Flexible Routing Algorithm for Digital Intermediate Frequency Interoperability Satellite Systems Using Fair Buffer Queuing

Walter Roensch Jr.
Electrical and Computer Engineering
Wichita State University
Wichita, KS, United States
W548V758@wichita.edu

Hyuck M. Kwon
Electrical and Computer Engineering
Wichita State University
Wichita, KS, United States
hyuck.kwon@wichita.edu

Amitav Mukherjee
Tiami Networks
Elk Grove, CA, United States
amukherjee@tiaminetworks.com

Khanh Pham
Air Force Research Lab
Kirtland Air Force Base, NM 87117 United States
khanh.pham.1@spaceforce.mil

***Abstract*—With changes in the Digital Intermediate Frequency Interoperability (DIFI) standard, there is now a standardized way for DIFI systems to communicate various flow control details. These details can be used to increase the reliability of satellite systems in terms of packet loss and delay. We will examine a deterministic routing scenario in which a satellite is used as a relay between two earth stations. To accomplish this goal, we propose a new queuing algorithm for network scheduling called "fair buffer queuing." We will also briefly discuss situations that would allow us to further leverage the changes from the updated flow control mechanisms included in the DIFI 1.2.0.**

I. Introduction

Digital intermediate frequency is a key technology in the world of flexible communication. The Digital Intermediate Frequency Interoperability (DIFI) standard takes this one step further, creating a message standard for all DIFI systems to ensure compatibility among them. This allows for a highly flexible communication system using minimal proprietary hardware.

Previously, the DIFI standard did not allow for the use of flow control, which limited the technologies that could be leveraged without proprietary implementation in the system. These new flow control packets allow for the creation of a generalized solution for satellite systems to minimize buffer overflow, and packet delay.

Many queuing and network scheduling algorithms could accomplish this goal, but for this project, we propose a unique solution to network scheduling made specifically for a DIFI system [5] [6]. We call this new queuing algorithm "fair buffer queuing." Fair queuing is an old technology first proposed in [3]. This class of queuing algorithm attempts to fairly distribute the resources of the system and was created to address problems with first in first out (FIFO) and priority queuing, where one device with high data rates could cripple the data rates of other devices and increase packet wait times significantly.

We propose an algorithm that distributes this fairness based on the number of packets waiting to be sent in the buffer. We determine this fairness based on a weighting factor. When a system can transmit data at multiple rates, it is possible to optimize the number of packets in the buffer so that the packet drop rate and packet waiting time can be minimized.

II. System Model

We consider the scenario shown in Fig. 1. This figure details our system model.

The following system model parameters are used:

- It is assumed that two separate uplink/downlink earth stations are communicating with a satellite. Each of these stations supports N user equipment (UE) running at randomly fluctuating data rates.
- The satellite is assumed to run at an intermediate frequency (IF) of 70 kHz, and it is assumed that each symbol has a duration of $2/(70 \times 10^3)$ seconds.
- It is assumed that $N/2$ of the UEs operate using quadrature phase shift keying (QPSK) and $N/2$ operate using 16-ary quadrature amplitude modulation (16QAM).
- It is assumed that there is no error correction coding, and fading is not taken into account for a simple demonstration of the efficiency of the proposed algorithm.
- It is assumed that each packet has a size of 14 symbols that each frame is made up of 512 packets. Between each frame there is a control frame.
- It is assumed that the system uses time division multiplexing (TDD) to dynamically allocate resources between the two earth stations.
- The system also uses TDD to allocate resources between N UE.

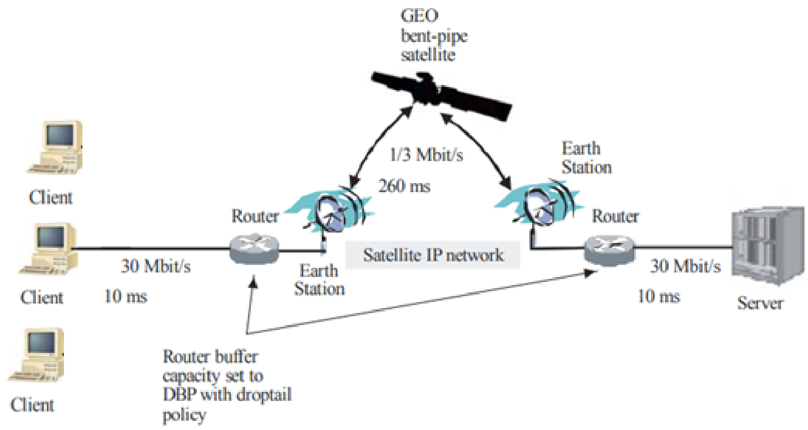

Fig. 1. Channel model.

- The satellite operates at a data rate of 0.1 mega bits per second (Mbps) when in 16QAM and 0.07 Mbps when in QPSK, and each of the N UE operates at a base data rate of $0.1/N$ Mbps for 16QAM and $0.07/N$ Mbps for QPSK.
- We consider a single-buffer design problem. We do not consider a server situation in which we have a service rate determined by a sever controller.

The transmitter and receiver are shown in Figs. 2 and 3, respectively. These systems are taken from [2].

To determine the buffer size, we used [1]. This buffer size is not entirely fit for a system of this specification, and for testing purposes, the buffer was scaled down 1,000 times to observe overflow within a shorter period of time. The buffer size in [1] is for an equivalent 5G system. In our buffer size calculation, the subcarrier spacing is set to $\Delta f = 15\ kHz$, and the minimum data rate is calculated as follows:

$$R_{min,PRB} = \frac{R_c}{1024} M_{order} N_{symbols} N_{subcarriers} \quad (1)$$

where R_c is the code rate assumed to be 0.5, and M_{order} is the modulation order of 2. This is due to the use of QPSK as the smallest order of the two (selected because we are determining minimum data rate), $N_{symbols} = 14$ due to 14 symbols per packet, and $N_{subcarrier} = 12$ per physical resource block (PRB), also referred to as the resource block (RB). Each PRB requires a bandwidth (BW) of $N_{subcarrier} \times \Delta f = 180\ kHz$.

We then determine $N_{PRB,20\ MHz}$, which is the number of PRBs in the 20 MHz BW with a 20 kHz guard-band as

$$N_{PRB,20\ MHz} = \frac{20\ MHz}{180\ kHz + 20\ kHz} = 100\ PRBs. \quad (2)$$

Finally, the minimum buffer size $N_{bits,buffer}$ to hold the number of bits per slot internal is determined using the following equation:

$$N_{bits,buffer} \geq R_{min,PRB} N_{PRB,20\ MHz} = 16.8\ Mbps. \quad (3)$$

We reduce this to 16.8 $kbps$ from 16.8 $Mbps$ so that we can quickly observe the difference between the proposed algorithm and the conventional system. We will compare the packet arrival rate and packet delay of the proposed algorithm with those of a system unable to use control frames to dynamically allocate the size of the frames or the amount of time given to each station by the satellite. In other words, a system that employs weighted fair queuing [6], with each queuing having equal weight. This is a suboptimal queuing solution, but acts as a baseline for comparison. We label this case conventional.

III. Proposed Algorithm

Figure 4 shows the proposed algorithm flow chart. We coin the term ”fair buffer queuing” for the proposed algorithm. The algorithm is designed to distribute the buffer size according to a weight array.

In Fig. 1, router 1 is on the left, and router 2 is on the right. We consider the total frame size equal to F, and N UE are in the described system. Also, we consider N_{B1} number of total packets waiting in the buffer of router 1 in Fig. 1, and N_{B2} in the buffer of router 2, as shown in Fig. 1. Also, let $N_{B1,UEi}$ packets belong to UEi in router 1 and $N_{B2,UEi}$ packets belong to router 2. Furthermore, let W_1 and W_2 be the weighting arrays at router 1 and router 2, respectively. Each is an array of length N. At every frame, router 1 receives $R_{B1,UEi}$ number of packets arrived for UE i, and router 2 receives $R_{B2,UEi}$ number of packets arrived for UE i. The sum of $R_{B1,UEi}$ is represented by R_{B1} for router 1 and R_{B2} for router 2. $F_{1,size,i}$ is an array that represents the number of packets allocated for each UE in the frame for router 1. Likewise, $F_{2,size,i}$ represents the same for router 2. Then, it can be written as

$$\sum_{i=1}^{N} [F_{1,size,i} + F_{2,size,i}] = F. \quad (4)$$

The objective of the routing algorithm is the calculation of these two arrays $F_{1,size,i}$ and $F_{2,size,i}$, $i = 1, ..., N$.

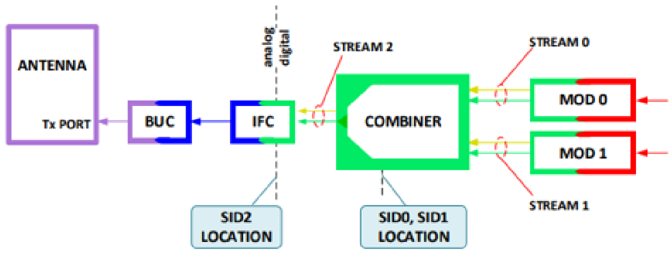

Fig. 2. Transmitter [2].

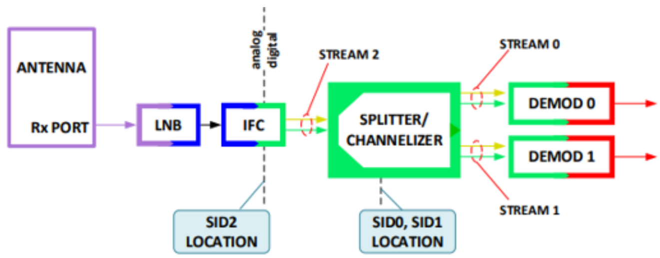

Fig. 3. Receiver [2].

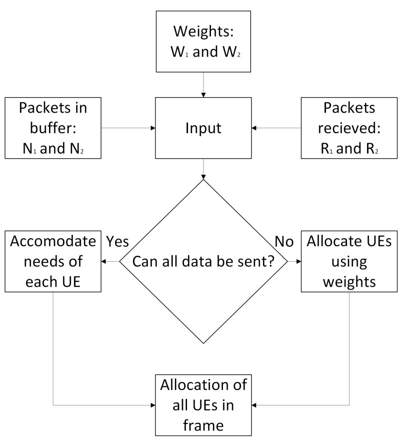

Fig. 4. Proposed algorithm flow chart.

In the use of the algorithm, there are two distinct cases. The first is case 1, where $N_{B1} + N_{B2} + R_{B1} + R_{B2} < F$. In this case,

$$F_{1,size,i} = N_{B1,UEi} + R_{B1,UEi} \tag{5}$$

$$F_{2,size,i} = N_{B2,UEi} + R_{B2,UEi}. \tag{6}$$

The needs of each UE are simply accommodated. Case 2 is the case of interest, $N_{B1} + N_{B2} + R_{B1} + R_{B2} \geq F$.

First, $\sum_{i=1}^{N} F_{1,size,i}$ and $\sum_{i=1}^{N} F_{2,size,i}$ must be calculated. They can be found as follows:

$$\sum_{i=1}^{N} F_{1,size,i} = F \frac{N_{B1} + R_{B1}}{N_{B1} + N_{B2} + R_{B1} + R_{B2}} \tag{7}$$

$$\sum_{i=1}^{N} F_{2,size,i} = F \frac{N_{B2} + R_{B2}}{N_{B1} + N_{B2} + R_{B1} + R_{B2}}. \tag{8}$$

Next, each element of $F_{1,size,i}$ and $F_{2,size,i}$ is calculated as follows:

$$F_{1,size,i} = N_{B1,UEi} + R_{B1,UEi} \tag{9}$$
$$-W_{1,i}((N_{B1} + R_{B1}) - \sum_{i=1}^{N} F_{1,size,i})^2$$

and

$$F_{2,size,i} = N_{B2,UEi} + R_{B2,UEi} \tag{10}$$
$$-W_{2,i}((N_{B2} + R_{B2}) - \sum_{i=1}^{N} F_{2,size,i})^2.$$

The algorithm maintains the ratio of buffered packets according to the W array. That is the basic function of the algorithm. The main benefit of this algorithm is the ability to decrease packet dropping. This is because the system tries to fairly distribute buffer size.

The algorithm is summarized in two steps: First, (7) and (8) are used to calculate the next step; and step two gives us the amount of resources allocated to each UE in the frame using (9) and (10).

IV. Results

Figure 5 shows the modulation of signals, along with the received signal and demodulation signal in both QPSK and 16QAM. All further results use this as the basis for transmission and receiving. The system runs at a rate of 0.07 $Mbps$ at QPSK and 1.05 $Mbps$ at 16QAM.

We will examine three cases: 2 UE, 4 UE, and 8 UE. First, we will examine the 2 UE case. In this case, each UE makes a rate request at every frame, with the rate between 0.042 and 0.058 $Mbps$. The results are shown in Figs. 6 and 7, respectively. It can be seen that the improvement over the conventional case is significant. For example, Fig. 6 shows

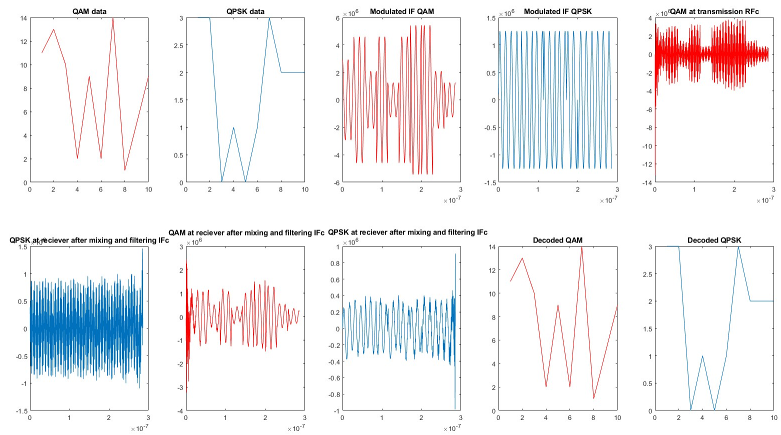

Fig. 5. TDD system with mixed IF modulation and demodulation of QPSK and 16QAM.

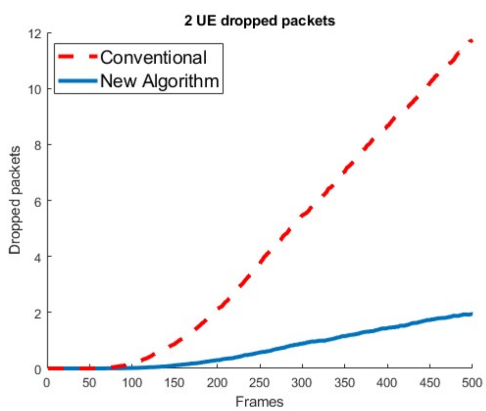

Fig. 6. 2 UE dropped packets.

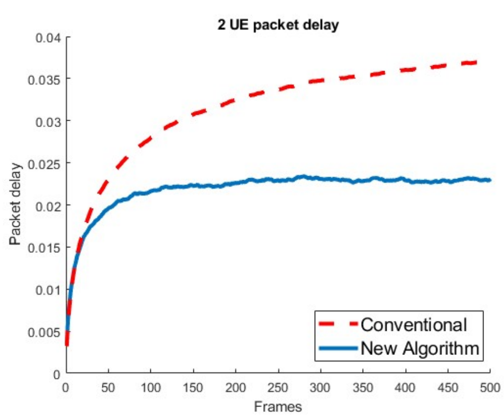

Fig. 7. 2 UE packet delay.

that the proposed DIFI algorithm can reduce the number of packets dropped due to the overflow at the buffer from 12 to 2 at frames 500. This is 6.038 times improvement over the conventional case. In addition, Fig. 7 shows that the proposed DIFI algorithm can reduce the packet delay at the buffer from 0.0375 second to 0.02 second at frames 500, which is 1.875 times improvement over the conventional case. Next, we will examine the 4 UE case. In this case, each UE makes a rate request at every frame, with a rate between .017 and .033 $Mbps$. These results are shown in Figs. 8 and 9. The improvement over the conventional case is a bigger improvement than 2 UE case. For example, Fig. 8 shows 26.43 times smaller number of dropped packets than the conventional case. Also, Fig. 9 shows 2.239 times smaller packet delay than the conventional case. This trend continues to the 8 UE case. It can be seen that as the number of UEs increases, so does the performance gap. This is due to the increased flexibility of the new algorithm.

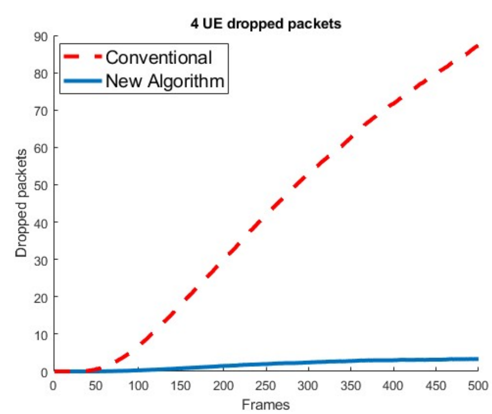

Fig. 8. 4 UE dropped packets.

Next, we will examine the 8 UE case. In this case, each UE makes a rate request every at frame, with a rate between 0.0065 and 0.0205 $Mbps$. These results are shown in Figs. 10 and 11.

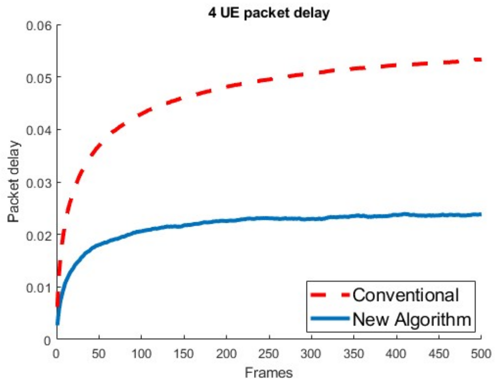

Fig. 9. 4 UE packet delay.

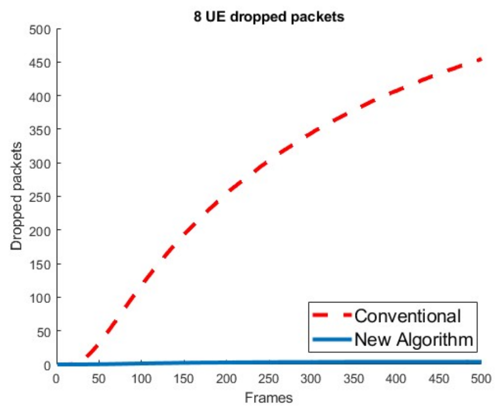

Fig. 10. 8 UE dropped packets.

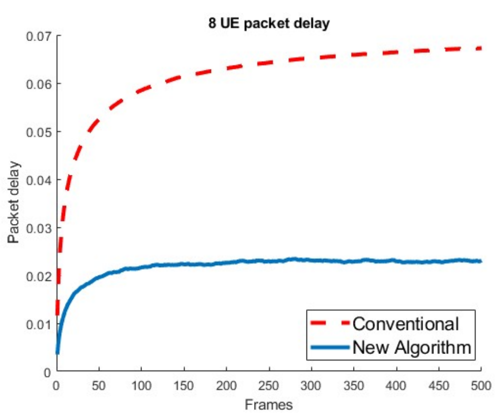

Fig. 11. 8 UE packet delay.

Again, the improvement is greater than the previous 4 UE case. For example, Fig. 10 shows that the proposed DIFI algorithm can significantly reduce the number of packets dropped due to overflow. The proposed algorithm reduces it by 117.616 times when compared to the conventional case. In addition, Fig. 11 shows that the proposed DIFI algorithm can reduce the packet delay 2.926 times shorter than the conventional case.

V. Future work

More complex multi-satellite scenarios are worth investigating. Additions to DIFI would allow the leveraging of complex routing algorithms in non-deterministic routing scenarios. Back-pressure routing [4] [7], in particular, would be an interesting point of comparison. We can consider, for example, a system where multiple earth stations could send data through multiple satellites. In this circumstance, determining proper routing of data by which uplink would be complex.

VI. Conclusion

The new changes to DIFI could provide the ability to easily optimize data transfer in complex multi-modulation systems. The new algorithm displayed in this paper combined with the improved DIFI standard could create an optimized system that can be implemented in any modern DIFI system. The performance benefit over the conventional is marked, and the gap widens as the number of UE increases.

VII. Acknowledgement

This work was supported in part under the FA945324PX008 project. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force.

References

[1] F. Rinaldi, A. Raschellà, and S. Pizzi, “5G NR system design: A concise survey of key features and capabilities,” Wireless Network 27, 5173–5188 (2021). https://doi.org/10.1007/s11276-021-02811-y.

[2] Digital Intermediate Frequency Interoperability (DIFI) Consortium, IEEE-ISTO Std 4900-2021: Digital IF Interoperability Standard, v1.2.0, August 2023.

[3] John Nagle: “On packet switches with infinite storage,” RFC 970, IETF, December 1985.

[4] L. Tassiulas and A. Ephremides, “Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[5] Ito, Y., Tasaka, S., and Ishibashi, Y. (2002). “Variably weighted round robin queuing for core IP routers,” Conference Proceedings of the IEEE International Performance, Computing, and Communications Conference (Cat. No.02CH37326), p. 159.

[6] Bennett, J. C. R. and Hui Zhang (1996), “WF/sup 2/Q: Worst-case fair weighted fair queuing,” Proceedings of IEEE INFOCOM ’96, Conference on Computer Communications, vol. 1, p. 120. doi:10.1109/INFCOM.1996.497885. ISBN 978-0-8186-7293-4. S2CID 17558577.

[7] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, “Routing without routes: The backpressure collection protocol,” Proceedings of 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), April 2010.