Deep Space Autonomous Network Using Reverse Channel for Congestion Control

Bighnaraj Panigrahi, Yanxiao Zhao and Kazem Sohraby
Electrical and Computer Engineering Department
South Dakota School of Mines & Technology
Rapid City, South Dakota, 57701 USA

Abstract—**The extremely long propagation delay between interplanetary nodes poses challenges to the investigation of Store-and-forward Deep Space Autonomous Network (DSAN). Long propagation delay combined with scarcity of energy at the planetary wireless nodes and relays, introduces new dimensions to the congestion control challenge. In this paper, we take advantage of long propagation delay between nodes and propose a novel approach for congestion control. The long delayed channel can be utilized as a memory resource. Instead of dropping the packets due to lack of storage at the intermediate nodes such as satellite or a robotic spacecraft, they are transmitted back to the source. This significantly reduces packet loss, and lowers the per packet energy consumption. In addition, this method provides feedback on the congestion status of the relay node. Consequently, the source is able to dynamically control traffic flow in the source-relay channel until congestion is abated. Two proposed protocols, i.e., (i) *reverse channel (without rate control)* and (ii) *reverse channel with rate control* are investigated, and compared with a conventional protocol where *no reverse channel* is utilized. Extensive simulations are conducted which validate that the performance *with reverse channel*, especially *reverse channel with rate control*, far outweigh the conventional approach, in terms of energy consumption, and retransmission per successful packet.**

Keywords—**deep space communications; reverse channel; energy consumption; retransmission.**

I. INTRODUCTION

Space exploration is the discovery of outer space and has been attracting increasing interests since the early 20^{th} century. The world-wide space agencies and private sectors, including National Aeronautics and Space Administration (NASA) and European Space Agency (ESA), have made considerable efforts to push forward the space technology. Recently, proposals to enable missions to beyond low earth orbit (LEO), such as Earth/Moon, Earth/Sun and Phobos or Mars orbit have been launched [1]. The fundamental mission of space exploration is to successfully transfer a great amount of scientific data from other planets/satellites/robotic space-crafts as wireless relays in the space.

The communications among Earth and other planets form a store-and-forward Deep Space Autonomous Network (DSAN), which is an integration of multiple independent networks. In this paper, we assume that the data that is collected on a planet is stored at a wireless gateway, and is subsequently forwarded to a relay in the sky, possibly a satellite in the planet orbit. The data is again stored in the satellite [8, 9] and when convenient it is forwarded to the Earth. For instance, sending multimedia messages from a surface sensors network in Mars to Earth may go through many nodes across multiple independent networks (we refer to planets/satellites/robotic space-crafts as nodes).

Fig. 1 depicts a DSAN, in which Mars and Earth are the source and the destination, respectively. Messages are first collected from the surface sensors network on Mars and stored in a gateway. They are delivered at an intermediate satellite/node first before received at the final destination i.e. Earth. In some cases multiple intermediate relay nodes may be used. Although, Fig. 1 describes a general DSAN with multiple-hop communications, in this paper, we assume a single hop, i.e., between the source gateway and the relay node. The unique characteristics of space communication pose challenges on DSAN. Three major challenges are summarized as follows:

978-1-4673-5208-6/13/$31.00 ©2013 IEEE

1) *Long delay*: deep space communication usually has extremely long propagation delays between the gateway and the destination, ranging from tens of minutes to hours, due to the long interplanetary distance. For instance, the end-to-end round trip time (RTT) typically ranges from 8.5 to 40 minutes between Mars and Earth with an available link [2, 3].

2) *Intermittent connectivity*: planets move around their orbits and their positions vary with time. The communication link between two nodes is not available if one node is not in line of sight of another.

3) *Limited buffer size*: intermediate nodes typically have limited buffer size. The data in such networks are mostly real time multimedia messages (photographs, videos etc.). However, due to the extremely long distance between two communicating nodes, it is not practical to use the conventional Transport layer protocol, such as TCP. In other words, it is very inefficient to use ACK/NACK, or utilize congestion control methods. Therefore, if a packet reaches a relay node where it cannot be stored, it is dropped and hence the source node cannot obtain feedback about the loss. Some of the earlier works in this area have proposed modified TCP protocols [4, 5, 6, 7] for deep space applications, but these protocols are hard to implement. Our focus in this paper is to utilize the long delay channel as a memory for congestion control.

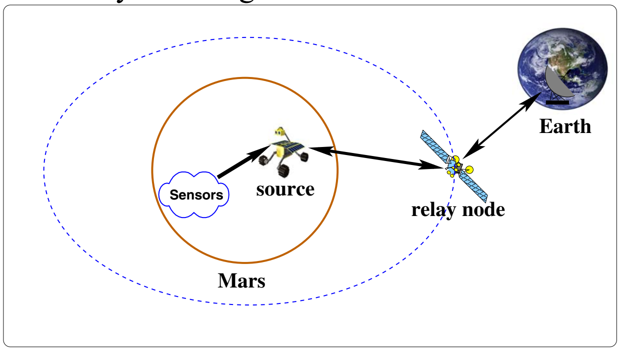

Fig. 1 Illustration of a DSAN: Messages from Mars are eventually delivered to Earth with assistance of relay nodes.

In this paper, we take advantage of the long propagation delay between nodes and propose a novel approach for congestion control. The long delayed channel can be utilized as a memory resource. Specifically, instead of dropping the packets at the intermediate node such as a satellite or robotic space-craft, due to lack of buffer space, overflow packets are transmitted back to the source. This will keep the data packets travelling back and forth in the channel itself until the destination node is available to receive packets.

One great advantage of this approach is that the probability of packet loss is significantly reduced. More importantly, this method indirectly provides feedback on the congestion status of the receiving relay node. Once the source node receives a packet back from the destination, it is able to perceive that the destination is currently unavailable. Therefore, it takes measures to control flow of packets from the source network. For instance, the source might subsequently notify the planet sensors to temporarily slow down their transmission, or the source may store the received data from the sensors and apply rate control at the transmitting channel. This action helps reduce flow of packets into the source-intermediate channel until the destination becomes available. This latter method is referred to as *reverse channel with rate control*.

We study both proposed methods, i.e., *reverse channel without rate control* and *reverse channel with rate control*, and compare with the traditional method (e.g., *no reverse channel*). Simulation results verify that the performance of *reverse channel with rate control,* outperforms the traditional approach, in terms of energy consumption, and number of retransmissions per packet.

The rest of the paper is organized as follows. Section II introduces the system model, Section III discuss in detail the proposed reverse channel method. The simulation results are shown in Section IV. Concluding remarks are in Section V.

II. SYSTEM MODEL

We consider a two-hop DSAN as the system model, in which a source, a relay node and a destination node is illustrated as in Fig. 2. A reverse channel (relay-to-source) is proposed in addition to the regular forward channel (source-to-relay). The source and the relay node both maintain a queue, which is denoted by S_queue and D_Queue, respectively, in Fig. 2.

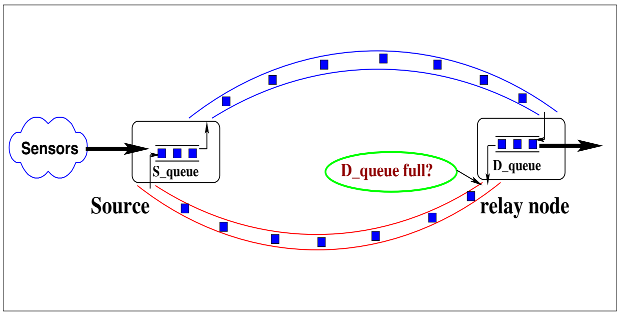

Fig. 2 Messages are transmitted from the source to the final destination through a relay node. Both the forward and the proposed reverse channel are illustrated.

It is likely that the D_Queue lacks storage, which might be caused by an unavailable connection to the destination or when the incoming traffic from the source node exceeds the outgoing traffic at the relay node. Once D_Queue is full, the packets at the relay node are lost. This may lower the throughput drastically and render the received information useless. As an alternate approach, the long delayed channel can be utilized as a memory resource. In other words, instead of dropping packets at the intermediate relay node, they can be transmitted back to the source (i.e., kept in the channel). These packets will further join the newly incoming traffic from the planetary gateway at the S_Queue and transmit to the relay node again. The process is repeated until the destination node is available to receive packets.

Note that once the source node receives a retransmitted packet from the destination, it may take action such as to control the traffic flow in the source-to-relay channel, i.e., by notifying sensors to slow down their transmission to the source, or by storing the received data in a storage at the gateway and subsequently apply a rate control at the source-to-relay channel.

Assumptions in this paper are as follows,

1. *Nodes are assumed to be static.* Although, in general planets move, but their mobility is slow compared to the packet transmission time and once nodes are in line of sight of another node, they stay connected for a period of time.
2. *No ACK mechanism.* Therefore, if a packet is dropped at the destination due to relay node congestion, the source is unaware.
3. *Dropping of packets at the destination is only due to unavailability of buffer space.* For simplicity we assume that no packets are dropped due to channel error or low SNR, etc. This is made possible by using a high transmission power at the source.

III. REVERSE CHANNEL

In this Section, we introduce the conventional approach (*no reverse channel*) and our two proposed approaches: *reverse channel* (no rate control) and *reverse channel with rate control*.

a) **No Reverse Channel**: This is the general scenario, where packet transmissions occur in the forward direction i.e., from the source to the relay node. The distance between the source and the relay node is usually large. Packets continue to be transmitted from the source to the relay node regardless of the status at the relay node. If a packet reaches the relay node and encounters lack of buffer space, it will be dropped immediately at the relay node. Obviously, this degrades the system throughput.

In order to improve the throughput, we have considered an alternate strategy in the conventional approach. In this case, source will send K redundant copies for each original packet to the relay node, with some time gap (at least relay node transmission time) between the copies so that at least the original or any one of the redundant copies of the packet may be received successfully at the relay node. However, this method consumes additional energy due to the transmission of redundant packets.

b) **Reverse Channel**: Instead of packet getting dropped at the destination, the long channel/link between the nodes is treated as temporary buffer and packets are kept in the channel until the relay node is available to receive packets. If the relay node is temporarily out of storage, then it can just send packets back to the source using the reverse channel. The reverse channel is assumed to be a downlink channel operating at a different frequency than the uplink/forward channel. This method has two benefits: 1) it may ensure that low packet loss; 2) we are able to control flow of packets into the system at the source gateway.

c) **Reverse Channel With Rate Control - α**: this is an extended model where as soon as the source receives the first packet returned from the relay node, it slows down the flow of packets entering the channel at the source by a factor $\alpha < 1$. This is accomplished by either notifying the sensors to reduce their transmissions, or storing the packets collected from the sensors at the gateway and releasing them at a lower rate. By doing this, the total number of packet transmissions in the system can be controlled and hence energy consumption can be reduced. The value of α will vary depending on the traffic on reverse channel. It also depends on the storing energy consumption.

Although, in general, energy consumption in packet transmission is greater than energy consumption in packet storing, however, due to long buffering periods and cost involving maintaining power sources at nodes (satellites etc.), sometimes storing energy becomes very significant in deep space scenarios. Therefore, in our model we have also tried to observe the effect of storing energy consumption on the system. Energy consumption for storing depends on many factors like system configurations, hardware specifications, operating system, memory management algorithms etc. For example, energy consumption in a Fujitsu MHV2120BH 120GB, 5400rpm SATA/150 with 3.6 GHz Intel processor and Microsoft Windows Server 2003 OS is 2.3 Watts [10].

Due to the difficulty in implementation, high cost, and limited size of the nodes, energy is a vital resource in such networks. Therefore, we employ the following two performance criteria which aim to reduce the energy consumption:

1. Energy consumption per successful packet, E_c: this is the average consumption per successful packet, considering total energy consumption, due to packet transmissions, retransmissions, as well as storage in the buffers.
2. Average number of retransmissions per successful packet, N_a: on average how many times a packet is being retransmitted before it is successfully relayed to the final destination, considering packet flow in both directions.

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed methods, i.e., *reverse channel* (with no rate control), *reverse channel with rate control*, and compare with the traditional approach (*no reverse channel*). Energy consumption and average number of retransmissions per successful packet are simulated.

We have developed a simulation model using gcc 4.6 on a Linux platform. A two-hop model has been considered, where the source gateway node gathers packets from the sensor networks and transmits them to a relay node. The relay node eventually delivers the packets to the final destination. The relay node may store the packet (for further relaying), drop the packet (if out of storage space), or send the packet back to the transmitter, depending on the availability of storage space and the connectivity to the final destination.

The distance between the source and the relay node assumed to have a propagation delay of 400 seconds. A direct channel of 100 Kbps is assumed. Packets with a constant size of 10 KB arrive at the source queue at the gateway, following an exponential inter-arrival time distribution with an average packet inter arrival time of 1 sec. In order to simulate the condition of congestion at the relay, we set the queue size at the relay node at 50 packets. Packets from the relay node are transmitted with a constant rate R_{dst}. If this rate is **smaller** than the transmission rate at source, then packets will accumulate at the D_queue and once the queue is full, packets may be lost or sent back to the source depending on the particular control method in use. Due to very large distances between the nodes (orders of thousands of Kilometers), high transmit power is used for deep space communication, using high gain antenna amplifiers [11]. The transmit and receive power (P_t and P_r) per packet are assumed to be 20 and 15 Watts respectively with a frequency of 30 GHz in the range of Ka-band. In order to have a fair comparison, performance of each method is observed in the steady state and for a duration of 2000 seconds.

In Fig. 3, average number of retransmissions per successful packet for different methods is shown. This is the ratio of the total number of retransmissions in the system and total number of successful packets received at the relay node during the simulation time. It is observed that number of retransmissions in the system decreases with a higher data rate (i.e., lower level of congestion) at the relay node. Average number of retransmissions is highest in the case of no reverse channel as it transmits K=3 redundant packets for every original packet. With rate control at the source, the number of packet transmission is reduced.

Fig. 4 shows average energy consumption per successful packet. This value is the combination of transmission energy as well as

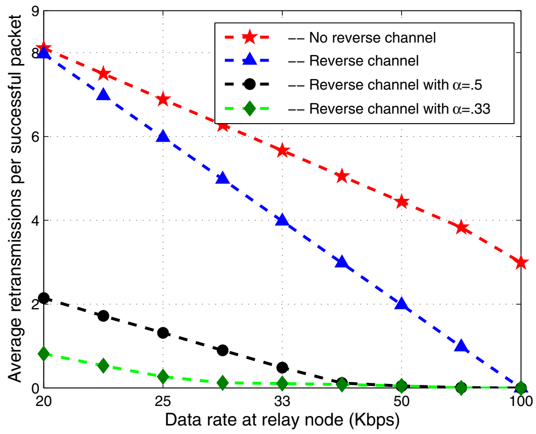

Fig.3 Average number of retransmissions per successful packet as data rate at relay node increases

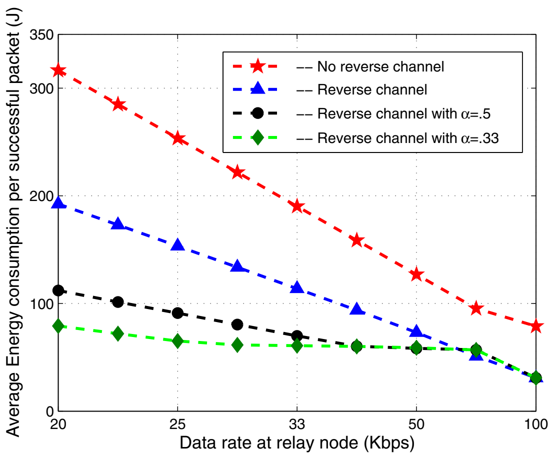

Fig.4 Average energy consumption per successful packet, E_s = 2 mJ/s

energy required to store packets at the buffer. It is noticed that *no reverse channel* leads to the most energy consumption and *reverse channel with rate control* is the most efficient one. In addition, energy consumption decreases with a higher data rate at the relay node. With a higher rate control at the source gateway, energy consumption significantly reduces, due to fewer number of transmissions in the system, and lower energy consumption due to storage

In Fig. 5, the effect of storing energy used for storage, E_s on average energy consumption per successful packet is shown. As, the exact value of storing energy is system specific (which is spacecraft or rover), we tried to test results with extreme case values. It is observed that at low or moderate storing energy cases, reverse channel with rate control has the lowest energy consumption. With very high storing energy, all reverse channel methods have relatively high energy consumption. This is because large number of packets retransmitted back from the relay node has to stay in S_queue before they again get transmitted. With rate control, the energy consumption is highest due to temporarily blocking packets in the S_queue.

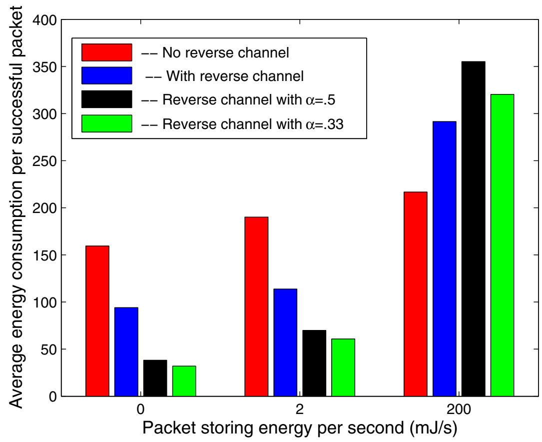

Fig.5 Average energy consumption per successful packet with different E_s values, R_{dst}=33Kbps

Fig. 6 shows average energy consumption per unit time period. This is calculated as the ratio of total energy consumption during the simulation time and total number of successful packet receptions at the relay node. It can be observed that although slowly, energy consumption per unit time increases with heavier congestion at the relay node. The slow increment is resulted by the total number of transmissions in the system, which also increases with a slower rate with congestion at the relay node.

In the case of *no reverse channel*, although 3 additional copies for each original packet is sent by the source, at very high congestion scenarios, it may happen that all the 4 copies of a packet may get dropped at the relay. Fig. 7 shows percentage of number of packets lost where all copies along with original could not be relayed. As it can be seen this percentage in-

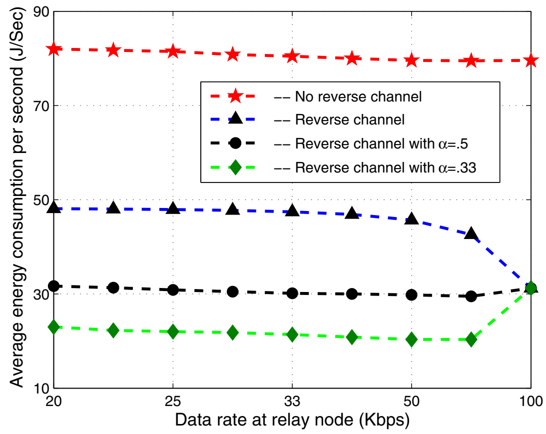

Fig.6 Average energy consumption per unit time period

-creases with high congestion at the relay node.

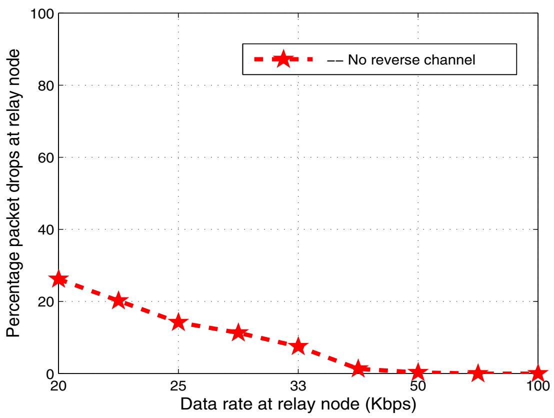

Fig.7 Packet dropping percentage at relay node

V. CONCLUSIONS

In this paper, we have proposed to use reverse channel for congestion control in deep space communications. Specifically, two methods, i.e., *with reverse channel* and *reverse channel with rate control* are proposed. Extensive simulations have verified that the performance with *reverse channel*, especially *reverse channel with rate control*, perform far better than the conventional one (no reverse channel) in terms of energy consumption, retransmissions per successful packet.

ACKNOWLEDGEMENT

This research is partly supported by NASA's grant NNX12AI13A.

REFERENCES

[1] Address at KSC 15 April 2010. "President Outlines Exploration Goals, Promise"

[2] R.C. Durst, P.D. Feighery, K.L. Scott, Why not use the Standard Internet Suite for the InterPlaNetary Internet? http://www.ipnsig.org/techinfo.htm.

[3] I. F. Akyildiz, B. Akan, C. Chen, J. Fang, W. Su, "InterPlaNetary Internet: state-of-the-art and research challenges", in *Computer Networks*, pp. 75-112, 2003.

[4] I.F. Akyildiz, O.B. Akan and J. Fang, "TP- Planet: A reliable transport protocol for InterPlaNetary Internet", *Journal on Selected Areas in communications*, Vol. 22, Issue 2, pp. 348-361, February 2004.

[5] Jian Fang and Ian F. Akyildiz, "RCP-Planet: A Multimedia Rate Control Protocol for InterPlaNetary Internet", International Journal of Satellite Communications and Networking, John Wiley and Sons, Vol. 25, Issue 2, pp. 167-194, February 2007.

[6] Space Communications Protocol Specification (SCPS). Transport Protocol SCPS-TP). *Recommendation for Space Data System Standards*, CCSDS 714.0-B-Blue Book. Issue 2. Washington, D.C.: CCSDS, October2006.

[7] I. Psaras , G. Papastergiou , V. Tsaoussidis, N. Peccia, "DS-TP: Deep-Space Transport Protocol ", *IEEE Aerospace Conference*, pp. 1-13, March, 2008.

[8] V. Cerf et al., "InterPlanetary Internet (IPN): architectural definition, Internet draft, draft-irtf-ipnrg-arc-00.txt, May 2001

[9] S Burleigh, V. Cerf , R. Dust, K. Fall, A. Hooke, K. Scott, H. Weiss, "The InterPlanetary Internet: a communications infrastructure for Mars exploration", in: 53rd International Astronautical Congress, The World Space Congress, Houston, Texas, Oct., 2002.

[10]http://www.tomshardware.com/reviews/conventional-hard-drive-obsoletism,1324-5.html

[11]http://mars.jpl.nasa.gov/mro/mission/spacecraft/parts/antennas/.