IPN Progress Report 42-217 • May 15, 2019

Adaptive Control of a Variable Coded Modulation Radio Using Delay Tolerant Networking Licklider Transmission Protocol Statistics

J. Leigh Torgerson,* Gregory J. Miles,* Nate J. Richard,* and Hua Xie*

ABSTRACT. — In this report, we describe proof-of-concept test results for the use of the Disruption-Tolerant Networking (DTN) Licklider Transmission Protocol (LTP) for the automatic determination of optimum coding and modulation strategies for the JPL Space Telecommunications Radio System (STRS) that is on the ISS as part of the SCaN Testbed. Under a SCaN Technology task, Variable Coded Modulation (VCM) modes [1] were implemented in an FPGA system for installation in the STRS radio on the ISS. These modes, described in [1], consist of combinations of modulation types (BPSK through 64APSK) and LDPC, C2, and Turbo codes of various rates. The flight test of several modulation types in the JPL STRS radio was reported in IPN Progress Report 42-212, dated February 15, 2018. Subsequent ground tests on the STRS field programmable gate array (FPGA) hardware used for development of the VCM capabilities successfully demonstrated the VCM coding and modulation strategies defined in [1]. Given the large number of new coding and modulation combinations in the specification, a means of automating selection of the appropriate mode is desired to make the system truly adaptive to changing link conditions. In this report, we describe the emulation experiments conducted using LTP segment retransmission statistics to make decisions in the radio as to which VCM mode is optimum, showing the data throughput trades between LTP retransmissions on a lossy channel and changing to a lower-rate code to eliminate errors.

I. Introduction

VCM is a digital communication methodology that allows for the change of coding and modulation in the course of a communication session in order to adapt the underlying information data rate to dynamic link conditions. In contrast to a conventional communication system that uses a fixed coding and modulation scheme designed to accommodate the worst-case link conditions, VCM can significantly increase the effective

* Communications, Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

data throughput when the transmitter is appropriately adaptively configured to fully utilize link capacity.

The Consultative Committee for Space Data Systems (CCSDS) VCM specification [1] refers to the automation of the selection of VCM mode as Adaptive Coded Modulation (ACM). The protocol methodology we investigate here explores the use of LTP retransmission rates to infer the quality of the RF channel, and then, based on LTP statistics, automatically command modes changes to the VCM-capable radio. The VCM combinations of coding and modulation formats standardized by the CCSDS are described below in Table 1.

Table 1. VCM Modes to be Supported by the STRS.

VCM Mode	Modulation	Code	Code Rate (Note 1)	Input Length (short) *K* bits (Note 2)	Input Length (long) *K* bits (Note 2)
0	Reserved				
1	BPSK	Turbo	1/6	1784	8920
2	BPSK	Turbo	1/4	1784	8920
3	BPSK	Turbo	1/3	1784	8920
4	BPSK	AR4JA LDPC	1/2	1024	16384
5	BPSK	AR4JA LDPC	2/3	1024	16384
6	BPSK	AR4JA LDPC	4/5	1024	16384
7	BPSK	C2	223/255	7136	7136
8	QPSK	AR4JA LDPC	1/2	1024	16384
9	QPSK	AR4JA LDPC	2/3	1024	16384
10	QPSK	AR4JA LDPC	4/5	1024	16384
11	QPSK	C2	223/255	7136	7136
12	8-PSK	AR4JA LDPC	1/2	1024	16384
13	8-PSK	AR4JA LDPC	2/3	1024	16384
14	8-PSK	AR4JA LDPC	4/5	1024	16384
15	8-PSK	C2	223/255	7136	7136
16	16-APSK	AR4JA LDPC	1/2	1024	16384
17	16-APSK	AR4JA LDPC	2/3	1024	16384
18	16-APSK	AR4JA LDPC	4/5	1024	16384
19	16-APSK	C2	223/255	7136	7136
20	32-APSK	AR4JA LDPC	1/2	1024	16384
21	32-APSK	AR4JA LDPC	2/3	1024	16384
22	32-APSK	AR4JA LDPC	4/5	1024	16384
23	32-APSK	C2	223/255	7136	7136
24	64-APSK	AR4JA LDPC	1/2	1024	16384
25	64-APSK	AR4JA LDPC	2/3	1024	16384
26	64-APSK	AR4JA LDPC	4/5	1024	16384
27	64-APSK	C2	223/255	7136	7136
28	Reserved				
29	Reserved				
30	Reserved				
31	Reserved				

NOTES:

1 The turbo codes have a slightly lower code rate than listed, because of the termination bits used at the end of the code block.

2. Information block length K is discussed in Section 3.3.1.2 and Section 3.3.2.2 of CCSDS VCM Protocol.

The table is taken from [1], and details of the LDPC codes may be found in [2].

A. Outline

In Section II, we describe the adaptation methods considered and subsequently chosen to automatically change VCM modes as a function of channel quality.

In Section III, we describe the use of LTP, the software needed to extract detailed LTP segment retransmission rates to monitor the effect of channel errors on LTP bandwidth use, and the simulation experiment setup to test the concept. In the final portion of Section III, we detail the overall approach to using LTP to affect Automatic Coded Modulation change.

Following that, in Section IV we describe the test bench and channel simulation designed to conduct these tests, and in Section V, we describe the results that prove the concept of using LTP to affect ACM.

Finally, concluding remarks are made in Section VI. There, we detail next steps to eventually implement a full ACM process in follow-on radio projects.

II. Adaptation Mechanisms for a VCM Radio

Adaptive Coded Modulation (ACM) is accomplished by using some means of automatically inferring the quality of a link channel, and then acting on that information to change coding and/or modulation modes to maximize error-free data throughput, also known as “goodput.”

This fundamentally requires some sort of feedback from the receiver, which takes at least one round-trip light time. At the physical layer, a receiver, after demodulation and decoding, can infer an estimate of signal-to-noise ratio (SNR) and send feedback to the transmitter. This provides a snapshot of the channel conditions one round-trip light time in the past. In an extreme case where channel condition worsens dramatically and forces the receiver out of lock, the receiver may not be able to provide a reliable SNR estimate.

Specialized radio hardware that can measure SNR exists; however, it is desired to provide a mechanism that is not dependent on any particular radio hardware. This allows an ACM implementation in software outside of a radio with VCM capability to control the coding/modulation mode used by the radio.

Because of this requirement, the use of LTP was explored in this project, as it is expected that many future missions will be using DTN with the LTP protocol on links where there are no other viable reliability protocols, such as deep space links.

An LTP-based mechanism, which uses accumulated data loss statistics that are readily available on the transmitter, provides a natural synergy between Variable Coded Modulation Radio and DTN protocols for reliable and efficient communications.

To see why we arrived at the conclusion that LTP would be a viable solution, we discuss the merits of control at each layer of the OSI stack from the bottom up.

A. Monitoring Channel Quality at Physical Layer

Inferring channel quality by observing signal-to-noise ratio requires receiver hardware at the physical (PHY) layer. Then, there must be a protocol to signal the remote transmitter when quality degrades, causing it to change coding and/or modulation mode. The existing UHF proximity communications radios on the Mars orbiters/landers perform this sort of signaling, automatically adjusting data rates to maximize throughput. The drawback to this approach is, of course, the need for specialized radio hardware and new protocol for physical layer signaling.

B. Monitoring Channel Quality at Coding/Synchronization/Multiple Access Layers

In the coding layer, this would require decoding firmware to make complex decisions that may be difficult to implement in field programmable gate arrays (FPGA). Likewise, no protocol standard exists for relaying channel condition information back to the transmitter, and this would require a similarly-equipped radio on the other end. Monitoring at the sublayer above the coding layer would be possible but would require (non-standard) modifications to the TM/TC/AOS/Prox-1 link layer protocols and additional decision-making intelligence in whichever devices were operating the link layer protocols, which are often built into radio firmware.

C. Monitoring Channel Quality at Transport Layer

Monitoring channel performance at the transport layer is practical, and since the transport mechanisms are normally distant enough from lower layers to be independent of radio hardware, a system using transport layer monitoring could be independent of the underlying radio hardware or link protocols in use. This is where the Disruption-Tolerant Networking (DTN) Licklider Transmission Protocol (LTP) could be used to infer channel quality based on retransmission rates.

III. Use of LTP for ACM Control

Using the transport layer operation to discern an optimum transmission coding/ modulation mode requires new software to use feedback from the LTP layer to infer channel quality based on the amount of re-transmission requests LTP is servicing. In addition, the LTP protocol stack is modified to provide the mode decision software with the necessary automatic retransmission request (ARQ) information. Then, a table-look-up or other algorithm can be used to select mode changes based on the ARQ statistics and command the radio accordingly.

This ACM software could be hosted in a software-defined radio or in the C&DH external to the radio. In addition, the ACM software could be hosted on just one of the transceivers as long as both ends of the link were using LTP. This might be the case where a transceiver on a DTN node recognized the VCM modes in use, but had limited ability to implement ACM control software for the return link. This could at least improve goodput in one direction.

A. Summary of DTN and LTP Operation

A simplified explanation of DTN and LTP is as follows: The DTN stack starts, at the top, with an application that creates a file or stream of data to be sent to another computer over what could be multiple hops over different links (including terrestrial networks and radio or optical wireless links), and hands that information, along with addressing and priority instructions to the DTN Bundle Protocol (BP). The application has no expectation of immediate feedback, as it must be aware of the limitations of networks with uncertain links or connections that are periodic (such as forced by orbital mechanics), and may request that the next node in the path take custody (and responsibility) of the data if the operations concept so demands.

BP is somewhat analogous to the IP layer in the terrestrial internet, as it creates a "bundle" by adding header information (defining source/destination/time-to-live/priority, and other parameters) to the data presented by the application. Like an IP packet, the BP bundle then may be routed, prioritized, and stored. Additional headers may be added to the bundle to specify encryption and integrity information for security. The size of the bundle is user selectable, with bundle size dependent on latency requirements, channel efficiency, or any number of network systems engineering considerations.

The next layer below BP would be either a reliable or an unreliable link layer. If end-to-end reliability is desired, the network could either use hop-by-hop custody transfer (custody is not accepted by the next node until the entire bundle of data is correctly received), or use a reliable transport or link layer, such as TCP/IP in terrestrial links, or CCSDS Prox-1 in reliable bitstream mode for orbital-to-surface communications. However, since a reliable link service for deep space was not available, the LTP protocol was developed and standardized. LTP allows for both reliable and unreliable data to be transmitted, but only the reliable mode of LTP was used for this ACM study.

Here is a somewhat simplified description of how LTP accomplishes reliable transmission (some details are left out that may be found in the LTP Blue Book [3]):

- BP provides DTN bundles to the LTP software, and the LTP software creates an LTP block which is the unit that LTP tracks as an LTP "session." The LTP block may consist of one or more DTN bundles (user configurable). The block is in turn fragmented into segments for transmission.

- LTP establishes a "session" for each block, and stores that block until it is assured that all the segments of that block are received by the destination node. The block is broken into segments that are sized to efficiently integrate with the underlying telemetry/coding frame sizes in use.

- At the end of the transmission of the block, the last segment transmitted is flagged as a "checkpoint," which signals the receiver to send the transmitter a report of how much of the block it received. If the report says all segments were correctly received, then that session is closed, and the memory of that block may be released at the LTP sender. If, however, the receiver reports that some segments were missing, LTP then will retransmit the missing segments and wait for another checkpoint report that the block was finally fully received. This scheme improves return link efficiency, as each segment is not individually acknowledged, only a short summary report detailing "missing pages" or segments is required in the return link.

- The size of the BP bundles, the size of LTP blocks, the size of LTP segments, and the number of concurrent active LTP sessions to allocate memory for are all user-configurable, which, although making DTN network engineering a complex proposition, allows for maximization of throughput and link efficiency.

- The NASA standard DTN suite for space missions is called the Interplanetary Overlay Network (ION), and is publicly available on SourceForge [6]. ION version 3.6.1 and above have the LTP software "instrumented" to report segment retransmission statistics to external software so that user-provided software can then use the statistics of LTP retransmissions to infer the quality of the channel as experienced over the transmission time of an LTP block.

B. ACM Control Components

The design of an ACM control method depends on both the characteristics of LTP retransmission operations under different loss conditions and the dynamics of the link channel SNR variations. The experiments described below serve to explore the characteristics of LTP retransmissions with different VCM modes and different SNR conditions and explore the possibility of lossless data throughput using a combination of coding and LTP.

The specifics of the algorithm to adaptively select coding/modulation modes based on LTP retransmission data and subsequent research in this area is discussed in the summary section.

C. Study Hypotheses and Expected Results

This study started with the hypothesis that LTP retransmission characteristics can be used to perform ACM and provide greater error-free throughput (goodput) than picking a single mode for an entire contact period.

To explain how this would be used, we present some hypothetical cases that might be observed over the course of a long Deep Space Network (DSN) pass where the signal-to-noise ratio (E_s/N_0) varies from +4 dB to –4 dB.

Figure 1 shows a notional plot of goodput in a link that uses both forward error correction coding (i.e., LDPC rate 4/5) as well as LTP. (The curves are smoothed and exaggerated for the purpose of the illustration.)

On the plot are both the goodput (left vertical axis) and the LTP segment retransmission rate (right vertical axis). When SNR is high, corrupted data are recovered solely by the coding, at the expense of coding overhead (20%) as shown. At a certain point, coding alone can no longer recover the data, and the LTP protocol starts retransmitting the missing data, which while improving the reliability, it uses bandwidth for the retransmissions, thus reducing the net goodput.

Next, we compare this plot to one of a more robust (but less efficient) code shown in Figure 2.

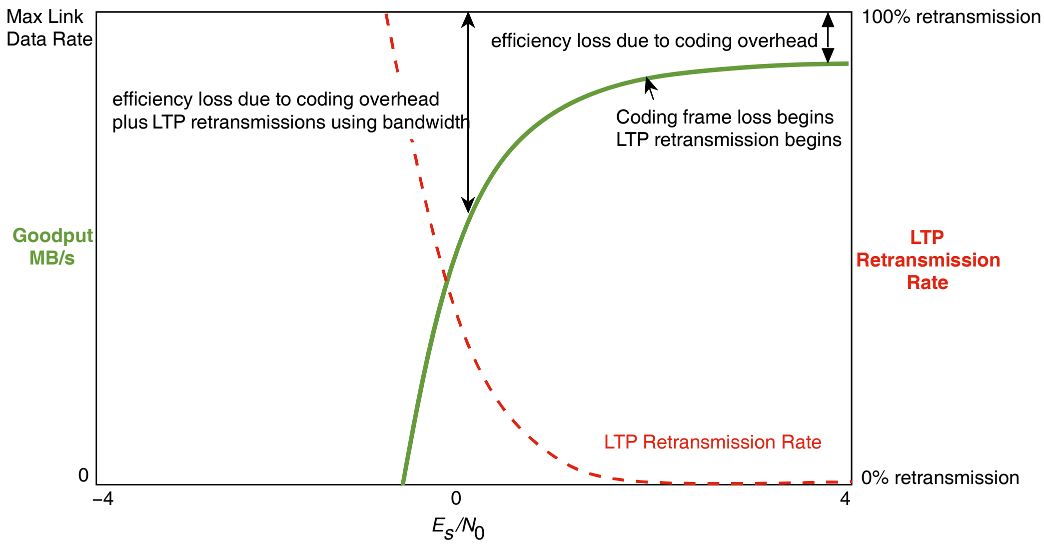

Figure 1. Notional reliable throughput with LTP using rate 4/5 LDPC, Mode 6.

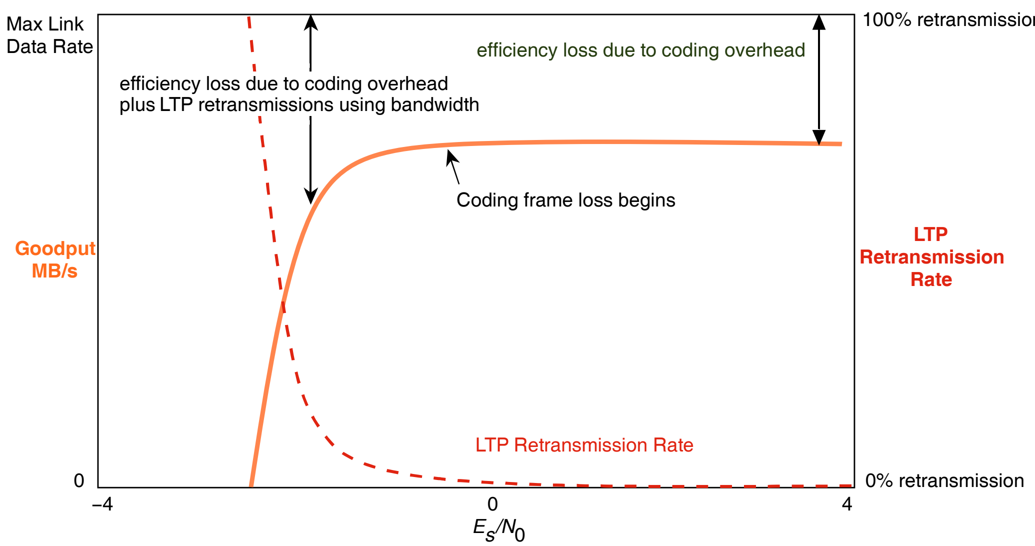

Figure 2. Notional reliable throughput with LTP using rate 2/3 LDPC, Mode 5.

Here the coding is less efficient (33% coding overhead), but it is able to recover bit errors down to a much lower SNR than the code in Figure 1. Likewise, LTP doesn't start retransmitting and recovering frames lost due to uncorrectable errors until conditions are much worse than in the case of Figure 1.

Figure 3 puts those two plots side-by-side to illustrate the mode switch strategy.

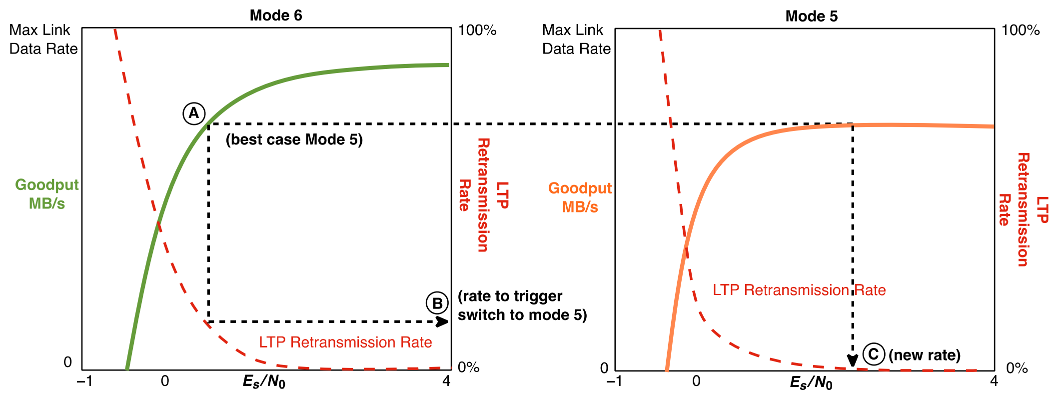

Figure 3. Using LTP Retransmission Rate to Decide to Switch to Mode 5.

If we start off at high SNR, using the most efficient rate 4/5 code on the left, and follow the goodput curve to the left as the SNR gets worse, we see that at point A we have overall goodput with coding plus retransmission equal to that of case on the right where the less efficient rate 2/3 code is performing without LTP retransmissions. The LTP retransmission rate indicated by the letter B would then signal the point at which a change to Mode 5 would be advantageous.

It can be seen that if the SNR degraded from the positive numbers at point A and we just stayed in Mode 4, the goodput would continue to drop. At the point where not even LTP could recover data by retransmission, Mode 5 is still near the maximum.

Figure 4 shows the aggregate of three notional VCM modes using LTP to trigger mode changes.

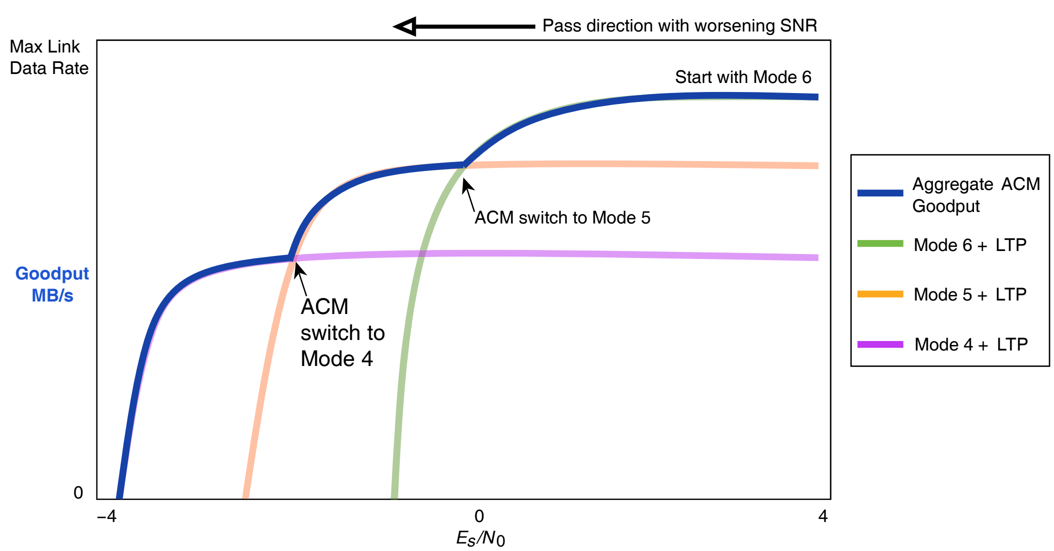

Figure 4. Reliable throughput with LTP and ACM combining modes.

Assuming the E_s/N_0 curves in Figure 4 represented the change of SNR over half of a DSN pass, the area under the various curves is proportional to the total reliable data return, and it is easy to see that automatically switching between modes based on using LTP to decide when to switch is a potentially advantageous strategy.

With the full complement of available VCM modes, the aggregate ACM goodput curve will be much smoother as there will be many more mode change possibilities available.

IV. ACM Tests with VCM Modes and LTP

In order to test this methodology with real implementations of DTN ION with LTP, CCSDS AOS link protocol and six of the available VCM modes, a simulation of a transceiver pair with bit-errors applied to the simulated link was developed, and a number of one-hour runs were made to collect goodput statistics under various channel conditions and the corresponding LTP segment retransmission rates. Figure 5 details the test setup.

This test configuration uses flight-qualified ION BP/LTP software, AOS software that has been validated in tests with commercial implementations, and software implementations of the coding that was implemented in the STRS FPGA and tested against reference decoder software. For this report, six VCM modes were tested, and the AWGN bit error rate link channel emulation was provided by the JPL Coding group. In practice, for longer term use, this testbed can also be configured to use a commercial WAN emulator that can inject bit errors into the symbol stream. In this hardware implementation, the symbol stream is sent over UDP/Ethernet, and the WAN emulator injects errors only in the symbol stream, so no losses due to UDP or ethernet frame header corruption are experienced.

A. Test Setup

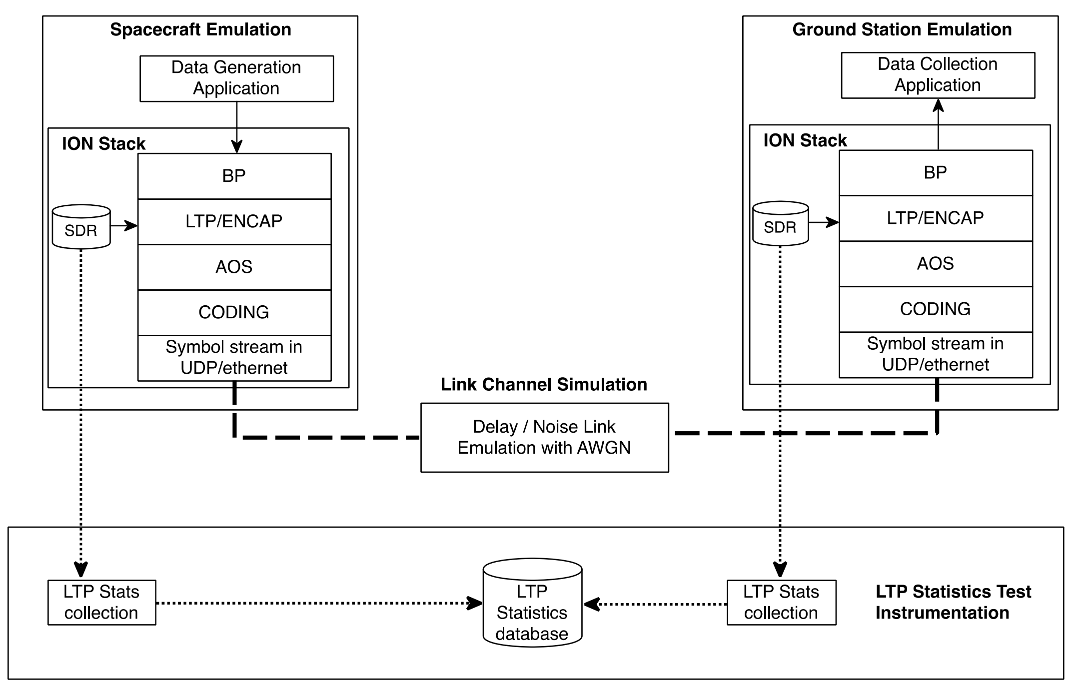

Figure 5. ACM Testbed Configuration.

The LTP Statistics Test Instrumentation detailed above took LTP segment retransmission data from the Simple Data Recorder (ION-SDR) portion of the ION DTN implementation and collected the data in a third computer.

The following section details the test parameters and the test results.

B. Protocol Set-up Details: BP, LTP, and AOS Configurations and Impact on Simulation Results

The protocol stack and sizes of the various elements in the stack are shown in Figure 6.

Test Data		187 bytes/data unit
DTN BP		228 bytes/bundle
LTP BLOCK		228 bytes/LTP block
LTP SEGMENT	LTP SEGMENT	110 bytes/LTP segment
ENCAP PACKET	ENCAP PACKET	114 bytes/encap packet
AOS FRAME	AOS FRAME	128 bytes/AOS frame
CODING	CODING	160–256 bytes/codeblock
SYM/UDP	SYM/UDP	(frame rate varied to normalize throughput for all codes)

Figure 6. Test protocol stack configuration.

Software was prepared to implement the BP/LTP/AOS/Coding stack for a selected subset of possible VCM modes as described in [1]. The details of the coding input/output frame sizes and the relevant parameters are shown in Table 2.

Table 2. Modes and frame sizes implemented for this study.

Bundle Size (bytes)	LTP Block Size (bytes)	LTP Segment Size (bytes)	AOS Frame Size (bits)	VCM Mode	Modulation	Modulation Rate, R_m	Code	Code Rate, R_c	Short Frame Input Size (bits)	Short Frame Output Size (bits)	Modulated Frame Size (symbols)	LTP Starts Below E_b/N_0 (dB)	Frame Rate (frames/s)	Symbol Rate (ksps)
288	228	110	1024	Mode 4	BPSK	1	AR4JA LDPC	0.5	1024	2048	2048	1.8	53.71	110
288	228	110	1024	Mode 5	BPSK	1	AR4JA LDPC	0.67	1024	1536	1536	2.6	71.61	110
288	228	110	1024	Mode 6	BPSK	1	AR4JA LDPC	0.8	1024	1280	1280	3.69	83.94	110
288	228	110	1024	Mode 8	QPSK	2	AR4JA LDPC	0.5	1024	2048	1024	1.8	107.42	110
288	228	110	1024	Mode 9	QPSK	2	AR4JA LDPC	0.67	1024	1536	768	2.6	143.23	110
288	228	110	1024	Mode 10	QPSK	2	AR4JA LDPC	0.8	1024	1280	640	3.69	171.88	110

Since we were comparing goodput between modes, and since we did not actually modulate the symbol streams, to account for the differences in modulation, the rate at which the frames were sent via UDP to the receiver simulation was varied to maintain an effective 110K symbols/sec rate, as detailed in Table 2.

Due to time and budgetary limitations, test runs were made and analyzed using the first six VCM modes only. Each data collection run ran long enough to collect data on at least 20,000 frames transmitted. The column labeled "Required E_b/N_0 (dB)" indicates the minimum E_b/N_0 expected before LDPC retransmissions are necessary, and was used to bound the AWGN noise injection simulation which was inserted between the transmitter and the receiver.

V. Results and Analysis

In this section, we summarize the tests conducted and data collected. Multiple runs were made for statistical significance, and the summary below is the final aggregation of the data.

A. Tests Conducted

Table 3 below details the range of E_b/N_0 channel quality simulated, the total number of runs per mode, the duration of each test and the average number of frames transmitted at each E_b/N_0 value.

The channel emulation used E_b/N_0 for the input values for determining the amount of AWGN added, whereas the final plots of goodput versus channel quality are expressed in E_s/N_0, with the relationship being:

$$\frac{E_s}{N_0} = \frac{E_b}{N_0} + 10\log_{10}\left(Rc * Rm\right)$$

where Rc and Rm are as listed in Table 2.

Table 3. Channel noise ranges, test durations, and frames tested/test run.

VCM Mode	E_b/N_0 Range Tested, dB	Resulting E_b/N_0 Range	E_b/N_0 Change, Step Size, dB	Test Duration, min	Number of Frames in Test (avg)	Test Steps
4	1.0–6.0	–1.0–3.0	0.1	10	32K	51
5	–.02–4.8	–1.8–3.0	0.1	10	43K	52
6	2.0–4.2	1.03–3.2	0.1	10	50K	23
8	–2.0–3.0	–2.0–3.0	0.1	10	64K	51
9	1.0–3.0	2.2–4.2	0.1	10	86K	21
10	2.0–4.2	4.0–6.2	0.1	10	103K	23

To describe a typical test, the table shows that for VCM Mode 4, the AWGN channel simulator was set at an initial value of 1.0 dB E_b/N_0, then data was flowed for 10 minutes, resulting in about 32,000 frames sent over the channel. Then the noise was raised by 0.1 dB, and the test was repeated. This process was repeated a total of 50 times, yielding data on code and LTP performance over the range of 1.0 to 6.0 E_b/N_0, with 0.1 dB granularity.

At 10 minutes per test, and between 21 and 51 test runs per mode, each mode was tested for between 210 and 510 minutes. Each of these test runs were repeated multiple times so that test repeatability and statistical validity could be assured.

B. Emulation Results

Figure 7 shows the final summary plot of goodput versus channel quality expressed in E_s/N_0. It is clear that the proper selection of mode can greatly improve overall goodput, and data shows that transitions between modes occur at E_s/N_0 levels where the code has begun to fail and has to be augmented by LTP retransmissions.

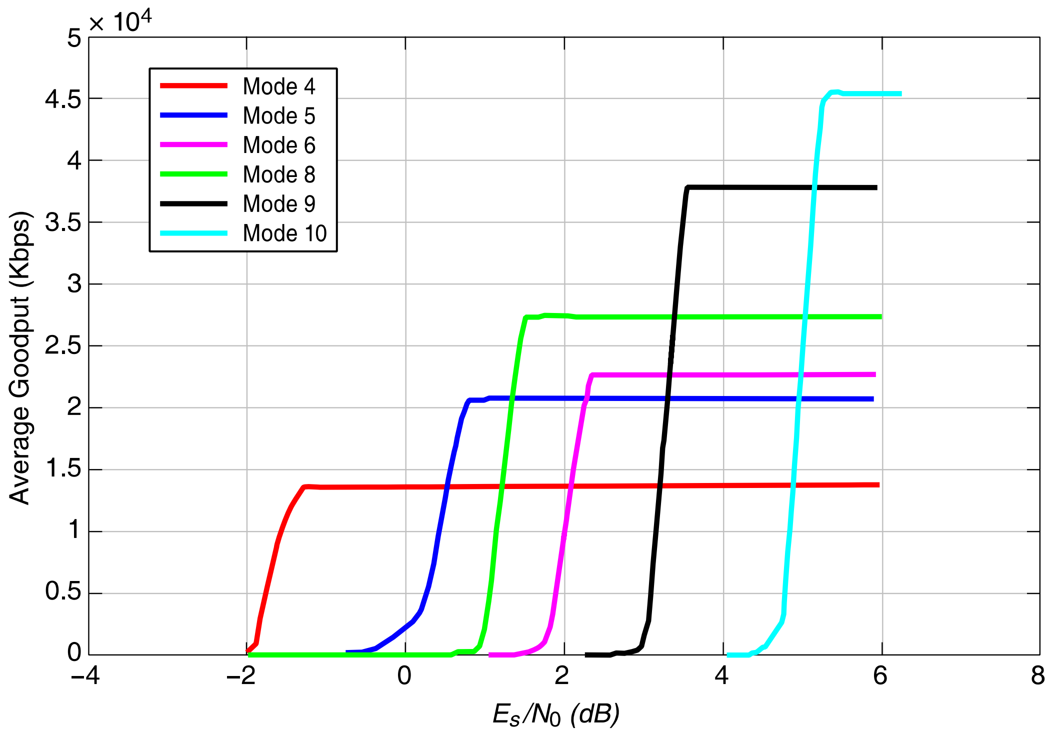

Figure 7. Simulation results: Goodput as a function of signal strength, with LTP and coding.

With LTP, it would be possible to achieve an automatic VCM mode transition as shown in Figure 8.

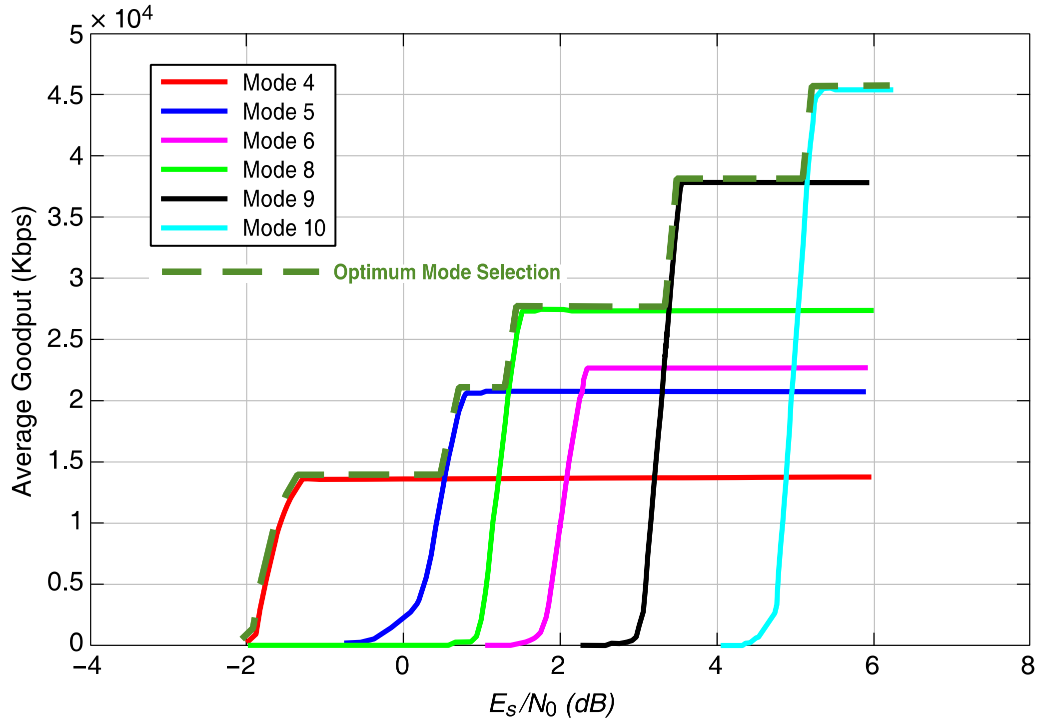

Figure 8. Optimum Mode Selection Strategy.

If Figure 8 represented, say, half of an overhead pass of a satellite over a ground receiver, it is seen that the total data volume received on the ground (proportional to the area under the Goodput curve) is greatly increased by using all of the available modes. In a real implementation, many more modes would be implemented, and the step-wise character of the green optimum mode line would be smoothed out.

VI. Conclusions

A. VCM Augmented by LTP Works Well

Our experiments have provided statistically significant data showing that LTP interoperating with any VCM mode increases the range over which useful data may be obtained over an increasingly noisy channel. When the coding starts to fail, LTP begins automatic retransmission, thus increasing the overall system goodput.

B. LTP Segment Retransmission May Be Used to Make Mode Change Decisions

LTP can be used to good effect to devise a table-driven on-board strategy for automatically changing VCN modes, and with multiple modes available on a given radio, strategies as shown in Figure 8 can greatly improve the goodput over a noisy channel.

C. Tuning and Network Engineering Are Critical Components

We saw that the performance of LTP and overall practicality of this approach were dependent upon many factors that need further study. These include “impedance matching” between various layers in the stack to avoid wasteful fill frames or fill bits, and this has to propagate up to the bundle layer so that bundle sizes used to convey the data down through the stack are optimized.

LTP allows the aggregation of bundles into a single LTP block, but as the retransmissions only begin after a checkpoint request at the end of a transmitted block, the channel noise dynamics and the round-trip light times involved need to be considered carefully to decide on the best strategy for using LTP to accomplish ACM. On one hand, large bundles and large LTP blocks maximize efficiency by minimizing the ratio between data and the various header overheads, but checkpoints (and changes in LTP segment retransmission statistics) may be too far apart in time to compensate for a channel that has rapidly-changing SNR.

Conversely, small bundles may be used, resulting in small LTP blocks and more frequent checkpoints and more rapidly updated retransmission statistics to improve the situation when the channel condition changes quickly, but at the expense of extra header overhead, as well as additional computational overhead, since each bundle needs to go through the contact graph routing computations.

The LTP specification [3] allows for interim checkpoints to be requested during a block transmission, but this feature is not currently implemented in ION. The necessity of interim checkpoints is still to be determined; with short distance links with very dynamic channel conditions, the latencies may not be too bad, and interim checkpoints may not be worth the extra software and header overhead – for deep space links with long one-way light times, the channel conditions tend to change slowly over the course of a full DSN pass, so again interim checkpoints may not be necessary. Further end-to-end comm system engineering studies are clearly in order.

D. Feedback Latency Will Drive ACM Control Algorithm Design

The utility of using LTP to infer channel quality and use of retransmission statistics were demonstrated in the experiments we have described. The next task will be to decide on how to design an automated algorithm to control the modes. This may be as simple as a lookup table where predetermined mode changes are specified, or something more complex based on the dynamics of the channel SNR changes.

Since the LTP retransmissions do not occur until a data report segment is received by the transmitting LTP software, any estimates of channel conditions will lag the instantaneous link conditions by the block transmission time to the destination node, plus the one-way light time required to receive the block checkpoint report from the destination node.

Due to the added latency of the feedback via LTP, this method will perform best in scenarios where the SNR is slowly changing, with the algorithms used to decide on mode changes using cumulative statistics to provide unbiased estimates of channel conditions. Since LTP is intended for use over deep space links where SNR changes occur over the course of many hours during a DSN pass, we expect that this approach will be satisfactory for LTP.

For proximity link conditions where passes are short, one-way light times are measured in milliseconds vice minutes, and lander antenna patterns are a factor for SNR dynamics, there are other reliable link protocols that will have to be evaluated for use in ACM algorithms. These include Proximity-1, used on Mars lander missions, and DTN Datagram Retransmission (DGR), which is akin to an LTP protocol designed with less than one second OWLT expected. ACM algorithms for proximity links are a subject for future research.

E. Next Steps

The VCM/ACM project, which was a part of a larger Cognitive Networking initiative, set out to first demonstrate the new VCM Modes in [1] on the ISS. The initial flight test accomplished a verification of the first eight VCM modulations in the STRS FPGA, with coding done in the STRS flight software. The report of that initial flight test was presented in IPNPR 42-212 [4]. In parallel with the FPGA firmware and flight software development, a ground receiver based on a National Instruments USRP transceiver and GNU radio software was developed. This system could switch receiver VCM modes without going through BPSK to signal the transition to the next modulation/code mode, as is the practice in commercial DVB-S2 used in Earth-orbiting systems. The ground receiver development was reported in IPNPR 42-213 [5].

The next steps taken were to implement the APSK VCM modulation modes in the FPGA, add the coding to the FPGA, and then repeat the flight test on the ISS with all VCM modes in Table 1 implemented in firmware. Unfortunately, the project was cancelled at the end of Year 2 before the full flight system could be flown.

The development of the algorithm to use LTP to switch modes and to flight test the system was planned for the third year of the project, while Year 4 planned to fine tune the algorithms, and conduct further lab testing of the system to prepare for infusion into future spacecraft radio transceivers.

After cancellation, sufficient funding for closeout was granted that allowed us to complete the ACM via LTP simulations presented in this IPNPR, and to complete the FPGA work needed to implement all the VCM modes in [1] and test the FPGA performance in a laboratory environment. We were also able to thoroughly document the work to date and archive all of the software so that, should future budget allow, we can continue this work to complete a full lab prototype system and potentially some sort of future flight test on a CubeSat.

VII. Summary of VCM/ACM Project

Overall, we have

- Implemented all new VCM coding/modulation modes into an FPGA.

- Flown a subset of them on the ISS.

- Developed a ground radio receiver system in GNU radio that can switch between modes on the fly without having to go to Mode 0 BPSK between modes as does DVB-S2, which will enable higher throughput.

- Published three IPN Progress Reports on the work (including this one). Shown the way toward implementation of VCM on future flight radios such as the UST.

- Demonstrated how DTN LTP retransmission data (using existing DTN ION software) can be used as a radio-independent means of automatically switching modes based on observation of LTP retransmission rates.

Plans for future proposed work to continue this project were presented at the December 5, 2018, SCaN Technology Annual Review, and the fact that this technology integrates DTN with the new deep space CCSDS VCM modes was discussed as grounds for future work on this concept and eventual continuation of the VCM/ACM Cognitive Networking project.

References

[1] Variable Coded Modulation Protocol, Consultative Committee for Space Data Systems (CCSDS), Draft Recommended Standard (Red Book), CCSDS 431.1-R-1, December 2018.

[2] Low Density Parity Check Codes for Use in Near-Earth and Deep Space, Silver Book, Consultative Committee for Space Data Systems (CCSDS), Experimental Specification, CCSDS 131.1-O-2, September 2007. https://public.ccsds.org/Pubs/131x1o2e2s.pdf .

[3] Licklider Transmission Protocol (LTP) for CCSDS, Blue Book, Consultative Committee for Space Data Systems (CCSDS), Recommendation for Space Data System Standards, CCSDS 734.1-B-1, May 2015. https://public.ccsds.org/Pubs/734x1b1.pdf .

[4] H. Xie, S. Dolinar, M. Chase, S. Clancy, M. Kilzer, L. Torgerson, and A. Tkacenko, Flight Test and Validation of Variable Coded Modulation Using SCaN Testbed, *Interplanetary Network Progress Report,* vol. 42-212, February 2018.

[5] A. Tkacenko, M. Kilzer, M. Chase, S. Clancy, H. Xie, L. Torgerson, and G. Miles, VCM Cognitive Radio Higher-Order Constellation Receiver Operation, *Interplanetary Network Progress Report,* vol. 42-213, May 2018.

[6] https://sourceforge.net/projects/ion-dtn/