2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Adaptive Forward Error Correction with Adjustable-Latency QoS for Robotic Networks

Ryan J. Marcotte Edwin Olson

***Abstract*— Mobile robotic teams require robust communication in order to coordinate effectively, which is a challenge given the dynamic, unpredictable nature of mobile ad hoc networks (MANET). These networks are subject to rapidly varying link qualities as robots move through their environment. Improving the robustness of these point-to-point links leads to greater overall network performance, which in turn allows the robots to perform their mission more effectively.**

In this work, we present a forward error correction (FEC) technique that exploits latency tolerance in network traffic to provide consistent packet delivery performance even on low-quality links. Our proposed system estimates link quality based on recent packet reception history and uses that estimate to determine FEC encoding strength. Furthermore, this system provides a novel Quality of Service (QoS) mechanism that trades latency tolerance for more reliable, lower overhead transmission. We evaluate the effectiveness of this technique in a real-world robotic testbed.

I. Introduction

Robust communication is necessary in order for mobile robotic teams to coordinate effectively. Typically, these teams utilize wireless radios that form ad hoc networks, which are challenging to design due to their dynamic network topology and lack of fixed physical infrastructure. Much research has been devoted to developing efficient routing protocols to marshal packets through the network. In most cases, these routing protocols monitor the state of links in the network, yet they lack tools to actually modify the performance of a given link. This work presents a technique that complements existing routing protocols by extending the effective range of links through automatic FEC adjustments.

Traditional wired and wireless networks are designed to support high-fidelity connections between nodes. Robotic networks, on the other hand, may be forced to function with links that are less reliable. As robots move through their environment performing tasks, they may extend communication links beyond their reliable range, either out of necessity or through ignorance of the situation. When this occurs, the links enter a transitional region in which performance is unreliable and unpredictable. Intermediate links in this state are not yet broken, but they do not support robust communication. In a robotic network, it is important to maintain useful connections over these transitional regions if only to command the robot to turn around.

In this paper, we present an approach to improving the performance of intermediate links in order to extend the range of robotic networks. This FEC technique can exchange available bandwidth and some tolerance to latency for a consistent packet reception ratio (PRR) throughout the range of the link. Specifically, the contributions of this paper include:

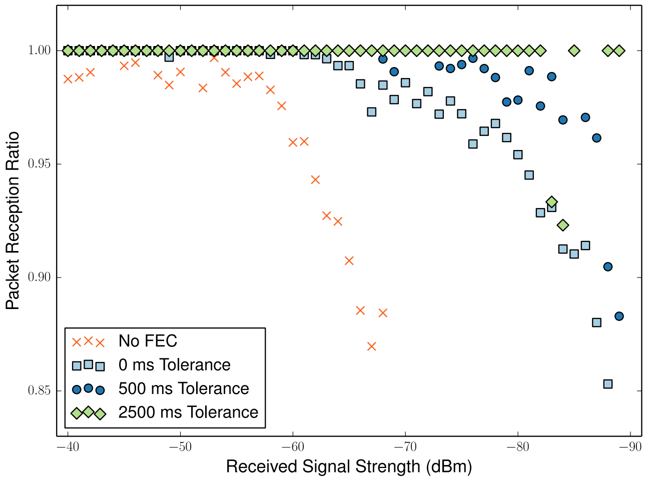

Fig. 1: Effective packet reception ratio with FEC applied to links with varying levels of latency tolerance. Our system allows applications to exchange latency tolerance for a consistent reception rate and lower FEC overhead throughout the range of a link. The FEC system shown here was targeting an effective PRR of 0.99. Note the use of a limited-range axis for clarity.

- A QoS mechanism that allows applications to dynamically exchange latency tolerance for improved communication reliability,
- A probabilistic technique for the online adaptation of FEC encoding strength based on the estimated quality of a link, and
- Extensive evaluation of this technique in real-world indoor and outdoor environments.

II. Related Work

The task of improving robustness and extending range in robotic network links can be approached from a variety of angles. Some techniques can be borrowed from the literature of traditional wireless networks, whereas others are unique to the realm of robotics.

A. Traditional Wireless Networks

Wireless networks are susceptible to significant packet loss and corruption due to environmental factors (e.g. multipath effects) and interference. Most existing wireless networking stacks, such as IEEE 802.11 or 802.15.4, have mechanisms in place to mitigate these effects. These include the following:

- Cyclic redundancy checks to ensure the integrity of data as it is transmitted and received,

The authors are with the Computer Science and Engineering Department, University of Michigan, Ann Arbor, MI 48104, USA. {ryanjmar,ebolson}@umich.edu

- Delivery acknowledgements and retransmissions, such as in link-layer unicast and TCP,
- Flow control at both the hardware and transport layers that seeks to prevent packet collisions, and
- Rate selection algorithms such as Adaptive Multirate Auto Rate Fallback (AMARF) that attempt to optimize bitrate based on link quality, reverting to lower rates as connectivity decreases [1].

In addition to these standard measures, much work has been done to apply error correcting codes to wireless communications through FEC. The general schema of utilizing FEC in this domain is that by including additional redundancy information in transmissions, the system can recover lost data. This has seen numerous applications, from more reliable video streaming [2] to a more stable platform for TCP in environments with high packet loss [3].

One particular form of FEC that is particularly relevant to our work is interleaving. It is well-known that both bit errors and packet loss have a strong temporal correlation [4]. By applying FEC and then interleaving the resultant data chunks before transmission, it is possible to achieve better temporal diversity and improve robustness to burst errors. This technique has been explored at both the bit level [5] and packet level [6]. The main drawback to interleaving is that it necessarily adds latency into transmissions, since all interleaved data must be received before the recovery process can take place. We will show that our FEC technique shares many of the benefits of interleaving but does not impose the same latency drawback.

B. Robotic Networking

One key way in which mobile robotic networks vary from their traditional counterparts is in the ability of nodes to take an active role in changing the performance of the network. In particular, robotic nodes are capable of transporting themselves through their environment. Though this comprises much of the challenge of designing these networks, it also presents some opportunities if network connectivity is factored into motion planning decisions.

Many researchers have examined how robotic teams can coordinate amongst themselves to maintain connectivity throughout their missions. Rooker and Birk present an exploration algorithm that allows a robotic team to maintain connectivity with one another and optionally with a fixed ground station [7]. Tardioli et al. combine a novel distributed control algorithm, task allocation schema, and network-layer link monitoring system to enforce network connectivity [8]. Michael et al. present a decentralized and asynchronous control algorithm for maintaining a connected network, including experimental validation on a physical robotic testbed [9].

In order for connectivity management systems like this to be more effective, it is important to be able to estimate spatial variations in communication signal strength. Mostofi et al. utilize a small number of signal strength measurements to extrapolate channel performance [10]. Strom and Olson present an algorithm that combines signal strength information with data from additional sensors, such as range data from LIDAR, to infer the locations of communication barriers in the environment [11].

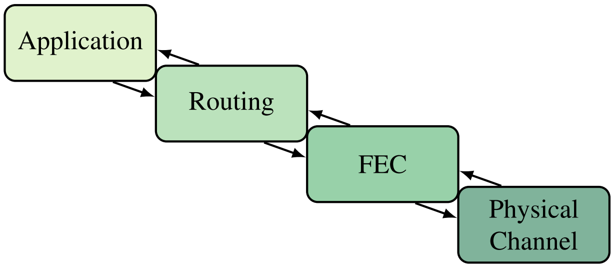

Fig. 2: The logical position of the FEC module in a network stack. The FEC module is situated between a MANET routing protocol and the physical communication channel.

Another promising area of research for improving robustness of robotic network links is cognitive radio (CR). CR techniques exploit the flexiblity of software defined radios to utilize the wireless medium more efficiently, opportunistically switching between channels and frequency bands. Chowdhury and Felice present a routing protocol for mobile cognitive radio ad-hoc networks that jointly selects routes and channels to avoid spectral congestion [12]. Cacciapuoti et al. modify the widely used AODV routing protocol to take advantage of CR technology [13]. Guan et al. use CR to predict the duration of links, information that then informs routing decisions [14].

Our technique builds on much of this existing work. Our approach to forward error correction draws inspiration from packet interleaving but only incurs additional latency when packets must be recovered from their redundancy information. We also utilize two methods prevalent in the robotics domain for estimation, Kalman filtering and Monte Carlo simulation, to perform online link-state estimation and inform FEC encoding strength decisions.

III. Approach

Our FEC module is designed to complement existing MANET routing protocols and any mechanisms already implemented in networking stacks to improve link robustness. The FEC module is logically situated between a routing module and the physical transmission medium. This arrangement could place the module at the link layer or at the application layer if routing occurs in a user space program. In this section, we explore the various components of our FEC system.

A. Reed-Solomon Coding

Reed-Solomon codes are a class of error-correcting codes first presented by Reed and Solomon in the early 1960s [15]. While Reed-Solomon codes are widely used for correcting both errors and erasures, our use case only requires the latter, since we assume the integrity of received packets is verified by a cyclic redundancy check. An (n, k) Reed-Solomon code is used to produce $m = n - k$ parity symbols from k data symbols. When correcting erasures, an (n, k) Reed-Solomon code has the property that if any k of the n transmitted symbols are received successfully, then all of the n symbols

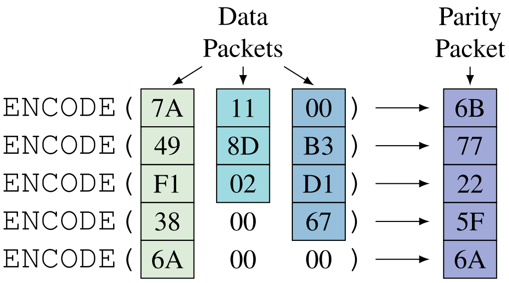

Fig. 3: The batch encoding process, using a $(4,3)$ Reed-Solomon code. The payloads of the three original packets in the batch are encoded to produce a single parity packet.

can be fully recovered. This property is the foundation on which our system operates.

B. Batch Formation and Encoding

The FEC module handles the encoding of groups of packets on the transmission side and the decoding of those groups on the reception side. We refer to these groups of packets as *batches*. At the transmitter, the FEC module is continually in the process of forming new batches.

As packets are input into the FEC module, they are associated with the batch that is currently being formed. These packets are prepended with a small header containing the batch's unique identification number and the position of the packet within the batch. Data packets are then immediately transmitted, ensuring that no additional latency is induced onto them. The FEC module temporarily retains a copy of the data packet payloads, which are used during the encoding process to generate parity packets.

Formation of a batch continues until a *timeout* occurs (to be discussed further in Section III-C), at which point, the parity packets are encoded. This encoding is illustrated in Fig. 3. For $i = 1, 2, \ldots, l_{max}$, where l_{max} is the maximum payload length of the data packets, the i^{th} bytes of the data packets are concatenated to form a codeword. For batches with packets of varying sizes, a zero byte is substituted when i exceeds the length of a packet. For a batch of k data packets and $m = n - k$ parity packets, an (n, k) Reed-Solomon code is used to generate m parity bytes from that codeword. These m parity bytes form the i^{th} bytes of the parity packets. Once the parity packets have been generated, they are prepended with a header containing the batch identification number and their position in the batch, as well as the number of data and parity packets in the batch, which will be needed by the decoder in order to select the proper Reed-Solomon code. The parity packets are then transmitted, completing the batch formation and encoding process

C. Latency Considerations

As we will show in Section IV-C, the FEC system provides routing protocols or other applications with a useful QoS mechanism, namely the ability to specify desired bounds on both latency tolerance and target packet reception rate. The desired latency tolerance level determines the duration of the previously described batch formation process.

Data packets only experience a delay in reception when they must be recovered from parity packets. The maximum next-hop latency a packet may experience is equal to the time interval between that packet's transmission and the reception of the parity packets that are used for recovery, plus a nominal amount of time for decoding. A timeout occurs once the maximum latency period has elapsed since transmission for any packet in the batch. This timeout signals the end of the batch formation process.

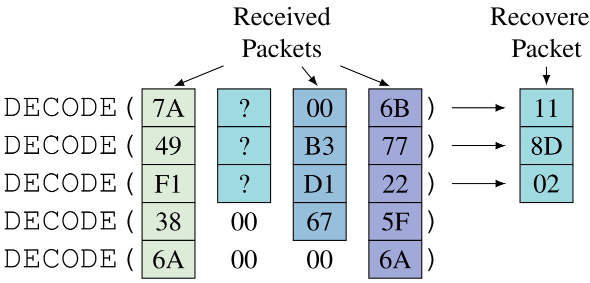
Received Packets
Recovered Packet
DECODE (7A ? 00 6B) → 11
DECODE (49 ? B3 77) → 8D
DECODE (F1 ? D1 22) → 02
DECODE (38 00 67 5F)
DECODE (6A 00 00 6A)

Fig. 4: The batch decoding process for the batch encoded in Fig. 3. The second original packet is not received, but the decoder is able to recover its contents using the other received packets.

D. FEC Encoding Strength Calculation

The purpose of the FEC encoding strength calculation is to determine the optimal amount of parity m to transmit along with a batch of size k given a raw PRR estimate of mean $\hat{x}$ and variance σ^2 in order to achieve a target effective PRR. We will show our method for generating link quality estimates in Section III-E. The target effective PRR is specified by the application and is the desired expected value of effective PRR. Due to the uncertainty that is always present in link quality estimations, a target effective PRR of 1.0 is not achievable.

Algorithm 1 details the encoding calculation. Beginning with 0, the CALCENCODESTRENGTH procedure iteratively proposes parity packet amount m to transmit along with the data packets. For each m, an expected distribution of received packets from the $n = k + m$ transmitted packets is obtained through Monte Carlo simulation. The algorithm then draws from this distribution to compute the expected value of the effective PRR for that parity amount. If this expected value is greater than or equal to the target, CALCENCODESTRENGTH returns m. Otherwise, the process continues until m reaches its maximum value, which can be set based on bandwidth or power consumption considerations.

The use of Monte Carlo simulation allows for the calculation of PRR distributions for any arbitrarily complex channel estimation model. Though a closed-form calcuation may be possible for some simple channel models, the Monte Carlo simulation generalizes well and supports models in which the packet loss rate is subject to some uncertainty.

E. Link Quality Estimation

In order to make the previously described FEC encoding strength calculation, the transmitting node must have an estimate of the quality of the link. Signal strength values, receipt acknowledgments, and retry counts can all contribute to this estimate, but these indicators are often not easily available in user space. Furthermore, some types or modes of

Algorithm 1 FEC Encoding Strength Calculation

```
procedure CalcEncodeStrength($k$, $\hat{x}$, $\sigma^2$, $target$)
    for $m = 0, 1, \ldots, m_{max}$ do
        $n \leftarrow k + m$
        $prr \leftarrow 0$
        $dist[n+1] \leftarrow$ MonteCarlo($n$, $\hat{x}$, $\sigma^2$)
        for $i = 0, 1, \ldots, n$ do
            if $i >= k$ then
                $res \leftarrow 1.0$
            else
                $res \leftarrow i/n$
            $prr \leftarrow prr + dist[i] * res$
        if $prr >= target$ then
            return $m$
    return $m_{max}$
```

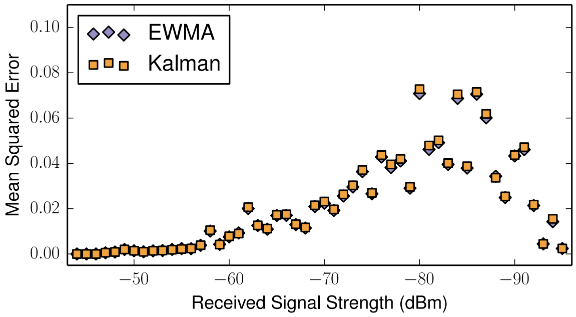

Fig. 5: Mean-squared error for PRR predictions of proposed Kalman filter-based method and standard exponential weighted moving average technique. Our method performs similarly to EWMA in predicting raw PRR while providing meaningful estimates of uncertainty.

networking stacks may not even support these metrics, such as an IEEE 802.11 radio operating in broadcast mode. Therefore, it is desirable to be able to estimate link quality simply by tracking received packets that contain sequence numbers. If a link is symmetric and both nodes are sending traffic to one another, link quality estimates for a transmission can be derived from the transmitter's recent reception history. It has been shown, though, that wireless links are generally not symmetric [16]. For this reason, our system calculates link quality estimates at the receiver and then feeds those estimates back to the transmitter either through dedicated status packets or encapsulated in the headers of other traffic.

An exponential time-weighted moving average (EWMA) is commonly used to estimate the average raw PRR of a link based on recent performance [17]. While this mechanism is reasonably accurate in its prediction of mean PRR, it fails to indicate uncertainty about that estimate. Understanding the confidence level of a raw PRR estimate is critical in computing optimal encoding strengths, since it determines how liberally the FEC system should act in sending additional parity packets. Our system uses a Kalman filter to estimate both the mean and the variance of raw PRR. A one-dimensional Kalman filter is functionally equivalent to EWMA in terms of state estimation, while at the same time providing the necessary uncertainty information. Fig. 5 shows the similar performance of the Kalman filter- and EWMA-based approaches at predicting PRR throughout our FEC experimentation.

To determine the variance of observational noise Q and process noise R in our system, we conducted simple experiments in which we measured variations in PRR for both stationary and mobile nodes. These experiments and their results will be discussed in Section IV-B.

The system performs the following prediction of the mean $\hat{x}_k$ and variance P_k of raw PRR at a regular rate:

$$\hat{x}_k = \hat{x}_{k-1} \tag{1}$$

$$P_k = P_{k-1} + Q \tag{2}$$

The update step of the Kalman filter is also performed at equal intervals based on the observation of received and lost packets during the previous interval. The sequence number of a received packet can be used to infer any packet loss that may have occurred between its reception and that of the last received packet. The mean $\hat{x}'_k$ and variance P'_k of the update step for an observed PRR of z are given by

$$\hat{x}'_k = \hat{x}_k + \frac{P_k}{P_k + R}(z - \hat{x}_k) \tag{3}$$

$$P'_k = \frac{R}{P_k + R} \tag{4}$$

F. Batch Reception and Decoding

Immediately after it is received from the physical channel, a packet is passed into the FEC module. If a packet's payload is not encoded (i.e. it is a data packet), the packet is immediately passed to the routing module. Copies of the contents of all received packets are stored temporarily to be used in the batch decoding process. Once any k of the batch's n packets have been received, the entire batch can be decoded. If any of the data packets were not received, the entire batch is passed through a decoder, as illustrated in Fig. 4. The decoding procedure is symmetric to the encoding process previously described, except that all of the data can be reconstructed from only k of the n transmitted packets. For $i = 1, 2, \ldots, l_{max}$, byte i of each of the received packets is inserted into a buffer. The index of each byte in the buffer is determined by the position of the corresponding packet in the batch. By passing this buffer into an (n, k) Reed-Solomon decoder, the m erasures can be corrected. Through this erasure-correction process, the contents of any missing packets can be recovered. Once the decoding process is complete, recovered packets are passed to the routing module.

IV. Evaluation

We have evaluated our FEC technique in a real-world robotic system to evaluate its effectiveness at making network links more robust. The main purpose of this evaluation is to show that our system is capable of achieving a target effective PRR and to show that latency tolerance can be used as a QoS parameter to achieve better effective PRR performance.

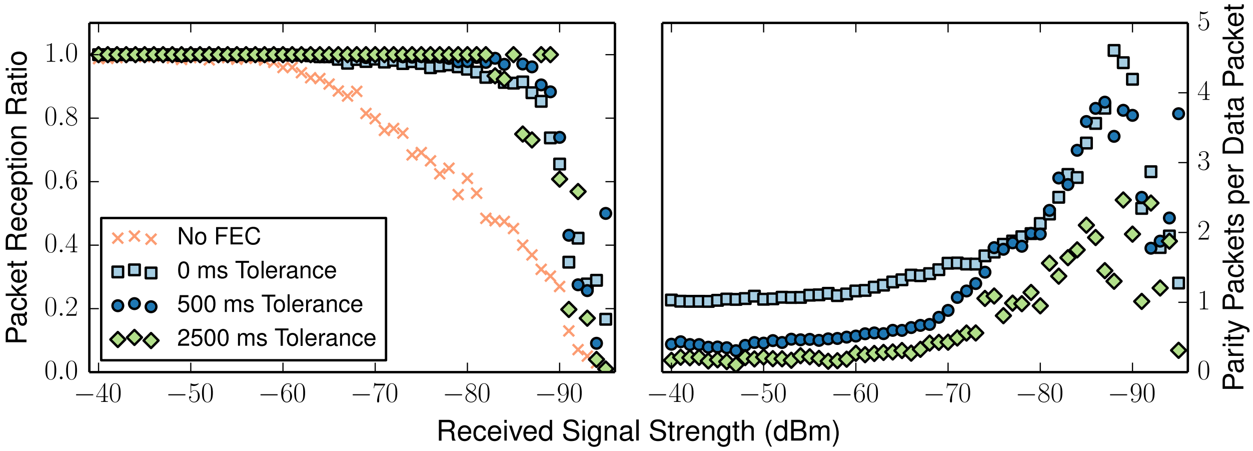

Fig. 6: PRR and overhead performance of FEC applied to links with varying levels of latency tolerance. Utilizing available tolerance to latency provides dual performance benefits: PRR levels are more consistently maintained through the range of link qualities, and the overhead cost of the system is reduced. The FEC system shown here was targeting an effective PRR of 0.99. See Fig. 1 for a closer look at PRR performance.

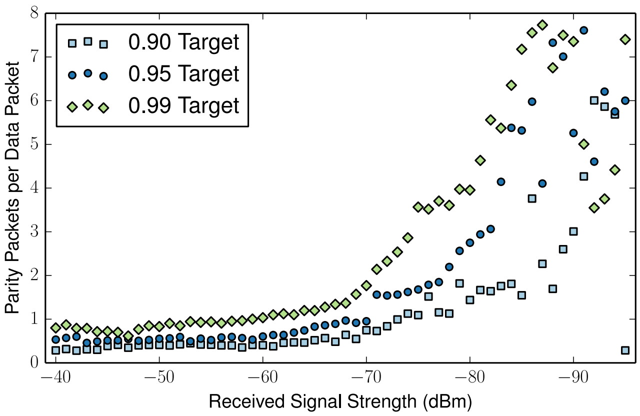

Fig. 7: Parity packets per transmitted data packet to achieve target effective PRR. Achieving a higher level of PRR performance requires more bandwidth overhead.

A. Test Apparatus

Our test platform was a robotic team previously designed for urban reconnaissance [18]. Because we are targeting the performance of single network links, we performed our evaluation with a single mobile robot and stationary ground station. Both the robot and the ground station were outfitted with a 2.4-GHz Open-Mesh OM2P-HS Wi-Fi radio. Network traffic in robotic systems is usually fairly regular, with sensor data such as camera images or LIDAR scans being shared at fixed rates. To simulate this type of regular traffic, we transmitted 1000-byte packets from the robot back to the ground station at a frequency of 10 Hz. The radios were configured to operate in IEEE 802.11 broadcast mode, transmitting at a 1 Mbps bitrate. This means that our data traffic consumed about 10% of the available channel bandwidth, accounting for some overhead at the PHY layer.

We initially planned to evaluate the performance of our FEC system at various mobility levels. After some testing, though, we found that there was no significant variation in the performance of the system within the mobility range of our test robots, which have a top speed of approximately 2 m/s. We fixed the robot speed at 0.5 m/s during the rest of our experimentation, which is a reasonable mobility level for a robotic team performing an exploration or mapping mission.

Real-world robotic teams must be capable of operating in both indoor and outdoor environments, so we included both in our testing. To test our FEC system, we simulated a realistic exploration or mapping mission, with the robot moving through its environment without any consideration for network connectivity in its motion planning, even moving entirely out of range at times. This allowed us to evaluate the performance of our method on links of varying quality.

B. Kalman Filter Parameter Measurement

Before we could evaluate our FEC system, we first needed to measure the variance of observational noise and process noise in our system for use in the link quality Kalman filter. For this experiment, we recorded one minute segments of communication using the previously described traffic model. During these trials, both the transmitting and receiving nodes were stationary. We found the PRR for each second of the trial and computed the mean variation between time steps. By computing the mean value of these variations for trials covering a range of link qualities, we found an approximation for the variance of observational noise in the system, $Q = 0.06$.

We then conducted a second set of tests in which the robotic node traveled through its environment at one of several fixed speeds. Using the same procedure as before, we computed the mean value of the variations in PRR while the robot was in motion. Taking the difference between this value and the variance in observational noise, we arrived at the variance of process noise, $R = 0.02$.

C. FEC System Testing

We evaluated our FEC system through two experiments. In the first, we assigned a latency tolerance of 500 ms to all packets being transmitted, resulting in parity being computed

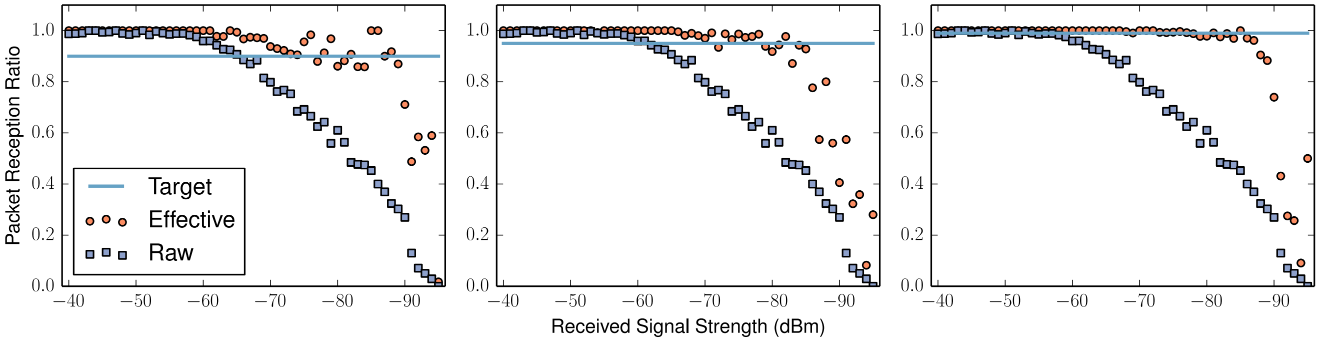

Fig. 8: Effective PRR achieved by our proposed FEC system for various target performance levels. The raw PRR declines as link quality decreases, but the FEC system is able to track a target effective PRR level. See Fig. 7 for information on overhead levels required to achieve these improvements.

on batches of five data packets. We varied the target effective PRR that the system was trying to achieve, setting goals of 0.99, 0.95, and 0.90. The results of this experiment are shown in Fig. 8. The FEC system can track the target effective PRR reasonably well across the spectrum of link qualities, which is a significant improvement over the raw PRR. As Fig. 7 illustrates, achieving high effective PRR targets costs more in terms of overhead.

In our second test of the FEC system, we fixed the target effective PRR at 0.99 and varied the level of latency tolerance. The results of this test are shown fully in Fig. 6 and are highlighted in Fig. 1. Increasing the latency tolerance yields dual benefits. First, it improves the PRR performance by increasing the time diversity of the transmissions, making the link more robust to burst errors. This is the same result achieved through the packet interleaving methods described in Section II. Unlike packet interleaving, though, our FEC system only incurs latency when packets are lost and must be recoverd.

The second result of increasing latency tolerance is reduced overhead in terms of bandwidth usage. Due to the uncertainty present in the estimation of link quality, encoding strength decisions are necessarily conservative. As latency tolerance increases, so do the sizes of the batches being encoded. This allows for a finer granularity in encoding strength decisions. Even with an estimated link quality of 100%, the system will still elect to send parity packets due to the uncertainty of the estimate. In the extreme case of a single-packet batch, adding a parity packet results in twice the bandwidth usage.

V. Conclusion

We have presented a novel system for applying adaptive forward error correction to robotic network links and have characterized the performance of this system using a real-world robotic testbed. We have demonstrated that by exploiting latency tolerance in network links, the FEC system can attain improvements to both the effective PRR and the amount of bandwidth overhead required to realize that result.

Our FEC system can greatly benefit robotic networks by extending the effective range of their links and providing more consistent performance for those links. This would allow robotic teams to cover larger areas than they were previously capable of without requiring modifications to hardware.

References

[1] Y. Xi, B. Kim, J. Wei, and Q. Huang, “Adaptive multirate auto rate fallback protocol for IEEE 802.11 WLANs,” in *Military Communications, IEEE Conference on*, 2006.

[2] P. Huang, K. Chu, H. Lo, W. Lee, and T. Wu, “A novel adaptive FEC and interleaving architecture for H.264/SVC wireless video transmission,” in *Intelligent Information Hiding and Multimedia Signal Processing, International Conference on*, 2009.

[3] B. Liu, D. Goeckel, and D. Towsley, “TCP-cognizant adaptive forward error correction in wireless networks,” in *Global Telecommunications Conference, IEEE*, 2002.

[4] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, “Measurement and modelling of the temporal dependence in packet loss,” in *Computer and Communications Societies*, 1999.

[5] L.-J. Chen, T. Sun, M. Sanadidi, and M. Gerla, “Improving wireless link throughput via interleaved FEC,” in *Computers and Communications, Ninth Int. Symposium on*, 2004.

[6] A. Botta and A. Pescape, “IP packet interleaving: Bridging the gap between theory and practice,” in *Computers and Communications, IEEE Symposium on*, 2011.

[7] M. Rooker and A. Birk, “Multi-robot exploration under the constraints of wireless networking,” *Control Engineering Practice*, 2007.

[8] D. Tardioli, A. Mosteo, L. Riazuelo, J. Villarroel, and L. Montano, “Enforcing networking connectivity in robot team missions,” *Robotics Research, International Journal of*, 2010.

[9] N. Michael, M. Zavlanos, V. Kumar, and G. Pappas, “Maintaining connectivity in mobile robot networks,” in *Experimental Robotics*, Springer Berlin Heidelberg, 2009.

[10] Y. Mostofi, M. Malmirchegini, and A. Ghaffarkhah, “Estimation of communication signal strength in robotic networks,” in *Robotics and Automation, IEEE International Conference on*, 2010.

[11] J. Strom and E. Olson, “Multi-sensor ATTenuation Estimation (MATTE): Signal-strength prediction for teams of robots,” in *Intelligent Robots and Systems, International Conference on*, 2012.

[12] K. Chowdhury and M. Felice, “Search: A routing protocol for mobile cognitive radio ad-hoc networks,” *Computer Communications*, 2009.

[13] A. Cacciapuoti, C. Calcagno, M. Caleffi, and L. Paura, “CAODV: Routing in mobile ad-hoc cognitive radio networks,” in *Wireless Days, 2010 IFIP*, 2010.

[14] Q. Guan, F. Yu, S. Jiang, and W. G., “Prediction-based topology control and routing in cognitive radio mobile ad hoc networks,” *Vehicular Technology, IEEE Transactions on*, 2010.

[15] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *Journal of Industrial and Applied Mathematics*, 1960.

[16] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Models and solutions for radio irregularity in wireless sensor networks,” *ACM Transactions on Sensor Networks*, 2006.

[17] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, “Radio link quality estimation in wireless sensor networks: A survey,” *ACM Transactions on Sensor Networks*, 2012.

[18] E. Olson, J. Strom, R. Goeddel, R. Morton, P. Ranganathan, and A. Richardson, “Exploration and mapping with autonomous robot teams,” *Communications of the ACM*, 2013.