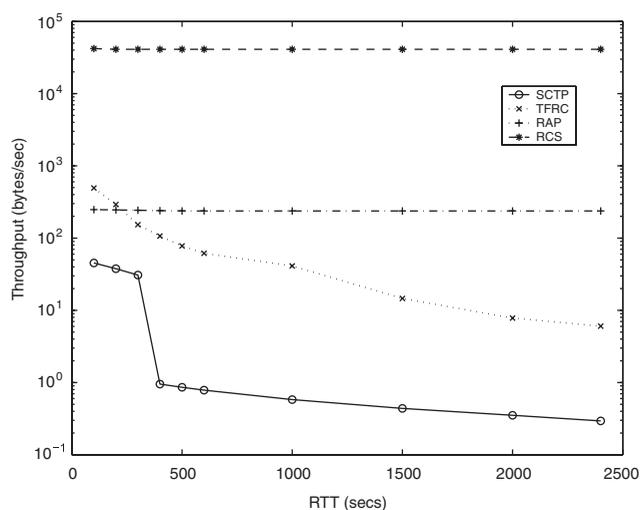


Figure 2. Throughput performance of existing rate control schemes for very high RTT ranges.

2400 s including the RTT range for Mars–Earth communication, i.e. 8.5–40 min based on the orbital location of the planets. Note that the throughput axis is plotted in logarithmic scale in order to be able to include the throughput curves of all protocols. The throughput achieved by TFRC [14] and Sctp [9] is below 100 B/s and it is 237 B/s for RAP [13]. Obviously, the throughput is very low and the entire link remains almost unutilized. Although RCS [12] outperforms other schemes, it can only utilize 41 kB/s of the 10 Mb/s link for 40 min round-trip time, thus, the performance degradation is still very serious. More detailed performance evaluation is presented in [16]. The reason behind is that Sctp, RAP, TFRC, and RCS are sensitive to propagation delays. Sctp is window-based and adopt TCP-like mechanisms. Since TCP protocols perform very badly for very long propagation delays [17], thus, the throughput Sctp achieved is very low. TFRC adopts the steady-state model of TCP, which is very sensitive to the propagation delay, and RAP's slope of linear increase of the transmission rate is inversely related to the propagation delay [13], for the similar reason, they cannot achieve high throughput in InterPlaNetary Internet. For RCS, its *initial state* cannot address long propagation delay and also it adjusts its transmission rate in RTT-grain, although it outperforms other schemes, but still cannot utilize the link bandwidth.

SCPS Rate-based protocol is proposed for space communication [18], but without a congestion control algorithm. The transmission rate in SCPS Rate-based protocol is defined by the user and also constrained by the receiver buffer size. In other words, SCPS Rate-based protocol does not adapt its transmission rate to the network conditions. Thus, it may cause congestion for InterPlaNetary Internet backbone links if its transmission rate is higher than the available bandwidth.

Consequently, the existing rate control schemes cannot address the challenges posed by the InterPlaNetary Internet Backbone Network. New rate control protocols should be proposed in InterPlaNetary Internet to address all the discussed challenges.

### 3. RCP-PLANET: BEGIN STATE

RCP-Planet consists of two states, i.e. *Begin State* and *Operational State*, as shown in Figure 3.

RCP-Planet starts a connection in the *Begin State* at time  $t = 0$  by calling `Begin_State()` algorithm as shown in Figure 11 and goes to the *Operational State* by calling `Operational_State()` algorithm at  $t = \text{RTT}$  as shown in Figure 12.

During the first RTT, no knowledge of the network is available. In order not to waste bandwidth for a duration of one RTT, which is extremely long in the InterPlaNetary backbone link, RCP-Planet determines the initial source-sending rate in a conservative and controlled manner so as to reduce the chances of congestion.

Due to the extremely long propagation delay in the InterPlaNetary Backbone Network, retransmissions of data packets cause high overhead. To address the error control problem, packet-level FEC is used for forward error correction. However, the packet loss rate is also unknown in the *Begin State*. Thus, the most recent history value of the packet loss rate is used to determine the number of redundancy and extra redundancy is added in order to address the possible worse network conditions.

In order to capture the available network bandwidth and increase the transmission rate fast and smoothly, a new rate probing mechanism is introduced in RCP-Planet and is used in both *Begin State* and *Operational State*. The basic idea is to send a number of NIL packets for a period of time to capture the available bandwidth during that period. The NIL packet is a new type of low-priority segments, which was used to probe the availability of network resources as well as error recovery [19, 20]. We update the source-sending rate every FEC block, the captured bandwidth reveals the available bandwidth for the entire FEC block. In addition, the sending rate of NIL packets is chosen properly so that the combined rate with the current source-sending rate during that period is the target rate. In other words, NIL packets only try to capture the available bandwidth up to the target rate.

#### 3.1. Packet-level FEC

Packet-level FEC has been widely used for communication networks [21, 22]. An important factor for the packet-level FEC is the encoding and decoding times. The traditional FEC schemes such as Reed–Solomon codes have rather slow encoding and decoding times, which limit the FEC block size to a very small number [23] and hence result in high FEC overhead. On the other hand, Tornado codes [23] are based on random bipartite graphs and exclusive or operations, thus, they are orders of magnitude faster than the standard erasure codes, which

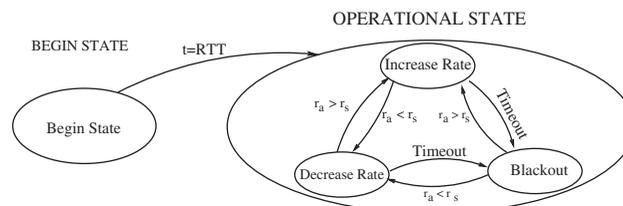


Figure 3. RCP-Planet operation state diagram.

makes Tornado codes appropriate for packet-level FEC with large FEC block size. For example, for data size of 250 kB, the encoding and decoding times of Tornado codes can be as small as 0.06 s. Although Tornado codes require slightly more encoding packets to reconstruct the original data, this disadvantage is compensated by the larger FEC block size, hence the lower FEC overhead. Furthermore, Tornado codes are simple to implement in practice because they use only exclusive or operations. Consequently, Tornado codes are adopted for packet-level FEC in RCP-Planet.

The encoding and decoding times for Tornado codes are proportional to  $(d + l) \ln(1/\varepsilon)S$  [24], while for Reed–Solomon, the times are  $dS$ , where  $d$  is the number of original data packets in a FEC block and  $l$  is the number of redundant packets,  $S$  is the packet size, and  $\varepsilon$  is the so-called *reception overhead*. Tornado codes require  $k$  packets to recover the FEC block, where  $k$  is defined as

$$k = (1 + \varepsilon)d \quad (1)$$

$\varepsilon$  is a very small number around 0.05 [23].

Packet-level FEC is used in both the Begin State and the Operational State. The structure of packet-level FEC in RCP-Planet is shown in Figure 4. At the sender side, data are encoded into FEC blocks and are transmitted using source-sending rate  $r_s$ .

Whenever the destination receives enough data packets for a FEC block, it decodes the FEC block to recover the lost original data packets. Assume  $n$  is the FEC block length and  $d$  is the original data length. If at least  $k$  out of  $n$  packets are received, the FEC block can be recovered successfully, thus, the lost original packets can be reconstructed and passed to the application layer. If fewer than  $k$  packets are received, the lost original packets cannot be reconstructed.

Let us define the probability of receiving at least  $k$  packets out of a group of  $n$  packets as  $P(n, k)$ , then,

$$P(n, k) = \sum_{i=k}^n \binom{n}{i} (1-p)^i p^{n-i} \quad (2)$$

If a FEC block can be recovered successfully, it must receive at least  $k$  out of  $n$  packets, where  $p$  is the packet loss rate. For a given packet loss rate  $p$ , in order to recover a FEC block successfully, the FEC block length  $n$  must satisfy

$$P(n, k) > D \quad (3)$$

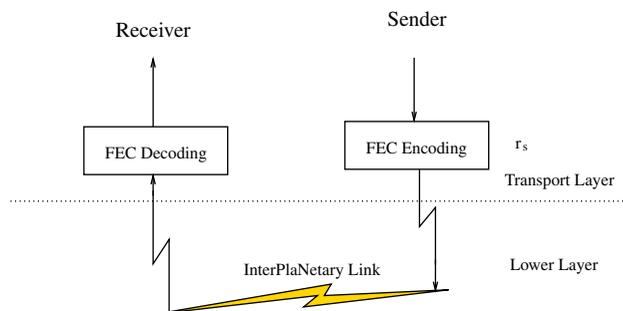


Figure 4. The structure of packet-level FEC.

where  $D$  is a constant smaller than but close to 1. In this paper, we choose  $D = 0.999$ .  $n$  can be calculated online by the recurrence relation

$$P(n, k) = P(n - 1, k) + \binom{i - 1}{n - 1} (1 - p)^k p^{n-k} \quad (4)$$

with the initial condition

$$P(k, k) = (1 - p)^k \quad (5)$$

For a given  $n$ , the overhead  $h$  for packet-level FEC is

$$h = \frac{n - d}{n} \quad (6)$$

The FEC block length should be chosen appropriately to minimize the FEC overhead  $h$ . The FEC overhead  $h$  for packet loss rates range from  $10^{-5}$  to  $10^{-1}$  is shown in Figure 5.

Obviously, the overhead  $h$  decreases with increasing original data length  $d$ , but  $h$  increases with increasing packet loss rate  $p$ . However,  $d$  should not be too large, because larger  $d$  results in longer encoding and decoding times. As a result,  $d$  should be chosen appropriately. In our simulations,  $d$  is chosen to be 86 packets as discussed in Section 5.1.

### 3.2. The rate-probing mechanism

Rate probing is a mechanism to measure the observed rate at the receiver to determine the available bandwidth. The probing techniques include one-packet, packet-pair, and multipacket methods [25]. WTCP [26] sends two back-to-back packets only during connection establishment and use their inter-packet delay as an approximate estimate of the transmission rate. However, two back-to-back packets may not be accurate enough to measure the available bandwidth. This problem can be worse in the backbone links of InterPlaNetary Internet, because long period of congestion can occur if the transmission rate, which is set based on the inaccurate information,

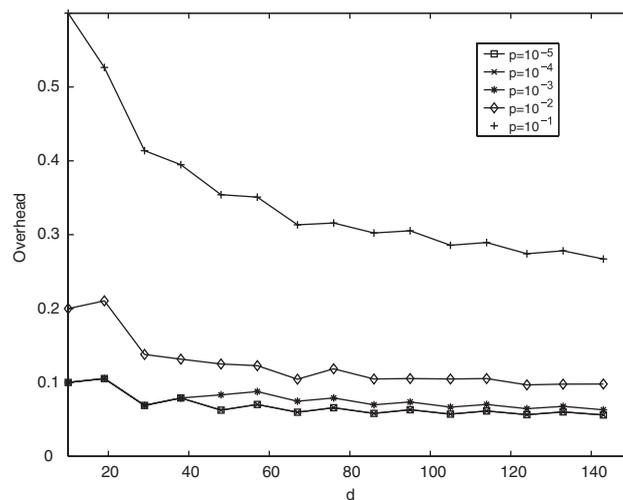


Figure 5. The overhead of FEC vs packet loss rate  $p$ .

is higher than the actual available bandwidth. TCP-Real [27] transmits packets in waves, i.e. a number of packets are sent back to back. The observed rate at the receiver side is compared with the previous observed rate to update the transmission rate by changing the number of packets in a wave. Since all packets are transmitted in the pattern of waves, i.e. back to back, this method might keep creating instantaneous bursty traffic in the network as the wave size increases and hence congest other traffic in the InterPlaNetary Internet. Multimedia traffic usually has relatively high transmission rate, which leads to a very large wave size and makes the problem worse. As a result, TCP-Real [27] is not suitable for InterPlaNetary Internet. The TOPP methods [28] extend the packet pair probing technique by sending carefully spaced probing packets rather than back-to-back packets, however, the probing packets still affect the regular data packets.

In RCP-Planet, we propose a novel rate-probing scheme to capture the available network bandwidth. Our rate-probing scheme is performed in each FEC block, i.e. for each FEC block, we first send a number of NIL packets and keep sending the data packets at its current source-sending rate  $r_s$  so that the combined sending rate  $r_p$  is equal to the target rate  $r_t$  for a time period  $T_p$ .

In our design, we always try to send a fixed number of NIL packets  $L$  so that it is easy to control the overhead. There are two cases, the first one is that the number of regular data packets for the period  $T_p$  denoted as  $L_p$  is smaller than the FEC block length  $n$  as shown in Figure 6. i.e.

$$r_t = \frac{L + L_p}{T_p} \quad (7)$$

$$r_s = \frac{L_p}{T_p} \quad (8)$$

thus,

$$L_p = \left\lceil \frac{L}{r_t/r_s - 1} \right\rceil \quad (9)$$

The second case is  $L_p = n$  as shown in Figure 7.

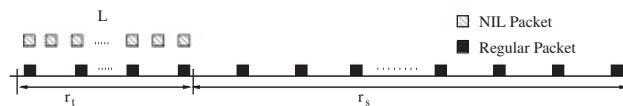


Figure 6. The rate probing for  $L_p < n$ .

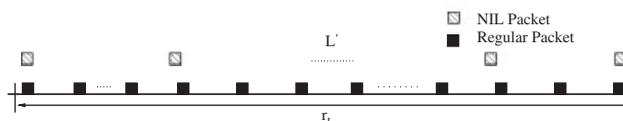


Figure 7. The rate probing for  $L_p \geq n$ .

Denote the discriminant of the two cases as  $\delta$ , we have

$$\delta = \left\lceil \left( \frac{r_t}{r_s} - 1 \right) n \right\rceil \quad (10)$$

To avoid the combined rate  $r_p > r_t$  we need to send less number of NIL packets. Assume the number of NIL packets that we actually send is  $L_a$ , we have

$$r_t = \frac{L_a + n}{T_p} \quad (11)$$

$$r_s = \frac{n}{T_p} \quad (12)$$

and

$$L_a = \left\lceil \left( \frac{r_t}{r_s} - 1 \right) n \right\rceil \quad (13)$$

i.e.  $L_a = \delta$ .

The rate-probing algorithm is summarized as shown in Figure 8.

The NIL packets are extra redundant packets. To generate the NIL packets, we add additional redundant packets  $L$  to the FEC block. Some NIL packets can be dropped by the gateway due to the limit of network bandwidth, as a result, the observed rate, i.e. the probed rate  $r_a$  at the receiver is assumed to be the available bandwidth. The regular data packets during the period  $T_p$  are marked in the header so that the receiver can calculate the observed rate at the receiver side by dividing the number of received NIL packets and data packets whose headers are marked by the duration of the probing sequence measured at the receiver. The receiver reports this information by sending a message packet back to the sender. For convenience, such message packet is also called an ACK here, but it is different from the ACK in TCP protocols. The message packet is the same size as the ACK packet in TCP protocols and it carries the value of  $r_a$  and current packet loss rate in the body of the packet.

The number of NIL packets  $L$  is a design parameter for the rate-probing mechanism. If  $L$  is too small, the observed rate  $r_a$  might not be accurate enough to capture the available bandwidth. On the other hand, if  $L$  becomes larger, the overhead also becomes higher. How to choose the appropriate value of  $L$  is discussed in Section 5.2.

```

Rate_Probing()
  For each FEC block
    Calculate  $\delta$  by Equation (10);
    If  $\delta \geq L$ 
       $L_a = L$ ;
       $L_p$  is calculated by Equation (9);
    Else
       $L_a = \delta$ ;
       $L_p = n$ ;
    End
    Mark the headers of the  $L_p$  data packets;
  End
End

```

Figure 8. The rate-probing algorithm.

The receiver calculates the observed rate  $r_a$  and the sender obtains the observed rate  $r_a$  from ACKs.

### 3.3. The Begin State algorithm

Since the packet loss rate is unknown in the Begin State, the most recent history value  $p_h$ , which is an approximation of the current packet loss rate, is first used to determine the FEC block length  $n$ . The procedure to determine the FEC block length is described in Section 3.1. However, the actual packet loss rate might not be exactly the same as  $p_h$ . In order to address the worse network conditions, a much higher packet loss rate  $p_l$  is conservatively chosen to calculate the actual number of redundancy. Assume the corresponding FEC block length is  $n'$ , then  $n'$  is used as the actual FEC block length to encode the data as shown in Figure 9.

The  $n' - n$  redundant packets are additional overhead to address the possible worse network conditions and they are used as NIL packets, because the low-priority packets are dropped first during congestions and thus, they do not affect regular data traffic during congestions. The remaining  $n$  packets are transmitted in high priority.

The low-priority packet can be identified by one of the eight bits of the TOS field in the IP header and more recent IP versions, e.g. IPv6 [29], explicitly provides several priority levels [20, 30]. For Mars–Earth communications, the gateway on the Mars surface, Mars orbiters, Earth satellites, and the gateway on the Earth surface can have routing capability and they are assumed to be able to identify low-priority packets. If they do not have such capability, the  $n' - n$  redundant packets are treated as regular data packets.

Since no network information is available in the Begin State, it is difficult to determine the initial transmission rate and the number of redundancy for the FEC block. The initial source-sending rate should be set conservatively in order not to inject too many packets into the network, because a very long period of congestion can be incurred if the initial transmission rate is higher than the available bandwidth. The source-sending rate  $r_s$  in the Begin State is designed as shown in Figure 10.

The Begin State consists of two phases, i.e. the exponentially increase phase and the linearly increase phase. The basic idea is to exponentially increase the rate during the first phase until the rate reaches one half of the target rate  $r_t$ . For the second phase, the rate is increased linearly to

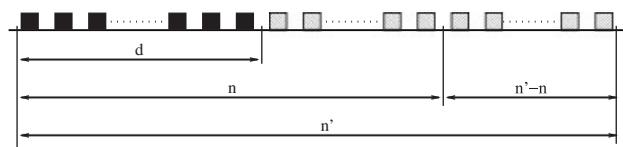


Figure 9. The FEC block in Begin State.

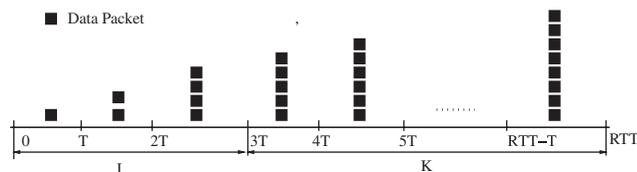


Figure 10. The source-sending rate in Begin State.

the target rate  $r_t$ . The reason to choose the one half of the target rate for the first phase is try to simulate TCP behaviour since TCP usually drops its sending rate to a half during congestion.

The initial time period RTT is divided into  $N$  time slots. The first phase has  $J$  time slots. Assume the rate increase factor is  $\Delta R$ . For the first phase, the source-sending rate for the  $i$ th step is

$$r_s = \Delta R 2^{i-1} \quad (14)$$

At the end of the first phase

$$\frac{r_t}{2} = \Delta R 2^{J-1} \quad (15)$$

For the second phase, the rate is linearly increased

$$r_s = r_s + \Delta R \quad (16)$$

at the end of the second phase, the rate reaches the target rate  $r_t$

$$r_t = \frac{r_t}{2} + K \Delta R \quad (17)$$

By Equation (15) and (17), we have

$$K = 2^{J-1} \quad (18)$$

and

$$N = J + 2^{J-1} \quad (19)$$

thus, the time slot length  $\Delta T$

$$\Delta T = \frac{\text{RTT}}{J + 2^{J-1}} \quad (20)$$

and the rate increase factor  $\Delta R$  is

$$\Delta R = \frac{r_t}{2^J} \quad (21)$$

To decide  $J$ , we first calculate the delay and target rate product

$$P = \text{RTT} r_t \quad (22)$$

and let

$$J = \lceil \log_{10} P \rceil \quad (23)$$

The rate-probing mechanism introduced in Section 3.2 is used in the Begin State, i.e. a probing sequence is included in each FEC block. The probing rate  $r_p$  is set to be  $r_t$  as discussed above so as to capture the available bandwidth as fast as possible.

The Begin State algorithm is summarized in Figure 11.

#### 4. RCP-PLANET: OPERATIONAL STATE

The sender leaves the Begin State for the Operational State at  $t = \text{RTT}$  and remains in the Operational State until the connection is terminated. The Operational State consists of three states, i.e. *increase rate*, *decrease rate*, and *blackout* as shown in Figure 3.

```

Begin_State()
  Calculate FEC block length  $n$  using  $p_h$ ;
  Calculate FEC block length  $n'$  using  $p_l$ ;
  If  $n' < L$ , let  $n' = L$ ;
  Encode the data using  $n'$  as the actual FEC block length;
  Calculate  $J$ ,  $\Delta T$ , and  $\Delta R$  by Equation (23), (20), (21);
  Set the initial rate as  $\Delta R$ ;
  While ( $t \leq RTT$ )
    For each rate update step  $i$ 
      If  $i \leq J$ 
         $r_s = \Delta R 2^{i-1}$ ;
      Else
         $r_s = \frac{r_t}{2} + (i - J) \Delta R$ ;
      End;
    End
  For each FEC block
    Call Rate_Probing() to determine  $L_a$  and  $L_p$ ;
    Send NIL packets using rate  $r_t - r_s$ ;
    Send regular data packets using rate  $r_s$ ;
  End
End
End

```

Figure 11. The Begin State algorithm.

In the Operational State, the sender goes to increase rate state or decrease rate state based on the probed available network bandwidth. Based on the rate-probing mechanism discussed in Section 3.2, a new rate control scheme is proposed in RCP-Planet to increase the source-sending rate fast and smoothly in order to address the extremely long propagation delay in the InterPlaNetary backbone link. RCP-Planet also incorporates the Blackout State into the protocol operation in order to reduce the throughput degradation due to blackouts. Moreover, the bandwidth asymmetry problem is addressed by FEC block-level ACKs in the Operational State.

#### 4.1. The new rate-control scheme

Packet-level FEC discussed in Section 3.1 is also used in the Operational State to address the error control problem. The RCP-Planet sender uses the probing mechanism discussed in Section 3.2 to capture the available network bandwidth and updates the probing rate adaptively based on the network conditions. Once received the probing sequence, the receiver reports the observed rate  $r_a$  to the sender by putting this information into the ACK.

Upon receiving an ACK from the receiver, the sender obtains the current probed rate  $r_{a,i+1}$ , which reveals the available network bandwidth for RCP-Planet. Obviously,  $r_{a,i+1}$  is the current upper bound for the source-sending rate. Since the probing rate  $r_p$  is set to be  $r_t$ , thus,  $r_{a,i+1} \leq r_t$ .

If  $r_{a,i+1} \geq r_{s,i}$ , where  $r_{s,i}$  is the current source-sending rate, the network bandwidth is not fully utilized and the source-sending rate should be increased. However, the source-sending rate should not be increased by the amount  $(r_{a,i+1} - r_{s,i})$  at once. The reasons are as follows:

- The available bandwidth might be shared by multiple connections, one RCP-Planet connection should not be too aggressive to take all the available bandwidth at once.

- The feedback of the rate change is only available after one RTT. Due to the extremely long propagation delay, the source-sending rate should be increased slowly in order to decrease the chances of congestions in the network.

Due to the above reasons, the extra amount  $(r_{a,i+1} - r_{s,i})$  is increased in one RTT linearly with respect to time. Thus, the next source-sending rate  $r_{s,i+1}$  is

$$r_{s,i+1} = r_{s,i} + \frac{r_{a,i+1} - r_{s,i}}{\text{RTT}} \Delta t \quad (24)$$

where  $\Delta t$  is the time to transmit the next FEC block

$$\Delta t = \frac{n}{r_{s,i+1}} \quad (25)$$

By combining Equation (24) and (25), we get

$$r_{s,i+1} = \frac{1}{2} \left[ r_{s,i} + \sqrt{r_{s,i}^2 + \frac{4n(r_{a,i+1} - r_{s,i})}{\text{RTT}}} \right] \quad (26)$$

Obviously,  $r_{s,i+1}$  increases very fast, if  $(r_{a,i+1} - r_{s,i})$  is large. However,  $r_{s,i+1}$  increases slowly, if  $r_{a,i+1}$  is close to  $r_{s,i}$ . This is a conservative behaviour to reach the available bandwidth. If  $r_{a,i+1} = r_{s,i}$ , then  $r_{s,i+1} = r_{s,i}$ , i.e. the sender holds its rate after it reaches the available bandwidth.

On the other hand, if  $r_{a,i+1} < r_{s,i}$ , i.e. the current rate is too high, the sender needs to back up and to decrease its rate. The rate is then decreased multiplicatively

$$r_{s,i+1} = \beta r_{s,i} \quad (27)$$

where  $\beta$  is the rate-decreasing factor and  $0 < \beta < 1$ .

The Operational State algorithm is summarized in Figure 12.

#### 4.2. The blackout state behaviour

Link outages due to loss of line of sight by orbital obscurations lead to burst packet losses and decrease in the throughput. In order to reduce the throughput degradation due to blackouts, *Blackout State* is developed and incorporated into the Operational State.

The sender starts to receive ACKs from the receiver after one RTT time. If it does not receive any ACKs for a certain period of time  $T_w$ , it infers this condition as blackout and goes to the *Blackout State*. During blackout, the sender stops sending any packets because power efficiency is critical in InterPlaNetary Internet.

The receiver also infers blackout after not receiving any packets from the sender for a certain period of time  $T_w$ . Then it starts to transmit ACKs with  $r_a = 0$  and  $p = 1$  periodically, which are called *Zero ACKs*. The objective of *Zero ACKs* is to help the sender to capture accurate information regarding the blackout situation and act accordingly.

Since RTT is very high, the effect of blackout on the performance changes with its relative location of blackout occurrence with respect to the receiver. Let  $t = t_0$  be the time when blackout occurs and  $B$  is the duration of the blackout. Assume that the blackout occurs at a position  $x$  s away from the RCP-Planet receiver, i.e. the propagation delay from the blackout location to the RCP-Planet receiver is  $x$  s. For  $\text{ott} = \text{RTT}/2$ , there are two distinct cases

```

Operational_State()
  Encode FEC block using Tornado Codes;
  For each FEC block
    Call Rate_Probing() to determine  $L_a$  and  $L_p$ ;
    Send NIL packets using rate  $r_t - r_s$ ;
    Send regular data packets using rate  $r_s$ ;
  End
  If (ACK_RECEIVED)
    Calculate FEC block length based on  $p$ ;
  Rate Control:
  If ( $r_a \geq r_s$ )
    /* Increase Rate */
    Update  $r_s$  by (26);
  Else
    /* Decrease Rate */
    Update  $r_s$  by (27);
  End
  If (NO_ACK_in_ $T_w$ )
    /* Blackout State */
    If (current sub-state =  $S_0$ )
      current sub-state =  $S_1$ ;
      Stop sending packet;
    End
  If (NORMAL_ACK_RECEIVED)
    If (current sub-state =  $S_1$ )
      /*  $B < 2x$  */
      current sub-state =  $S_2$ ;
      Update  $r_s$ ;
      Send packets using  $r_s$ ;
    End
    If (current sub-state =  $S_3$ )
      current sub-state =  $S_0$ , Blackout State is over;
      Update  $r_s$ ;
      Send packets using  $r_s$ ;
    End
  End
  If (ZERO_ACK_RECEIVED)
    If (current sub-state =  $S_1$ )
      /*  $B \geq 2x$  */
      current sub-state =  $S_3$ ;
      Send packets using  $r_s$ ;
    End
    If (current sub-state =  $S_2$ )
      /*  $B > 2x$  */
      current sub-state =  $S_3$ ;
      Send packets using  $r_s$ ;
    End
  End
End

```

Figure 12. The Operational State algorithm.

observed at the sender side according to the duration of the blackout and its relative distance to the receiver in time:

- $B < 2x$ : After  $ott - x$  from  $x_0$ , i.e. at  $t_1$ , RCP-Planet sender detects the period without ACKs. If the duration of this period with no ACKs takes more than  $T_w$ , then the sender moves to Blackout States at  $t = t_1$  as in Figure 13. After the period of time  $B$ , the sender starts to receive ACKs, which are on-fly ACKs when blackout occurs. At  $t_3 = t_2 + 2x - B$ ,

the source starts to receive Zero ACKs. After time  $B$ , the sender starts to receive ACKs and it goes to either the increase rate or decrease rate state based on the network conditions.

- $B \geq 2x$ : In this case, RCP-Planet sender detects no ACK period and goes to Blackout State at  $t_1 = t_0 + \text{ott} - x$  for the blackout occurred at  $t = t_0$ . At  $t_2 = t_1 + B$ , the sender starts to receive Zero ACKs for duration  $2x$  as shown in Figure 14. After that, the sender starts to receive ACKs and the Blackout State is over.

Consequently, Zero ACKs and the on-fly ACKs when blackout occurs are used for the sender to capture the accurate information regarding the blackout situation and to prevent misprediction of blackout, which is novel over the existing approaches. For example, in Figure 13, the interval  $t_3-t_4$  can be mispredicted as another blackout period without Zero ACKs. Also in Figure 14, the blackout period can be detected as  $B + 2x$  if Zero ACKs are not used.

The operation state diagram of the Blackout state is shown in Figure 15, where state  $S_0$  is the period of  $t_0$  to  $t_1$  in Figures 13 and 14. State  $S_1$  is the blackout period and state  $S_2$  corresponds to the period  $t_2-t_3$  in Figure 13. State  $S_3$  is the Zero ACK period.

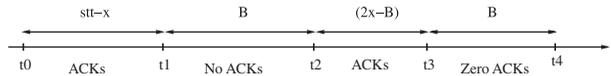


Figure 13. Blackout condition observed from RCP-Planet sender for  $B < 2x$ .

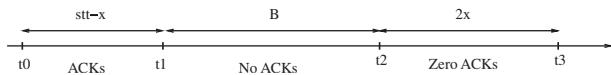


Figure 14. Blackout condition observed from RCP-Planet sender for  $B \geq 2x$ .

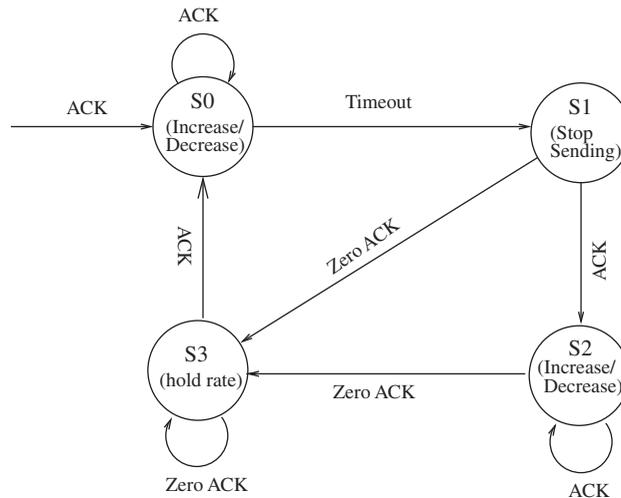


Figure 15. Blackout operation state diagram.

During  $S_1$ , the blackout is detected by the timeout mechanism and the sender stops sending packets. Then it goes to either  $S_2$  upon receiving an ACK or to  $S_3$  upon receiving a Zero ACK. In  $S_2$  and  $S_3$ , the sender resumes sending packets. In  $S_2$ , the sender either increases or decreases its sending rate based on the feedback information from the ACKs. While in  $S_3$ , the sender holds its rate. Upon receiving an ACK in  $S_3$ , the sender goes to  $S_0$  and the Blackout State is over.

Consequently, the Blackout State reduces the throughput degradation due to blackout conditions and improves the link utilization for duration of  $B$  or  $2x$  in the cases  $B < 2x$  and  $B \geq 2x$ , respectively. Furthermore, InterPlaNetary backbone links usually have intermittent connectivity within a round-trip time period, which can also be addressed by the Blackout State. Since power efficiency is critical for InterPlaNetary Internet, especially on the Mars surface, the sender stops sending any packets after it infers blackout situation and it only resumes transmitting packets after it receives ACKs from the receiver.

#### 4.3. Bandwidth asymmetry

RCP-Planet receiver needs to send message packets, which include the observed rate for a probing sequence and the packet loss rate of a FEC block, back to the sender so that the sender can adjust its transmission rate and the amount of FEC redundancy accordingly. Since the InterPlaNetary backbone links are usually asymmetrical in the order of 1000:1 or more [5], too many message packets can cause congestions in the reverse channel.

In RCP-Planet, FEC block-level ACK is used, i.e. only one ACK is sent for an entire FEC block, which includes the observed rate and the current packet loss rate. If the FEC block size is large enough, the bandwidth asymmetry problem can be solved by the FEC block-level ACKs. Delayed ACKs can also be used to further reduce the number of ACKs in the reverse link, i.e. only sends one ACK for a certain number of FEC blocks. In this case, the observed rate and the current packet loss rate are the average values over multiple FEC blocks.

The bandwidth asymmetry factor  $f$  is defined to measure the ratio of the traffic in the forward and reverse channels for an RCP-Planet connection, i.e.

$$f = \frac{N_d S}{N_a A} \quad (28)$$

where  $N_d$  is the number of packets received at the receiver for a period of time,  $N_a$  the number of ACKs sent by the receiver for the same duration,  $S$  packet size and  $A$  ACK size.

$f$  is a measure to illustrate the traffic ratio in the forward and reverse channels for RCP-Planet. If the bandwidth asymmetry is smaller than  $f$ , RCP-Planet will not cause congestions in the reverse link, i.e. the bandwidth asymmetry problem is solved for the bandwidth asymmetry ratio up to  $f$ .

## 5. PERFORMANCE EVALUATION

We conducted extensive simulation experiments to investigate the performance of RCP-Planet. How to choose the appropriate probing sequence length  $L$  is discussed in Section 5.2. The source-sending rate is illustrated in Section 5.3. Throughput performance of RCP-Planet is analysed along with the overhead, FEC block recovery rate, and fairness in Sections 5.4–5.7,

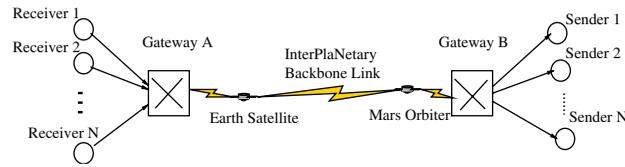


Figure 16. Simulation scenario.

respectively. Bandwidth asymmetry is discussed in Section 5.8. Finally, the blackout performance is analysed in Section 5.9.

### 5.1. Simulation scenario

The simulation scenario is shown in Figure 16.  $N$  RCP-Planet senders on Mars transmit data to  $N$  receivers on Earth. The data are first transmitted from gateway B on the Mars surface to the Mars orbiter, then through the InterPlaNetary backbone link to the Earth satellite, and finally arrives at the gateway A on the Earth surface. The feedback message is transmitted on the reverse links. The  $N$  data flows are multiplexed in gateway A on the Mars surface and the reverse data flows in gateway B on the Earth surface. Segments in forward and reverse channels may get lost due to link errors with a probability  $p_{\text{loss}}$ . If not specified, we assume  $N = 10$ , the gateway buffer size is 200 packets. We also assume that the link capacity is  $c = 1300$  packets/s, which is approximately 10 Mb/s for a data packet of size 1 kB. The target rate is assumed to be 140 kB/s unless otherwise stated

In the Begin State, we assume the history value of packet loss rate  $p_h$  is  $10^{-4}$  and the much larger packet loss rate  $p_l = 10^{-1}$ .

As shown in Figure 5,  $d = 86$  packets is a suboptimal point, the variance of the FEC overhead for  $86 \leq d < 150$  is only 0.7% for  $p_l < 10^{-1}$  and 2.8% for  $p_l = 10^{-1}$ . Since usually  $p_l < 10^{-1}$ , the FEC overhead cannot be reduced greatly for  $d > 86$  packets. On the other hand, higher  $d$  incurs higher encoding and decoding times of the FEC block. As a result, the original data length  $d$  is chosen to be 86 packets in the simulations.

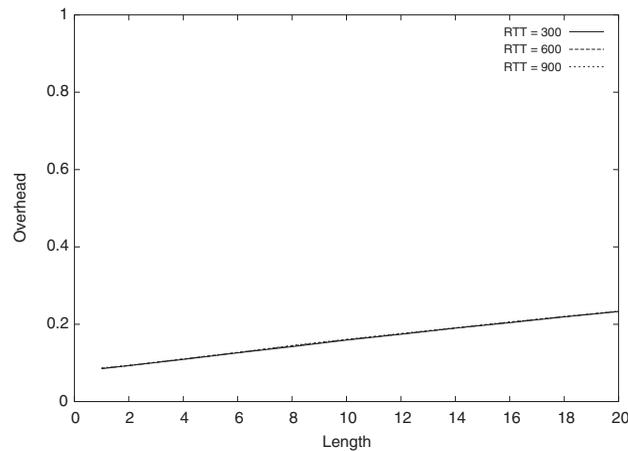
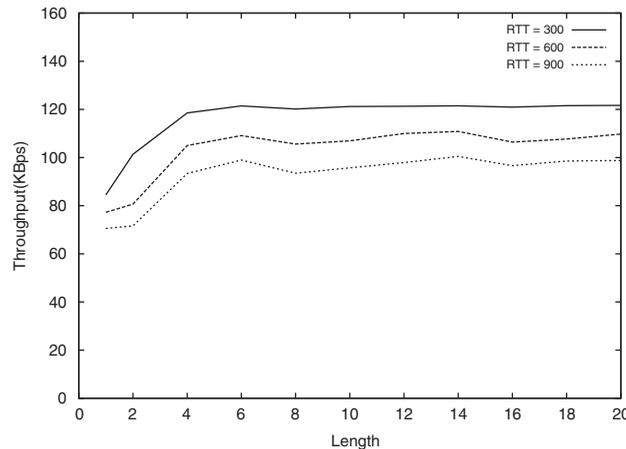
The blackout threshold  $T_w$  is set to be  $4n/r_s$ . Since the end-to-end path for the InterPlaNetary Internet is not always available, the communication time cannot last for very long time. Because existing TCP and rate control protocols have very poor performance in InterPlaNetary Internet [16, 17], no TCP and other background traffic are included.

### 5.2. NIL packet number

Rate-probing mechanism presented in Section 3.2 is used to capture available bandwidth in RCP-Planet, the NIL packet number  $L$  should be chosen appropriately to capture the available network bandwidth as accurate and fast as possible and to reduce its overhead.

To investigate how  $L$  affects the overhead and throughput, only one connection is used to eliminate other effects on it from other connections. We assume RTT = 300, 600, and 900 s, respectively. Packet loss rate due to link errors  $p_{\text{loss}} = 10^{-4}$ , and the simulation time is 3600 s. The overhead corresponding to  $L$  is shown in Figure 17.

The overhead for different RTTs are almost the same. With  $L$  increasing from 1 to 20, the overhead increases from 0.085 to 0.234. Obviously, larger  $L$  will lead to higher overhead.

Figure 17. Overhead vs NIL packet number  $L$ .Figure 18. Throughput vs NIL packet number  $L$ .

The throughput is defined as the number of data packets that are recovered successfully from the FEC blocks divided by the run time. The throughput for different RTTs is shown in Figure 18.

When  $L$  is small, the throughput is low. For example, when  $RTT = 600$  and  $L = 1$ , the throughput is only 85.76 kbps. This is because small  $L$  cannot capture the available bandwidth. The throughput increases with increasing  $L$ . But when  $L$  reaches 14, the degree of throughput increase is very small. When  $RTT = 300$ , the throughput increase is only 0.18 kbps from  $L = 14$  to  $L = 20$ , but the overhead increases from 19 to 23.3%.

The FEC recovery rate  $R_{blk}$  is the percentage of FEC blocks that are recovered successfully and it is defined as

$$R_{blk} = \frac{N_r}{N_t} \quad (29)$$

where  $N_r$  is the number of FEC blocks recovered successfully and  $N_t$  is the total number of received FEC blocks. If a FEC block cannot be recovered successfully, the lost original packets cannot be reconstructed and thus, lead to the degradation of the throughput performance.

After the FEC block length  $n$  is calculated from the current packet loss rate  $p$  by Equation (4) as discussed in Section 3.1, we conservatively add  $L$  extra redundant packets to recover the lost probing packets. As a result, RCP-Planet tries to achieve as high  $R_{\text{blk}}$  as possible. When  $L$  increases, the number of extra redundant packets also increases, thus, the FEC block recovery rate also increases.

To achieve good performance, the probing sequence length  $L$  should be appropriately chosen such that RCP-Planet has high FEC block recovery rate, high throughput, and low overhead. Considering these factors, we choose  $L = 14$  packets for all the subsequent simulations.

### 5.3. The probed rate and source-sending rate

To show the behaviour of the probed rate  $r_a$  and source-sending rate  $r_s$ , we assume  $\text{RTT} = 600$  s, packet loss rate due to link errors  $p_{\text{loss}} = 10^{-3}$ , and the simulation time = 3600 s. Other parameters are the same as defined in Section 5.1. The resulting probed rate and source-sending rate are illustrated in Figure 19.

In the Begin State, we conservatively set the source-sending rate  $r_s$  in a controlled manner because no knowledge of the network is available. Thus, during the first RTT period, the RTT and the target rate product  $P = 84\,000$ , from Equation (22),  $J = 5$ . Thus,  $K = 16$ ,  $\Delta T = 28.57$  s, and  $\Delta R = 4.38$  kbps. From Figure 19, in the first phase of the Begin State,  $r_s$  increases exponentially from 4.38 to 70 kbps, i.e. half of the target rate 140 kbps. After time  $t = 142.85$  s, RCP-Planet enters the second phase,  $r_s$  increases linearly and reaches 140 kbps at  $t = 600$  s.

After one RTT, the observed rate  $r_a$  and the current packet loss rate  $p$  become available. Since 10 RCP-Planet connections compete for the bandwidth, congestion occurs after  $t = 514.29$  s. The RCP-Planet sender backs up and decreases its source-sending rate upon receiving the probed rate. We choose the rate decrease factor to be 0.9, so that the rate will not drop too

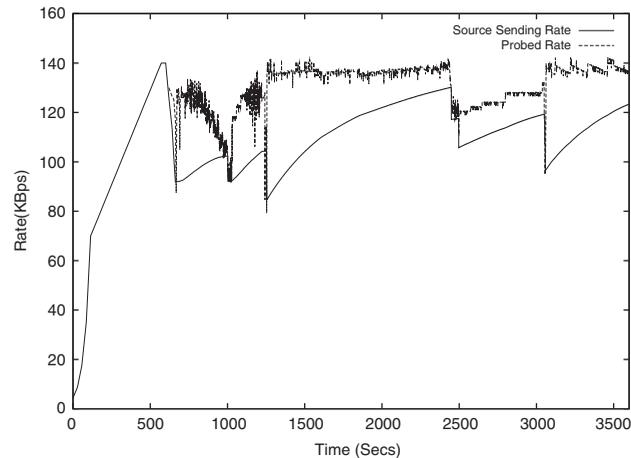


Figure 19. Probed rate and source-sending rate vs time.

sharply. Since the probed rate is not higher than the target rate,  $r_s$  will not capture more bandwidth than the target rate. After the congestion is over, it starts to increase the rate again. Consequently, the source-sending rate is updated according to the network condition, the corresponding average source-sending rate is 121.40 kBps, which is close to the equal share of the bandwidth for each connection, i.e. 130 kBps for link capacity  $c = 1300$  kBps shared by 10 RCP-Planet connections.

#### 5.4. Throughput performance

We use parameters defined in Section 5.1 and assume  $RTT = 300, 600, 900$  s, respectively. The packet loss rate due to link errors  $p_{loss}$  is  $10^{-5}$ – $10^{-2}$ . There are 10 connections and each connection will send 100 MB data. The throughput performance is illustrated in Figure 20. Note that the throughput is calculated only for the original data packets from the application, FEC redundant packets are not included.

RCP-Planet achieves high throughput for different RTT values and packet loss rates due to link errors. The throughput for one individual RCP-Planet connection is around 87 kBps for  $RTT = 300$  s, 74 kBps for  $RTT = 600$  s, and 67 kBps for  $RTT = 900$  s. Considering FEC redundancy is not counted and the simulation time is relatively short due to network constraints and RCP-Planet stays in the Begin state for a quite long time, the achieved throughput is high. The reasons that RCP-Planet achieves high throughput can be summarized as follows:

- The rate probing mechanism discussed in Section 3.2 is used to capture the available bandwidth.
- The source-sending rate is increased in a controlled manner to capture the bandwidth as fast as possible and conservely.
- The new rate control scheme updates the source-sending rate smoothly and conservatively to address the extremely long propagation delay problem.

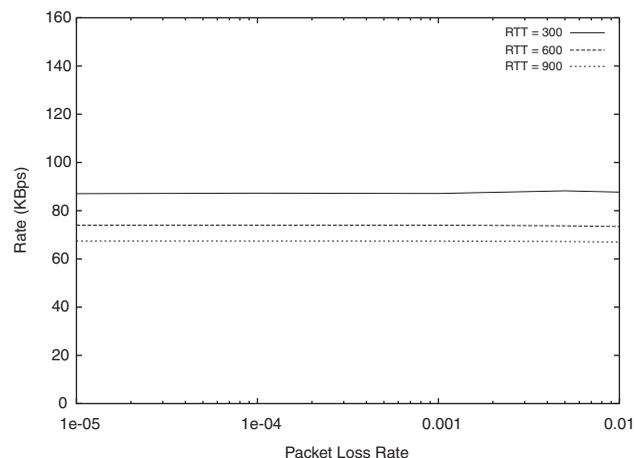


Figure 20. Throughput vs packet loss rate due to link errors.

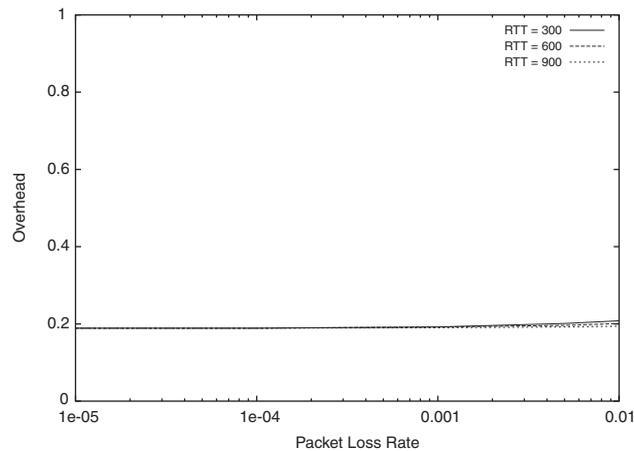


Figure 21. Overhead vs packet loss rate due to link errors.

For a given RTT, the throughput varies slightly for  $p_{\text{loss}}$  in the range of  $10^{-5}$ – $10^{-2}$ . For example, the throughput only decreases about 0.47 kbps from  $p_{\text{loss}} = 10^{-5}$  to  $p_{\text{loss}} = 10^{-2}$  for RTT = 900 s. This reveals that RCP-Planet can recover lost packets due to link errors effectively.

### 5.5. Overhead

We use the same parameters as in Section 5.4 for our simulation. The resulting overhead vs packet loss rate due to link errors is shown in Figure 21. Here, the overhead includes all the redundant packets sent in both high and low-priority.

Overhead is introduced by NIL packets to probe the available bandwidth and the redundant packets to recover packet losses due to link errors and congestions. First, we observe that the overhead is approximately the same for different RTTs and fixed  $p_{\text{loss}}$ . The overhead increases with increasing  $p_{\text{loss}}$ . For RTT = 600 s, it increases from 0.189 for  $p_{\text{loss}} = 10^{-5}$  to 0.201 for  $p_{\text{loss}} = 10^{-2}$ . The reason is that more redundancy is required to recover packet losses due to link errors.

Since packet-level FEC is used in RCP-Planet for the traffic to recover packet losses due to link errors and congestions, the overhead is about 0.189 even when  $p_{\text{loss}} = 10^{-5}$ . This amount of overhead is mainly introduced by the following factors:

- Tornado codes require slightly more packets to recover a FEC block.
- In the Begin State, a much higher packet loss rate  $p_1$  is chosen in order to address the possible worse network conditions. This conservative method can incur extra overhead if the channel is good.
- NIL packets also introduce overhead.

However, this amount of redundancy is quite reasonable for packet-level FEC and is also compensated by the high throughput as discussed in Section 5.4 and the high FEC block recovery rate as discussed in Section 5.6.

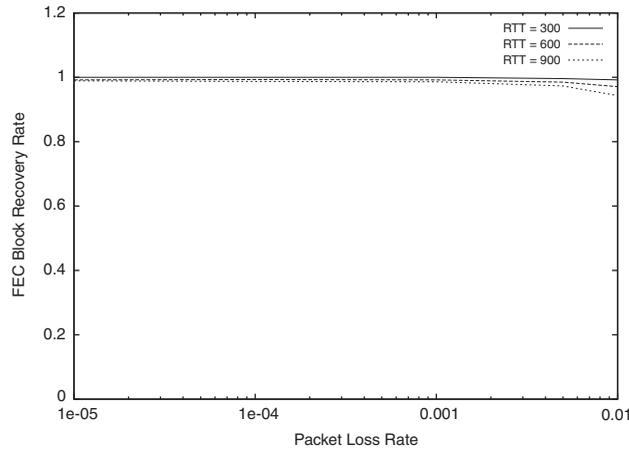


Figure 22. FEC block recovery rate vs packet loss rate due to link errors.

### 5.6. FEC block recovery rate

We use the same parameters as in Section 5.4 for our simulation. The resulting FEC block recovery rate vs packet loss rate due to link errors is shown in Figure 22.

As discussed in Section 5.2, RCP-Planet tries to achieve as high  $R_{\text{blk}}$  as possible. For  $\text{RTT} = 300$  s,  $R_{\text{blk}}$  is 1 for  $p_{\text{loss}} \leq 10^{-3}$ , but drops to 0.992 for  $p_{\text{loss}} = 10^{-2}$  because of high packet loss rate due to link errors. For  $\text{RTT} = 600$  and  $900$  s,  $R_{\text{blk}}$  is also around 0.99. The high FEC block recovery rates that RCP-Planet achieves are mainly due to two reasons, the first one is the smooth update of the source-sending rate, which leads to less congestions and hence less packet losses, and the other is the way to choose FEC redundancy as discussed in Section 5.6. On the other hand,  $R_{\text{blk}}$  almost does not change with RTT increasing from 300 to 900 s. This also illustrates that RCP-Planet is delay-tolerant.

### 5.7. Fairness

Since to the best of our knowledge, no existing rate control scheme has been proposed in InterPlaNetary Internet so far, we only consider homogeneous fairness here, i.e. the fairness of the 10 RCP-Planet connections.

As described in [31], the fairness index based on throughput for a bottleneck link is defined as

$$FI = \frac{[\sum_{i=1}^N T(i)]^2}{N \sum_{i=1}^N T(i)^2} \quad (30)$$

where  $T(i)$  is the throughput of the  $i$ th flow and  $N$  is the number of flows sharing the resource.  $FI$  always lies between  $1/N$  (indicating one of them gets all the bandwidth and all others starve) and 1 (indicating all get an equal share of the bandwidth).

The same parameters in Section 5.4 are used in the simulation. The resulting fairness vs packet loss rate due to link errors is shown in Figure 23.

The resulting fairness in Figure 23 shows that the fairness is approximately 1 for different packet loss rates due to link errors. Consequently, RCP-Planet connection always shares

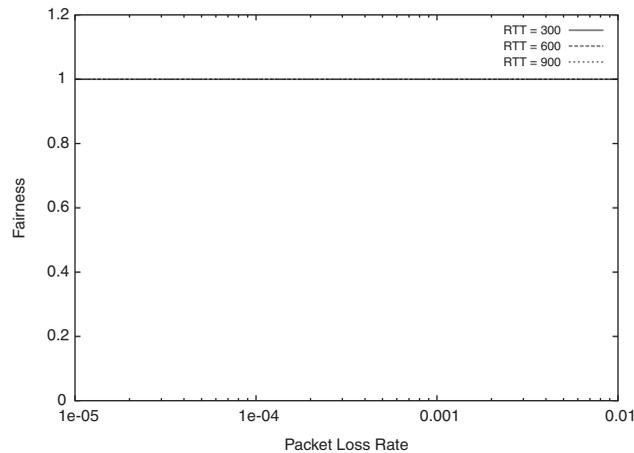


Figure 23. Fairness vs packet loss rate due to link errors.

the available network bandwidth equally for different application requirements and late join flows.

#### 5.8. Bandwidth asymmetry factor

Bandwidth asymmetry factor  $f$  as defined by Equation (28) is introduced in Section 4.3 to measure the ratio of the traffic in the forward and reverse channels for an RCP-Planet connection. The bandwidth asymmetry problem is addressed by block-level ACKs in the RCP-Planet. If the bandwidth asymmetry is smaller than  $f$ , RCP-Planet will not cause congestion in the reverse link, i.e. the bandwidth asymmetry problem is solved for the bandwidth asymmetry ratio up to  $f$ .

The same parameters in Section 5.4 are also used in the simulation. The resulting bandwidth asymmetry factor vs packet loss rate due to link errors is shown in Figure 24.

For  $RTT = 300, 600, 900$  s,  $f$  remains approximately constant at 2152 for different  $p_{\text{loss}}$ . This means that RCP-Planet solves bandwidth asymmetry up to 2152:1 by using FEC block-level ACKs. Since the asymmetry in the bandwidth capacity of forward and reverse channels is typically in the order of 1000:1 in space missions [5], the bandwidth asymmetry ratio 2152:1 is quite high for InterPlaNetary Internet links, hence, RCP-Planet works well in InterPlaNetary Internet with high bandwidth asymmetry. Furthermore, delayed ACKs can also be used to further reduce the number of ACKs in the reverse link as discussed in Section 4.3.

#### 5.9. The blackout performance

When a blackout is detected, RCP-Planet moves to the Blackout State as shown in Figure 3 in order to reduce its effect on the throughput performance as explained in Section 4.2. Throughput achieved by RCP-Planet for different blackout durations is shown in Figure 25, where  $RTT = 600$  s,  $p_{\text{loss}} = 10^{-4}$ , and the blackout occurs at a position 150 s away from the receiver at time  $t = 1200$  s. Throughput is used to measure the performance of RCP-Planet in blackout conditions. In order to eliminate the effect of congestions on the throughput performance, only one RCP-Planet connection is used in the simulation. Simulation time is 3600 s and the other parameters are the same as in Section 5.1.

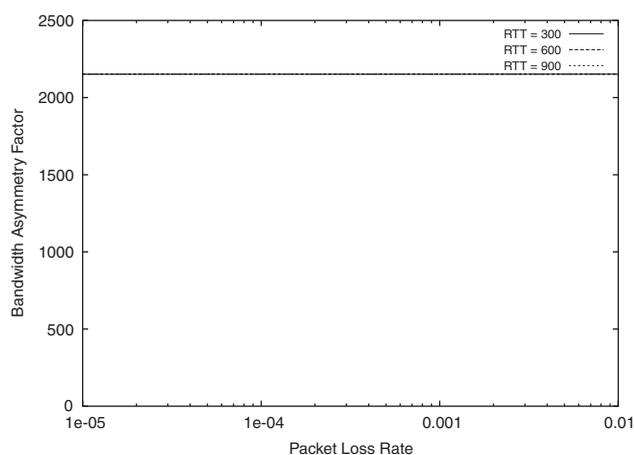


Figure 24. Bandwidth asymmetry factor vs packet loss rate due to link errors.

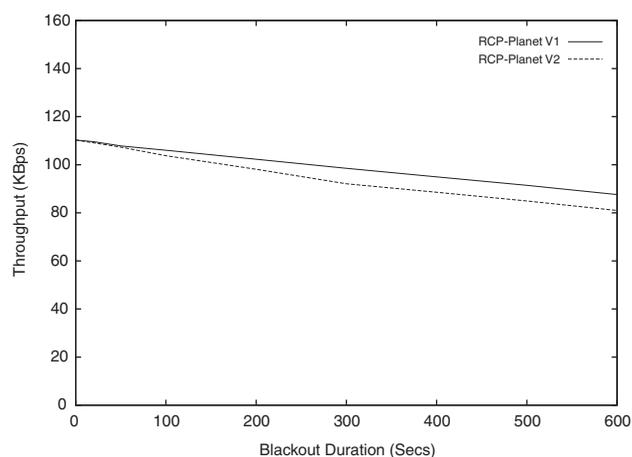


Figure 25. Throughput vs blackout duration.

In order to investigate RCP-Planet performance in blackout conditions, we consider two versions of the RCP-Planet:

- *RCP-Planet V1*: This version incorporates the Blackout State.
- *RCP-Planet V2*: This version does not incorporate the Blackout State. It uses the *link-probing scheme* introduced in SCPS-TP [32] to detect when the blackout is over.

In the link-probing scheme, when the sender detects that a blackout occurs, it sends link-probing segments periodically to the receiver. Upon receiving a link-probing segment, the receiver sends an ACK immediately back to the sender. Once received an ACK for the link-probing segment, the sender infers that the blackout is over and resumes sending data packets.

Although the throughputs of RCP-Planet V1 and V2 both decrease with increasing blackout duration  $B$ , RCP-Planet V1 always outperforms RCP-Planet V2. For  $B \leq 300$  s, i.e.  $B < 2x$ , the throughput difference between RCP-Planet V1 and V2 increases with increasing  $B$ . For example, the throughput difference is 0.45 kBps at  $B = 20$ , but goes up to 6.44 kBps at  $B = 300$ , which agrees with the analysis in Section 4.2, i.e. the gain of the blackout state is proportional to  $B$  for  $B < 2x$ . Since  $B$  increases, the gain also increases.

For  $B \geq 300$  s, i.e.  $B \geq 2x$ , the throughput difference between RCP-Planet V1 and V2 remains approximately constant. For example, the throughput difference is 6.42 kBps at  $B = 400$  s, 6.52 kBps at  $B = 500$  s, and 6.61 kBps at  $B = 600$  s. This also matches our conclusion in Section 4.2, i.e. the gain of the blackout state is proportional to  $2x$  for  $B \geq 2x$ . Thus, the gain remains the same for  $B \geq 2x$ .

## 6. CONCLUSIONS

The rate control problem in InterPlaNetary Backbone Network is very challenging because of *extremely long propagation delays, high link errors, asymmetrical bandwidth, and blackouts*. Due to the lack of rate control protocols in InterPlaNetary Backbone Network, a rate control protocol, RCP-Planet, is proposed to address the challenges of the rate control problem in InterPlaNetary Internet. RCP-Planet consists of two novel algorithms, i.e. *Begin State* and *Operational State*. In the *Begin State*, the source-sending rate and the number of redundancy are determined conservatively so that it can address possible worse network conditions and reduce the chances of congestion. A novel rate-probing mechanism is proposed to capture the available bandwidth. Based on the rate-probing mechanism, the new rate control scheme updates the source-sending rate smoothly and conservatively in the *Operational State*. To recover packet losses due to link errors and congestions, Tornado codes are used for packet-level FEC because of their very fast encoding and decoding times. The FEC block length is chosen appropriately to minimize the FEC overhead. Furthermore, FEC block-level ACKs are used to address bandwidth asymmetry problems and the bandwidth asymmetry factor is introduced to measure up to what degree of bandwidth asymmetry RCP-Planet can solve. Apart from that, the blackout state is incorporated into RCP-Planet to improve the performance in blackout conditions.

Simulation experiments show that RCP-Planet reaches the available rate fast and smoothly using the rate probing mechanism and the new rate control scheme. It achieves high throughput and FEC block recovery rate with reasonable overhead. Multiple RCP-Planet connections can share the available bandwidth equally. The Blackout State in RCP-Planet always outperforms the link-probing scheme introduced in SCPS-TP [32]. Moreover, the simulation results also reveal that RCP-Planet is delay tolerant. As a result, RCP-Planet is a rate-control protocol with diverse set of algorithms and functionalities, which addresses the challenges of rate control in InterPlaNetary Internet.

## REFERENCES

1. Akyildiz IF, Akan O, Chen C, Fang J, Su W. InterPlaNetary Internet: state-of-the-art and research challenges. *Computer Networks Journal* 2003; **43**(2):75–113.
2. Travis E. The InterPlaNetary Internet: architecture and key technical concepts. *Presented at the Internet Global Summit*, INET, June 2001.

3. Bhasin K, Hayden J, Agre JR, Clare LP, Yan TY. Advanced communication and networking technologies for Mars exploration. *19th Annual AIAA International Communications Satellite Systems Conference*, Toulouse, France, 17–20 April 2001.
4. Akan OB, Fang J, Akyildiz IF. TP-Planet: a new transport protocol for deep space networks. *IEEE Journal of Selected Areas in Communications (JSAC)* 2004; **22**(2):348–361.
5. Durst RC, Feighery PD, Scott KL. Why not use the standard Internet suite for the InterPlaNetary Internet? <http://www.ipnsig.org/techinfo.htm>
6. Hassan S, Kara M. Performance evaluation of end-to-end TCP-friendly video transfer in the Internet. *The Ninth IEEE International Conference on Networks (ICON'01)*, Bangkok, Thailand, October 2001.
7. Kuhmunch C, Kuhne G. Efficient video transport over lossy networks. *Technical Report TR-98-798*, University of Mannheim, April 1998.
8. Cen S, Pu C, Walpole J. Flow and congestion control for Internet media streaming applications. *Proceedings of Multimedia Computing and Networking*, January 1998.
9. Stewart R, Xie Q, Morneault K, Sharp C, Schwarzbauer H, Taylor T, Rytina I, Kalla M, Zhang L, Paxson V. Stream control transmission protocol. *RFC 2960*, <http://rfc.net/rfc2960.html>, October 2000.
10. Rhee I, Ozdemir V, Yi Y. TEAR: TCP emulation at receivers—flow control for multimedia streaming. *Technical Report*, Department of Computer Science, North Carolina State University, Raleigh, NC, April 2000.
11. Floyd S, Handley M, Padhye J. A comparison of equation-based and AIMD congestion control. *ACIRI Technical Report*, May 2000.
12. Akyildiz IF, Akan OB, Morabito G. A rate control scheme for adaptive real-time applications in IP networks with lossy links and long round trip times. *IEEE/ACM Transactions on Networking* 2005; **13**(3):554–568.
13. Rejaie R, Handley M, Estrin D. RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the Internet. *Proceedings of IEEE INFOCOM 1999*, vol. 3, March 1999; 1337–1345.
14. Handley M, Padhye J, Floyd S, Widmer J. TCP friendly rate control (TFRC): protocol specification. *Internet Draft*, April 2002.
15. Miyabayashi M, Wakamiya N, Murata M, Miyahara H. MPEG-TFRCP: video transfer with TCP-friendly rate control protocol. *Proceedings of IEEE International Conference on Communications (ICC'01)*, Helsinki, vol. 1, June 2001; 137–141.
16. Fang J, Akan O. Performance of multimedia rate control protocols in InterPlaNetary Internet. *IEEE Communications Letters* 2004; **8**(8):488–490.
17. Akan OB, Fang J, Akyildiz IF. Performance of TCP protocols in deep space communication networks. *IEEE Communications Letters* 2002; **6**(11):478–480.
18. Tran DT, Lawas-Grodek FJ, Dimond RP, Ivancic WD. SCPS-TP, TCP and rate-based protocol evaluation for high-delay, error-prone links. *SpaceOps 2002*, Houston, TX, October 2002.
19. Akyildiz IF, Fang J. TCP Peachtree: a multicast transport protocol for satellite IP networks. *IEEE Journal of Selected Areas in Communications (JSAC)* 2004; **22**(2):388–400.
20. Akyildiz IF, Zhang X, Fang J. TCP-Peach+: enhancement of TCP Peach for satellite IP networks. *IEEE Communications Letters* 2002; **6**(7):303–305.
21. McAuley AJ. Reliable broadband communications using a burst erasure correcting code. *ACM SIGCOMM'90*, Philadelphia, PA, September 1990; 297–306.
22. Park K, Wang W. QoS-sensitive transport of real-time MPEG video using adaptive forward error correction. *Proceedings of IEEE Multimedia Systems*, Florence, Italy, June 1999; 426–432.
23. Byers JW, Luby M, Mitzenmacher M, Rege A. A digital fountain approach to reliable distribution of bulk data. *Proceedings of ACM Sigcomm'98*, Vancouver, Canada, September 1998.
24. Byers JW, Luby M, Mitzenmacher M. Accessing multiple mirror sites in parallel: using Tornado Codes to speed up downloads. *Proceedings of INFOCOM*, New York, NY, March 1999; 275–283.
25. Pasztor A, Veitch D. On the scope of end-to-end probing methods. *IEEE Communications Letters* 2002; **6**(11): 509–511.
26. Sinha P, Venkitaraman N, Sivakumar R, Bharghavan V. WTCP: a reliable transport protocol for wireless wide-area networks. *Proceedings of ACM MOBICOM'99*, Seattle, Washington, August 1999.
27. Tsaoussidis V, Zhang C. TCP-Real: receiver-oriented congestion control. *Computer Networks Journal* 2002; **40**(4):477–497.
28. Melander B, Bjorkman M, Gunningberg P. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. *Proceedings of IEEE Globecom'00*, San Francisco, CA, November 2000; 415–421.
29. Deering S, Hinden R. Internet Protocol Version 6 (IPv6) Specification. *IETF RFC 2460*, December 1998.
30. Akyildiz IF, Morabito G, Palazzo S. TCP Peach: a new congestion control scheme for satellite IP networks. *IEEE/ACM Transactions on Networking* 2001; **9**(3):307–321.
31. Rhee I, Balaguru N, Rouskas GN. MTCP: scalable TCP-like congestion control for reliable multicast. *Computer Networks Journal* 2002; **38**(5):553–575.
32. Space Communications Protocol Specification—Transport Protocol (SCPS-TP). *Recommendation for Space Data Systems Standards, CCSDS 714.0-B-1*. Blue Book, Issue 1, Washington, DC, CCSDS, May 1999.

## AUTHORS' BIOGRAPHIES

**Jian Fang** received his PhD degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, U.S.A., in 2004.

From 1996 to 1997, he had been a research associate in the department of computer science and engineering, the Chinese University of Hong Kong, Hong Kong. After that, he became a lecturer in Shanghai Jiao Tong University, Shanghai, China. From 2001 to 2004, he was a Research Assistant in the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, U.S.A. Currently, he works in Jewelry television, Knoxville, U.S.A.

His current research interests are in Wireless Networks, InterPlaNetary Internet, Satellite Networks, Peer-to-Peer Networks and Network Security.

**Ian F. Akyildiz** received his BS, MS, and PhD degrees in Computer Engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981 and 1984, respectively.

Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology and Director of Broadband and Wireless Networking Laboratory.

He has held visiting professorships at the Universidad Tecnica Federico Santa Maria, Chile, Universite Pierre et Marie Curie (Paris VI), Ecole Nationale Superieure Telecommunications in Paris, France, Universidad Politecnico de Cataluna in Barcelona, Spain, and Universidad Illes Balears, Palma de Mallorca, Spain.

He has published over two-hundred technical papers in journals and conference proceedings. He is an Editor-in-Chief of Computer Networks (Elsevier) and for the newly launched AdHoc Networks Journal and an Editor for ACM-Kluwer Journal of Wireless Networks.

He is a past editor for IEEE/ACM Transactions on Networking (1996–2001), Kluwer Journal of Cluster Computing (1997–2001), ACM-Springer Journal for Multimedia Systems (1995–2001) as well as for IEEE Transactions on Computers (1992–1996).

He guest-edited more than ten special issues for various journals in the last decade. He was the technical program chair of the '9th IEEE Computer Communications' workshop in 1994, for ACM/IEEE MOBICOM'96 (Mobile Computing and Networking) conference, IEEE INFOCOM'98 (Computer Networking Conference), as well as IEEE ICC'2003 (International Conference on Communications). He served as the General Chair for the premier conference in Wireless Networking, ACM/IEEE MOBICOM'2002, Atlanta, September 2002. He is the co-founder and co-General Chair of the newly established ACM SenSys'03 conference on Sensor Systems which will take place in Los Angeles in November 2003.

Dr Akyildiz is an IEEE FELLOW (1995), an ACM FELLOW (1996). He received the 'Don Federico Santa Maria Medal' for his services to the Universidad of Federico Santa Maria in Chile in 1986. He served as a National Lecturer for ACM from 1989 until 1998 and received the ACM Outstanding Distinguished Lecturer Award for 1994.

Dr Akyildiz received the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society) for his paper entitled 'Multimedia Group Synchronization Protocols for Integrated Services Architectures' published in the IEEE Journal of Selected Areas in Communications (JSAC) in January 1996.

Dr Akyildiz received the 2002 IEEE Harry M. Goode Memorial award (IEEE Computer Society) with the citation 'for significant and pioneering contributions to advanced architectures and protocols for Wireless and Satellite Networking'.

Dr Akyildiz received the 2003 IEEE Best Tutorial Award (IEEE Communication Society) for his paper entitled 'A Survey on Sensor Networks', published in IEEE Communication Magazine, in August 2002.

Dr Akyildiz also received the 2003 ACM Sigmobility Outstanding Contribution Award with the citation: 'for pioneering contributions in the area of mobility and resource management for wireless communication networks'.

His current research interests are in Sensor Networks, InterPlaNetary Internet, Wireless Networks, Satellite Networks and the Next Generation Internet.