

## RESEARCH

## Open Access



# Transfer time analysis of file transfer framework with AL-FEC in SOTM networks

Kyu-Hwan Lee, Jong-Mu Kim and Jae-Hyun Kim\*

## Abstract

In this paper, we propose a fully reliable file-transfer framework with application layer forward error correction (AL-FEC) in satellite communications on the move (SOTM) systems. In particular, the proposed framework uses a two-way acknowledgement (ACK) exchange mechanism. The proposed framework is implemented into a performance-enhancing proxy to minimize the system change. Furthermore, we analyze the file-transfer time of the proposed framework by Markov chain. The analysis results show that the proposed framework outperforms TCP in terms of the file-transfer time and the goodput. Furthermore, in the transfer of a large-sized file, the overhead of the proposed AL-FEC mechanism is small in terms of the resource efficiency.

**Keywords:** Satellite communications; SOTM network, AL-FEC; File transfer time; Queuing analysis

## 1 Introduction

Satellite communications on the move (SOTM) systems have become an essential part of commercial and military communications because they offer the high-speed wireless link to moving vehicles such as maritime vessels, trains, and land vehicles through a satellite [1–3]. In the SOTM system, the antenna of the SOTM terminal, which is equipped with an active control system and an inertial navigation system is pointed to the satellite. Thus, the wireless link between the SOTM terminal and the satellite can be continuously maintained. However, the link between the SOTM terminal and the satellite can experience a temporary outage owing to channel blockage due to pointing errors of the antenna on account of the rugged terrain and obstructions such as high buildings and trees. It can cause packet loss in the satellite link [2, 3].

There are two main error recovery techniques: retransmission and forward error correction (FEC). In satellite communications, both retransmission and FEC are used. However, in satellite communications using retransmission of TCP, even though optimized TCP can be applied to satellite link by a performance enhancing proxy (PEP) [4, 5], the throughput can be sharply reduced within a lossy environment because of the coupling of the congestion control and losses in TCP. Because TCP was designed

for wired environment, a loss in TCP is always ascribed to congestion. As a result, the TCP congestion window is decreased even though the loss is actually not due to congestion but to channel blockage. This problem is worsened on satellite communications since it takes time to re-open the TCP congestion window and retransmit data due to the long propagation delay. Furthermore, an FEC technique such as channel coding that is applied in the physical layer of the satellite communications only corrects data corruption due to the noise and the interference. It cannot recover packets lost owing to the channel blockage in the SOTM network because of its fixed time diversity and fixed amount of protection [1].

Recently, an application-layer FEC (AL-FEC) is applied to satellite communications for SOTM terminals [6, 7]. The AL-FEC system can not only correct for packets lost owing to the channel blockage with its flexible time diversity and flexible amount of protection but also eliminate retransmission delay by FEC. Furthermore, the AL-FEC system can potentially achieve better performance as compared with an acknowledgement (ACK)-based system as TCP because the AL-FEC system is a rate-based system with a constant transmission rate through a user datagram protocol (UDP) [8]. In this system, losses do not cause any impact on the transmission rate, resulting in enhancing the throughput. However, an additional feedback mechanism that notify the file-decoding completion is needed

\*Correspondence: [jkim@ajou.ac.kr](mailto:jkim@ajou.ac.kr)

Department of Electrical & Computer Engineering, Ajou University, 443-749 Suwon, Korea

to provide fully reliable file transfer because AL-FEC system is based on the unreliable transfer method such as UDP [9]. Some studies have been researched to provide the full reliability of file transfer based on AL-FEC by the additional feedback [9–11]. To deploy these systems, file-transfer systems of all servers and end-hosts should be changed.

In this paper, we propose a fully reliable file-transfer framework (FRFTF) with AL-FEC in SOTM networks to minimize the additional system change. In the proposed FRFTF, we use an ACK exchange scheme to provide the reliability of the end-to-end data transfer. To minimize the system change, FRFTF is implemented inside a performance-enhancing proxy (PEP) with the splitting connection. We then analyze the transfer time of a file of FRFTF in SOTM networks. The main contributions of the paper are as follows:

1. A fully reliable file transfer framework with AL-FEC in SOTM networks
2. A performance analysis model evaluating the performance of the reliable AL-FEC mechanism with Markov chain model

## 2 Related works

### 2.1 TCP

Generally, TCP is used in the file-transfer system because TCP can provide the reliable, in-order delivery, and the congestion control thanks to its sliding window and ACK-based systems [12]. However, TCP performance can be degraded in the wireless environment because of the coupling of the congestion control and losses in TCP. In the satellite communication with the long propagation delay, this problem is worsened. Thus, in the satellite communication, the optimized TCP such as TCP Hybla is used [13]. In the optimized TCP, the TCP congestion window is aggressively increased to enhance the TCP throughput. However, the problem of the ACK-based system still remains. Therefore, we propose the file transfer framework with AL-FEC that is rate-based system with a constant transmission rate aided by the cross-layer design.

### 2.2 Benefit of AL-FEC

In wireless communications, channel coding is important because it ensures the reliability of data transmission protecting it from the data corruption by noise and interference. However, in mobile satellite networks, channel blockage due to intermittent shadowing can cause packet loss even though channel coding is applied, as shown in Fig. 1. To solve this problem, many studies have investigated AL-FEC in many communication systems [8, 14–16]. AL-FEC covers the packet loss not recovered by channel coding because it is applied above layer 2 and uses the fountain code known as the rateless erasure code.

Figure 1 shows the example for the benefit of AL-FEC in the channel blockage. It is shown that AL-FEC compensates the packet loss caused by the channel blockage by means of longer source block and the repair block generated by AL-FEC [17]. Recently, Raptor code has been commercially used in AL-FEC. It is fountain codes with linear time encoding and decoding because the encoding with degree distribution and the belief propagation decoding are used [15]. Furthermore, it provides dynamic packet loss protection, exceptionally high computational efficiency, and low transmission and reception overheads [15, 18]. Therefore, its advantages allow a software implementation and also provide end-to-end error correction without requiring any change in legacy standards, resulting in ease of deployment in the network [15]. To the best of our knowledge, the file-transfer time of the fully reliable file transfer with AL-FEC has not been explored. Therefore, we propose not only fully reliable file transfer framework with AL-FEC for SOTM networks but also analyze the transfer time of the proposed FRFTF.

## 3 Proposed file transfer framework

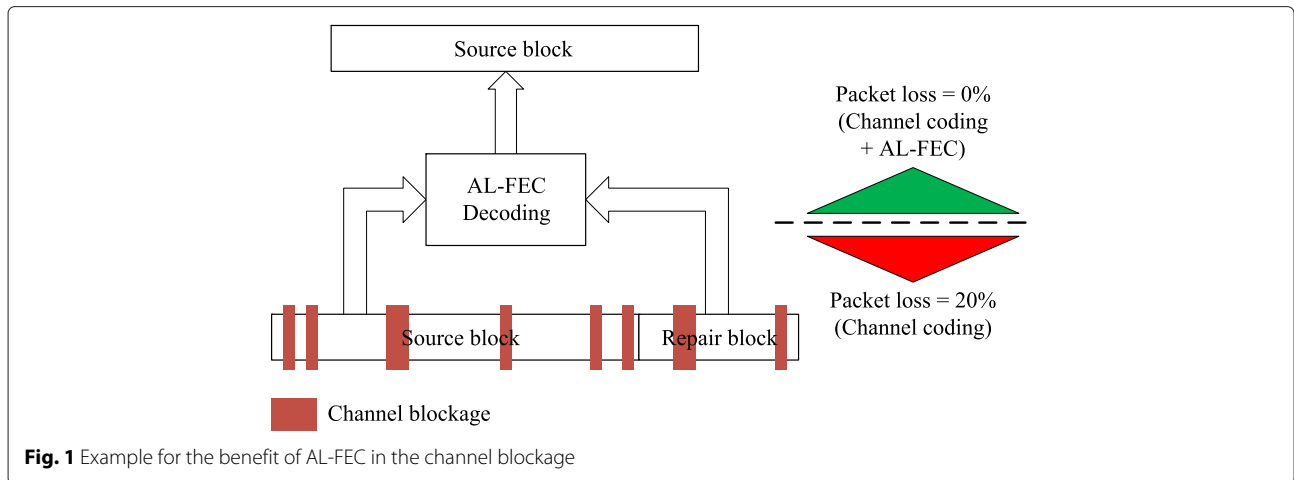
### 3.1 System model

The system model consists of a ground station, a SOTM node, a satellite, and file servers as shown in Fig. 2. The SOTM node is connected to the ground station by satellite links. Because SOTM nodes have mobility, the satellite link can be intermittently disconnected owing to the rugged terrain and obstructions such as high buildings and trees. The ground station is connected to file servers by high-speed wired links, and a performance-enhancing proxy (PEP) with the splitting connection is implemented in it [4, 5]. The ground station that received files from file servers transmits data files to the SOTM nodes. In the AL-FEC system of the proposed FRFTF, a file is segmented into  $k$  native packets. The native packet is the original data segmented from a file.  $k$  can be calculated as

$$k = \left\lceil \frac{L_F}{L_S} \right\rceil, \quad (1)$$

where  $L_F$  and  $L_S$  are the sizes of a file and a native packet, respectively. Repair packets are generated from  $k$  native packets by RaptorQ code. RaptorQ code is the advanced version of Raptor code [19]. In RaptorQ code, the decoding probability is 99.9999 % when  $k + 2$  packets are received regardless of the block length  $k$  [19, 20]. It is transmitted to recover packets lost due to the channel blockage. The size of the repair packet is  $L_S$ . We make the following assumptions regarding our FRFTF:

1. A two-state Markov chain model is used as the channel model [2]. The channel model has channel open (o) and blockage (b) states.



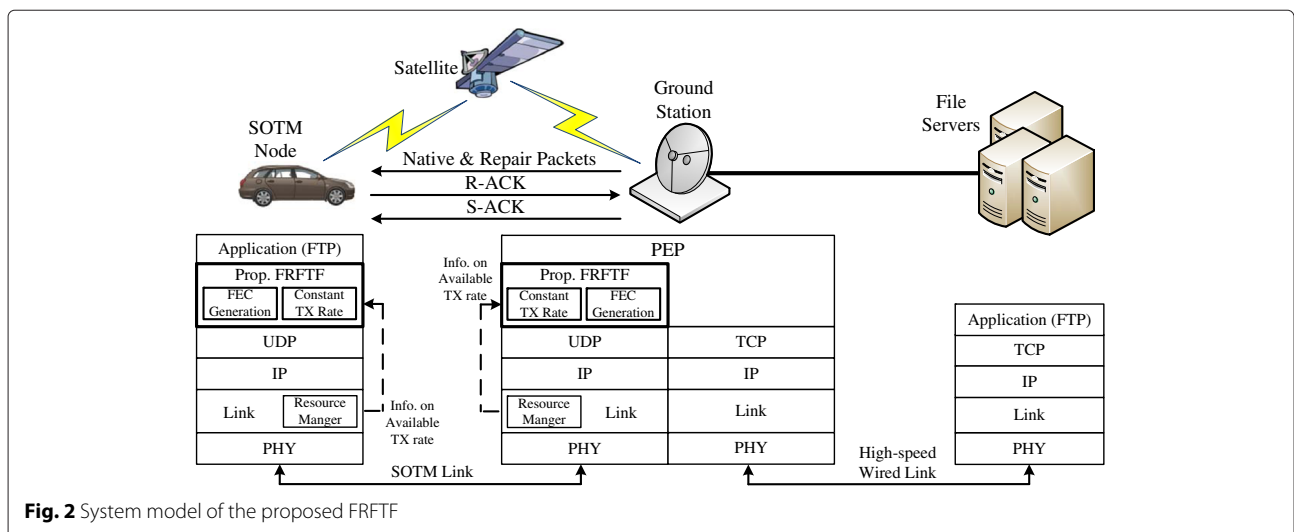
2. In the link layer of satellite communications, the information on satellite resource is shared by a cross-layer design [21, 22]. Thus, this information can be used in FRFTF.
3. If the SOTM node receives  $k + 2$  packets generated from a file, it can decode the file without error [19, 20].
4. The processing time for AL-FEC is insignificant [20].

**3.2 Architecture of proposed file-transfer framework**

The proposed FRFTF is shown in Fig. 2. To apply the AL-FEC to the file-transfer system, the AL-FEC system should be implemented in all file servers and end-host nodes. However, it is not easy to change the file system in all servers. Thus, in FRFTF, the PEP solution with the splitting connection is exploited [4, 5]. FRFTF is only implemented in the PEP of the ground station and SOTM nodes as shown in Fig. 2. In FRFTF, the TCP connection is used between the ground station and the file server.

The UDP connection is used between the ground station and the SOTM node. As a result, FRFTF can be compliant with conventional a file-transfer system. In FRFTF, the file download should be completed between the ground station and the server to generate repair packets. However, the additional delay by the splitting connection is not incurred thanks to the high capacity of the wired link between the ground station and the file server. Its capacity is much greater than that of the satellite link. Therefore, the file download from the server to the ground station can be finished while transmitting native packets from the ground station to the SOTM node.

All packets generated from FRFTF are transferred to the UDP layer by constant transmission rate,  $R_{TX}$ . However, if the sum of the speed for parallel connections exceeds the overall available bandwidth, FRFTF can incur the buffer overflow in the link layer, resulting in the poor performance. Thus, the ruler to decide  $R_{TX}$  is needed. FRFTF uses the information on available resource from a resource



manager in link layer by cross-layering to select  $R_{TX}$  [21, 22]. For the fair bandwidth sharing among parallel connections,  $R_{TX}$  can be selected as

$$R_{TX} = \min(R_D, R_A/\mu), \quad (2)$$

where  $R_D$  and  $R_A$  are the demanded rate for a file-transfer service and the available rate in FRFTE, respectively.  $\mu$  is the number of connections for FRFTE. We assume that  $R_A$  can be calculated by the information on available resource in link layer and header overheads of link layer and UDP/IP [21, 22]. If the quality of the service should be guaranteed in the system, a call admission control can be applied to limit the number of connections in FRFTE [23, 24]. When the SOTM node is file server, the information on available resource in the link layer is used to select  $R_{TX}$  by the cross-layer design [25, 26].

### 3.3 Procedure

The proposed FRFTF lies between the application layer and the transport layer as shown in Fig. 2. The detailed procedures of the file transfer in FRFTF in the sender and receiver are as follows:

1. Sender: Upon receiving the data of a file from the file server, FRFTF of the sender segments it into native packets by the size of  $L_S$ . FRFTF then inserts native packets into both transmission and encoding queues. Packets in the transmission queue are transferred to UDP layer by constant TX rate of  $R_{TX}$ . Next, when FRFTF receives a file completely, it generates repair packets from the  $k$  native packets in the encoding queue with RaptorQ code. After the repair packet generation, all the packets are inserted into the transmission queue. When receiving the receiver-ACK (R-ACK) message that indicates the completion of the file reception from the receiver, the FEC framework removes all packets from the transmission queue and terminates the file transfer to the receiver. The sender then sends the sender-ACK (S-ACK) that indicates the reception of R-ACK to the receiver. If the sender receives duplicated R-ACK, the sender retransmits the S-ACK.
2. Receiver: Upon receiving packets from the sender, FRFTF of the receiver inserts the packets into the reception queue. FRFTF checks whether the packets in the reception queue can be decoded to a file by RaptorQ code. If the decoding is completed, the receiver sends an R-ACK message to the sender (ACK-based scheme) and the file is forwarded to the application layer. If the receiver does not receive S-ACK message within round trip time (RTT), it retransmits the R-ACK message to the sender.

In FRFTE, the two-way ACK exchange is used to provide the reliability of data transfer because the S-ACK and R-ACK messages can be lost due to the channel blockage in the satellite communication environment. At the receiver, there is no extra delay caused by the ACK exchange because a file is forwarded to the application layer immediately after the completed decoding. However, the ACK exchange can incur the additional usage of satellite resource at the sender because the sender continuously transmits the packets to the receiver until receiving the R-ACK. The detailed analysis of this overhead is discussed in Section 4.

## 4 Performance evaluation

In this section, we theoretically derive the average file-transfer time and resource efficiency for transmitting a file by the simple Markov chain model. We also evaluate the performance of the proposed FRFTF in SOTM networks through the theoretical analysis and the simulation.

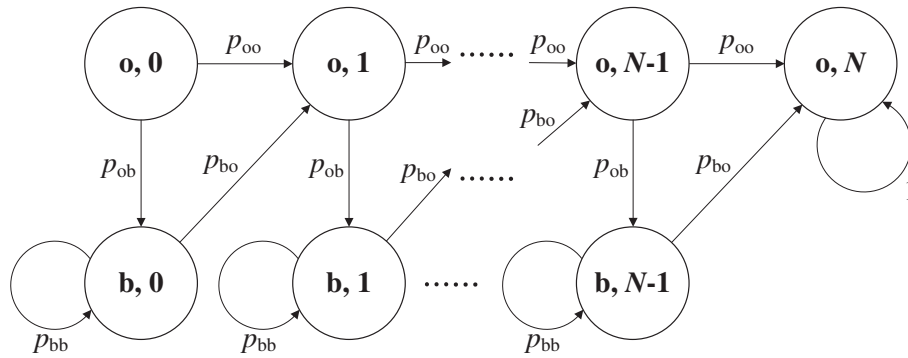
### 4.1 File transfer time

To calculate the average file-transfer time,  $T_{FT}$ , we model the bidimensional process  $\{s(t), n(t)\}$  with the discrete-time Markov chain depicted in Fig. 3. In this Markov chain,  $s(t) \in \{b, o\}$  is the channel state,  $n(t) \in \{0, 1, \dots, N\}$  is the number of packets transmitted successfully, and  $t$  is the time measured in slots.  $N$  is the required number of packets received at the receiver for decoding a file. Thus,  $N$  is  $k + 2$  as mentioned in Section 2. B. A slot is equal to the time  $T_S$  used to transmit one packet of size  $L_S$ .  $T_S$  is  $\frac{L_S}{R_{TX}}$ . Since the channel model is a two state Markov chain, the state transition probabilities are

$$P\{s(t), n(t) | s(t-1), n(t-1)\} = \begin{cases} P\{o, i|b, i\} = p_{ob}, & i \in (0, N-1) \\ P\{o, i|o, i+1\} = p_{oo}, & i \in (0, N-1) \\ P\{b, i|o, i+1\} = p_{bo}, & i \in (0, N-1) \\ P\{b, i|b, i\} = p_{bb}, & i \in (0, N-1) \\ P\{o, N|o, N\} = 1, \end{cases} \quad (3)$$

where  $p_{xy}$  is the state transition probability from the channel state  $x \in \{b, o\}$  to the channel state  $y \in \{b, o\}$  in the channel model. In our model, the file transfer is completed in state  $\{o, N\}$ . Therefore, similar to [27],  $T_{FT}$  can be calculated from the expected number of times that the process is in state  $\{o, N\}$  if it is started in state  $\{s(t), n(t)\}$ . The expected number of times  $\psi$  can be calculated as

$$\psi(s(t), n(t)) = \begin{cases} \psi(o, i) = \left(1 + \frac{p_{ob}}{1-p_{bb}}\right)(N-i), \\ \psi(b, i) = \frac{1}{1-p_{bb}} \{1 + (N-i-1)(p_{ob} + p_{bo})\}, \end{cases} \quad (4)$$



**Fig. 3** Markov chain model for average file-transfer time of the proposed FRFTF

where  $i \in (0, N - 1)$ . The possible initial states are  $\{o, 0\}$  and  $\{b, 0\}$  in Fig. 3. Consequently,  $T_{FT}$  can be derived as

$$T_{TF} = (\pi_o \psi(o, 0) + \pi_b \psi(b, 0)) \times T_S + T_P, \quad (5)$$

where  $\pi_o$  and  $\pi_b$  are the steady state probabilities in the channel model. They can be calculated from the transition probabilities of the channel model [27].  $T_P$  is the propagation delay of the satellite link.

The average goodput  $G$  can be defined as

$$G = \frac{L_F}{T_{FT}}. \quad (6)$$

#### 4.2 Resource efficiency

To derive the resource efficiency of FRFTE, we consider the additional consumption of the resource caused by the ACK-based scheme. Because the sender can terminate the file transfer upon receiving the ACK message, the resource is basically wasted in  $2T_P$ . Furthermore, if the R-ACK message is lost, the resource is additionally wasted during retransmission of ACK messages. The resource usage time due to retransmission of ACK messages is  $2T_P(1 - \pi_b)(\pi_b + 2\pi_b^2 + 3\pi_b^3 + \dots)$ . This is approximately  $2T_P \frac{\pi_b}{(1 - \pi_b)}$ . On the other hand, if the S-ACK message is lost, the additional resource is only needed to retransmit ACK messages. However, it is insignificant because the size of ACK messages is very small. Thus, we do not consider the loss of the S-ACK message in the resource efficiency. Therefore, the average resource efficiency  $\eta$  is given as

$$\eta = \frac{L_F}{T_{RU} R_{TX}}, \quad (7)$$

where  $T_{RU}$  is the total resource usage time to complete a file transfer with FRFTF. For the average file transfer time, the resource usage time is  $T_{FT} - T_P$ . Thus, taking into consideration the resource usage time for the average file transfer time and the wasted resource by ACK exchange,  $T_{RU}$  can be calculated as

$$T_{RU} = T_{FT} + T_P \left( 1 + \frac{2\pi_b}{1 - \pi_b} \right) + \varepsilon, \quad (8)$$

where  $\varepsilon$  is the processing time of the ACK message. However, it may be negligible.

#### 4.3 Performance analysis

In the performance analysis, we compare the performance of the proposed FRFTF with those of TCP Reno, PEPsal and conventional AL-FEC mechanisms for the environments listed in 1 [2, 4, 6, 8, 12, 14]. PEPsal is conventional PEP solution in the satellite communication [4]. PEPsal uses the PEP with the TCP splitting connection. In the satellite link, PEPsal can use TCP variants to enhance TCP throughput [4]. In the simulation, we used TCP Hybla as TCP variant in PEPsal. TCP Hybla is one of well-known TCP variants for the satellite link [13]. In conventional

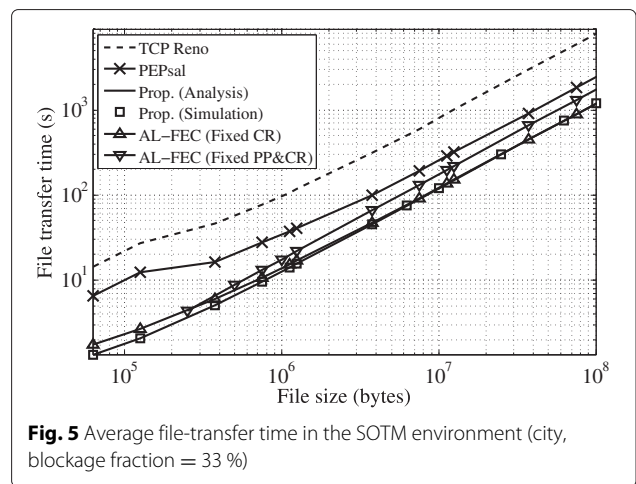
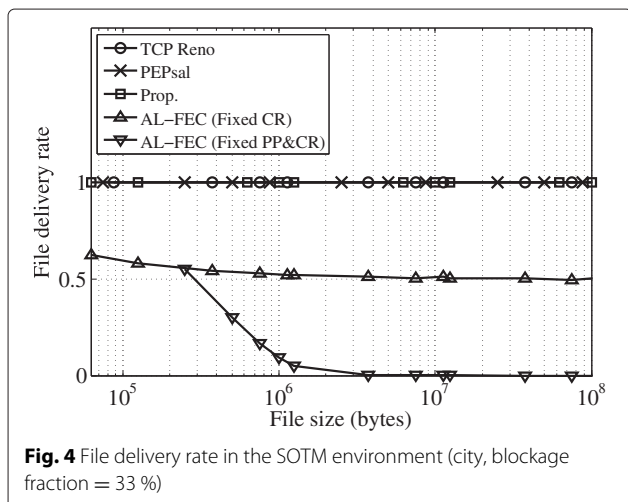
**Table 1** Environment in the performance analysis

Parameter	Value
Blockage fraction	33 %
Mean blockage	15 m
Mean open	30 m
Velocity of SOTM node	60 km/h
$T_P$	0.25 s
$R_{TX}$ (Prop. framework)	1 Mbps
$L_F$	62.5 kbytes – 112.5 Mbytes
$L_S$	500 – 5000 bytes
Maximum window size (TCP)	65,535 bytes
Segment size (TCP)	1500 bytes
$K$ of RTO (TCP)	4
$\alpha$ of RTO (TCP)	0.125
$\beta$ of RTO (TCP)	0.25
Maximum RTO (TCP)	64 s
SACK (TCP)	Enabled
$RTT_0$ (TCP Hybla in PEPsal)	20 ms

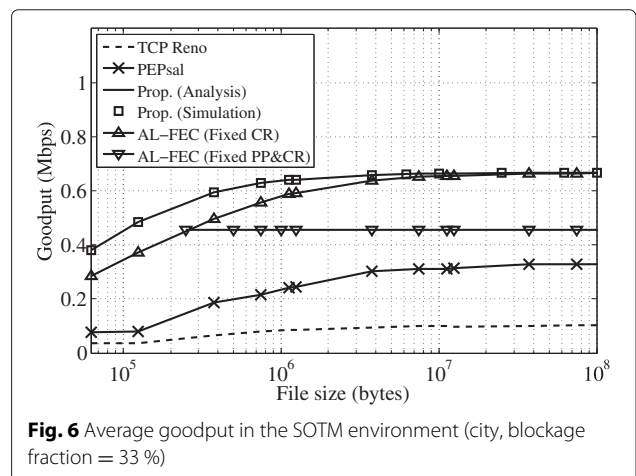
AL-FEC mechanisms, fixed protection period (PP) and code rate (CR) are used [8, 14]. We have implemented an event-driven simulator of FRFTF and conventional AL-FEC mechanisms in MATLAB. Because of the complex operation of TCP, we use the Riverbed modeler formerly known as OPNET to implement streamlined simulators of TCP Reno and PEPsal [28]. In this streamlined simulator, main functions of TCP such as sliding window, slow start, congestion avoidance, retransmission, retransmission timeout (RTO) calculation, and selective ACK (SACK) are included [12]. TCP Hybla algorithms used in PEPsal are also implemented in the simulator [13]. In the simulation, we used the channel blockage statistics as the channel model of SOTM. Channel blockage statistics are based on the measurements in the field test [2]. We consider environment in the city. In FRFTF,  $L_S$  is varied according to  $L_F$  because of the size limitation for  $k$ . For the conventional AL-FEC mechanisms, we assume that the loss rate on channel is known in AL-FEC mechanisms by the cross-layer design [21, 22]. Thus, fixed PP and CR are 2 s and 2/3, respectively, because the mean open distance, velocity, and the blockage fraction are 30 m, 60 km/h, and 33 %, respectively [2].

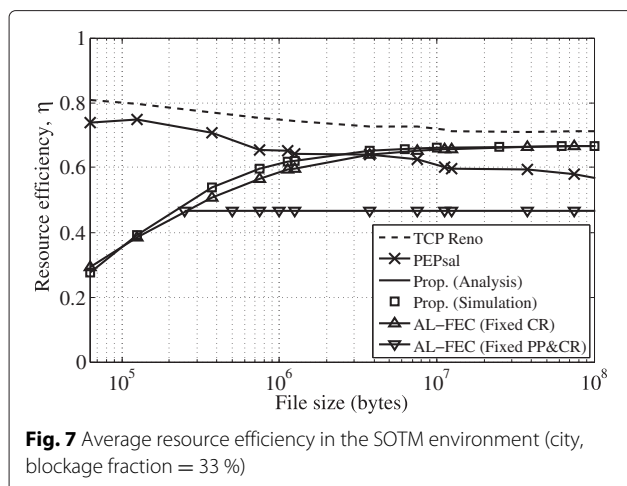
Initially, we evaluate the reliability of FRFTF. Figure 4 shows a file-delivery rate in the application layer. It is shown that FRFTF offers the fully reliable file transfer in the application layer thanks to the transmission of repair packets. However, in conventional AL-FEC mechanisms, the fully reliable file transfer is not supported because there is no ACK exchange scheme.

Figures 5, 6, and 7 show the average file-transfer time, the average goodput, and the average resource efficiency in the SOTM environment, respectively. In these results, we apply the ACK exchange scheme to conventional AL-FEC mechanisms. Additionally, a negative ACK (NACK) exchange is applied in conventional AL-FEC mechanisms



because they use the fixed CR. In simulation, if the SOTM node with conventional AL-FEC mechanisms does not completely receive a file, it sends the NACK message to the ground station. In Fig. 5, the file-transfer time of FRFTF is less than that of TCP and PEPsal because FRFTF is a rate-based system and eliminates the retransmission process by the transmission of repair packets by AL-FEC. In the satellite communication using TCP, the coupling of the congestion control and losses in TCP reduces the TCP throughput in the long propagation delay. In Fig. 6, it is shown that FRFTF outperforms TCP and PEPsal in terms of the goodput because of the lower file transfer time of FRFTF. In the transfer of a small-sized file, the goodput of FRFTF is 5–10 times more than that of TCP and PEPsal. In the transfer of a large-sized file, the goodput is improved by about 560 and 100 %, respectively. In Fig. 7, it is shown that the resource efficiency is lower in FRFTF than in TCP. For the transfer of a small-sized file using FRFTF, the resource efficiency is greatly reduced because of the wasted resource during the ACK transmission, and the continuous data transmission regardless of whether





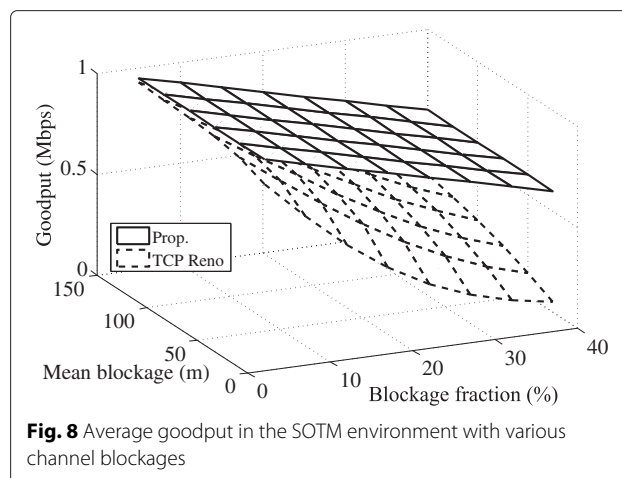
the channel is open or blocked. On the other hand, the resource efficiency of FRFTF is slightly lower than that of TCP in the transfer of a large-sized file. Since the wasted resource during the ACK transmission in FRFTF is fixed, the overhead due to the wasted resource during the ACK transmission is reduced with increasing file size. In the case of PEPsal, the resource efficiency can be lower than that of FRFTF because the optimized TCP is used. The optimized TCP increases the TCP window size aggressively to enhance the TCP throughput in the satellite communication with the long propagation delay [13]. Thus, the number of the packets transmitted during the channel blockage increases, resulting in reduction of the resource efficiency. In addition, retransmitted packets can be repeatedly transmitted during the channel blockage in the frequent channel blockage.

In Figs. 5, 6, and 7, it is observed that FRFTF outperforms conventional AL-FEC mechanisms in terms of the average file-transfer time, the average goodput, and the average resource efficiency because fixed PP and CR reduce the efficiency of the AL-FEC mechanism in the SOTM system.

Figure 8 shows the average goodput in an environment with various channel blockages for varying the blockage fraction and the mean blockage. It is clear that FRFTF also outperforms TCP in terms of the goodput in an environment characterized by various blockage fractions and mean blockages.

## 5 Conclusions

In this paper, we proposed a reliable file-transfer framework with AL-FEC in SOTM networks to reduce the file transfer time. We also analyze the file transfer time of the proposed FRFTF with the Markov chain model. Simulation results indicated that FRFTF can reduce the file-transfer time owing to the rate-based approach and elimination of retransmission delay. As a result, FRFTF



outperformed conventional TCP in terms of the goodput. In the file transfer, the goodput of FRFTF is 6.6–10 times more than that of TCP. Furthermore, the overhead of FRFTF is small for large-sized file transfers in terms of the resource efficiency. Therefore, FRFTF can be applied to various services such as the transfer of urgent messages, mail service, web browsing, and file-transfer protocol in SOTM environments of military and commercial communications.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

This research was supported by National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(NRF-2014R1A2A2A01002321).

Received: 10 March 2015 Accepted: 7 October 2015

Published online: 28 October 2015

## References

1. T Taleb, Y Hadjadj-Aoul, T Ahmed, Challenges, opportunities, and solutions for converged satellite and terrestrial networks. *IEEE Wireless Commun.* **18**(1), 46–52 (2011)
2. WM Smith, in *Channel characterization and modeling for satellite communications on the move*. Proc. IEEE MILCOM, (2005), pp. 821–827
3. V Weerackody, EG Cuevas, Technical challenges and performance of satellite communications on-the-move systems. *JOHNS HOPKINS APL TECHNICAL DIGEST.* **30**(2), 113–121 (2011)
4. C Caini, R Firrincieli, D Lacamera, PEPsal: A performance enhancing proxy for TCP satellite connections. *IEEE Aerosp. Electron. Syst. Mag.* **22**(8), 9–16 (2007)
5. P Davern, N Nashid, CJ Sreenan, A Zahran, HTTPPEP: A HTTP performance enhancing proxy for satellite systems. *Int J Next Gener Comput(JNGC).* **2**, 242–256 (2011)
6. Digital video broadcasting (DVB), upper layer FEC for DVB systems. ETSI TR 102 993, 1–89 (2011). [http://www.etsi.org/deliver/etsi\\_tr/102900\\_102999/102993/01.01.01\\_60/tr\\_102993v010101p.pdf](http://www.etsi.org/deliver/etsi_tr/102900_102999/102993/01.01.01_60/tr_102993v010101p.pdf). Accessed 27 Oct. 2015
7. Digital video broadcasting (DVB), IP datacast: Content delivery protocols (CDP) implementation guidelines; part 2: IP datacast over DVB-SH, ETSI TS 461 102, 1–64 (2010). [http://www.etsi.org/deliver/etsi\\_ts/102400\\_102499/102461/01.02.01\\_60/ts\\_102461v010201p.pdf](http://www.etsi.org/deliver/etsi_ts/102400_102499/102461/01.02.01_60/ts_102461v010201p.pdf). Accessed 27 Oct. 2015

8. J Calabuig, J Monserrat, D Gozalvez, D Gomez-Barquero, AL-FEC for streaming services in LTE E-MBMS. *EURASIP J. Wirel. Commun. Netw.* **2013**(1), 73 (2013)
9. H Chiao, K Li, H Sun, S Chang, H Hou, in *Application-layer fec for file delivery over the wimax unicast networks*. Proc. IEEE ICCT, (2010), pp. 685–688
10. M Baguena, CK Toh, CT Calafate, J-C Cano, P Manzoni, Rcdp: Raptor-based content delivery protocol for unicast communication in wireless networks for its. *J. Commun. Networks.* **15**(2), 198–206 (2013)
11. C Jiang, D Li, M Xu, LTP: An LT-code based transport protocol for many-to-one communication in data centers. *IEEE J. Select. Areas Commun.* **32**(1), 52–64 (2014)
12. M Allman, V Paxson, W Stevens, TCP Congestion control, RFC 2581 (1999). <https://tools.ietf.org/html/rfc2581>. Accessed 27 Oct. 2015
13. C Caini, R Firrincieli, TCP Hybla: a TCP enhancement for heterogeneous networks. *Int. J. Satell. Commun. Netw.* **22**(5), 547–566 (2004)
14. S Yousefi, T Chahed, SMM Langari, K Zayer, in *Comfort applications in vehicular ad hoc networks based on fountain coding*. Proc. IEEE VTC Spring, (2010), pp. 1–5
15. M Luby, in *Best practices for mobile broadcast delivery and playback of multimedia content*. Proc. IEEE BMSB, (2012), pp. 1–7
16. H-T Chiao, S-Y Chang, K-M Li, Y-T Kuo, M-C Tseng, in *WiFi multicast streaming using AL-FEC inside the trains of high-speed rails*. Proc. IEEE BMSB, (2012), pp. 1–6
17. M Luby, in *Raptor codes: algorithms and applications*. Proc. IEEE ICNC (Qualcomm Distinguished Lectures), (2012)
18. G Liva, E Paolini, M Chiani, Performance versus overhead for fountain codes over  $F_q$ . *IEEE Commun. Lett.* **14**(2), 178–180 (2010)
19. M Luby, A Shokrollahi, M Watson, T Stockhammer, L Minder, RaptorQ forward error correction scheme for object delivery, RFC 6330 (2011). <https://tools.ietf.org/html/rfc6330>. Accessed 27 Oct. 2015
20. T Stockhammer, A Shokrollahi, M Watson, G Michael Luby, Application layer forward error correction for mobile multimedia broadcasting case study. Digital Fountain (2009). <https://www.qualcomm.com/media/documents/files/raptor-codes-for-mobile-multimedia-broadcasting-case-study.pdf>. Accessed 27 Oct. 2015
21. M Angeles Vazquez Castro, F Vieira, Dvb-s2 full cross-layer design for qos provision. *IEEE Commun. Mag.* **50**(1), 128–135 (2012)
22. J Alins, J Mata-Diaz, JL Muñoz, E Rendón-Morales, O Esparza, XPLIT: A cross-layer architecture for tcp services over dvb-s2/etsi qos bsm. *Comput. Netw.* **56**(1), 412–434 (2012)
23. MZ Chowdhury, YM Jang, ZJ Haas, Call admission control based on adaptive bandwidth allocation for wireless networks. *J. Commun. Networks.* **15**(1), 15–24 (2013)
24. SA Khanjari, B Arafeh, K Day, N Alzeidi, Bandwidth borrowing-based qos approach for adaptive call admission control in multiclass traffic wireless cellular networks. *Int. J. Commun. Syst.* **26**(7), 811–831 (2013)
25. M Park, D Oh, in *Cross-layer design for improving tcp pep performance in dvb-rcs2 networks*. Proc. IEEE ICTC, (2013), pp. 846–847
26. K Lee, J Kim, in *Cross-layer design for tcp splitting connections with network coding in dvb-rcs networks*. Proc. IEEE ICTC, (2014), pp. 970–973
27. H Rutagemwa, S Xuemin, JW Mark, Transfer delay analysis of WAP 2.0 for short-lived flows. *IEEE Trans. Veh. Technol.* **56**(3), 1418–1426 (2007)
28. Riverbed Modeler . <http://www.riverbed.com>. Accessed 27 Oct. 2015

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---