

Henrik Axelsson
KenCast, Inc.
Email: HAxelsson@KenCast.com

Resilient Data Delivery Techniques for SmallSat and Hybrid Space–Ground Architectures

Abstract

SmallSat missions frequently rely on space–ground communication paths that are asymmetric, multicast-oriented, intermittently available, or intentionally unidirectional. Many SmallSat and LEO architectures operate under tighter power, antenna size, and link-budget constraints than traditional GEO systems, and may rely on shorter contact windows, which can contribute to lower signal-to-noise ratios, burst packet loss, and limited or unavailable return channels. These conditions are common in missions supporting synthetic aperture radar, other high-volume sensing payloads, on-orbit processing, artificial intelligence workflows, alternative positioning, navigation and timing concepts, and intelligence, surveillance and reconnaissance data dissemination.

These constraints become more pronounced in hybrid GEO–LEO architectures, mobile or distributed ground stations and government-controlled satellite networks, where heterogeneous links, frequent handovers and fragmented reception limit the usefulness of feedback. In broadcast and multicast delivery, approaches that depend on feedback or complete reception within a single downlink contact lead to unpredictable delays, incomplete deliveries and strong operational dependence between space and ground segments.

This paper examines application-layer data delivery techniques for broadcast and multicast environments, including packetization strategies, application-layer forward error correction, time-distributed redundancy and rateless erasure coding. These methods allow receivers to reconstruct data across multiple, non-contiguous contact windows rather than relying on retransmission or receiver feedback. The paper contrasts link-layer and application-layer error correction, examines the limits of retransmission-based models, and describes how accumulation-based delivery better aligns with intermittent, one-way and hybrid space–ground networks.

I. Introduction

SmallSat systems are increasingly deployed in roles that require the delivery of large volumes of data under communication conditions that are inherently constrained and variable. Applications such as synthetic aperture radar (SAR), remote sensing, and ISR (intelligence, surveillance, and reconnaissance) generate data sets that are both large and operationally significant. These data sets are often time-sensitive and, in many cases, difficult or impractical to retransmit once an opportunity has passed.

Unlike traditional geostationary (GEO) systems, which are commonly associated with persistent visibility and relatively stable link conditions, many SmallSat architectures—particularly in LEO constellations and hybrid space–ground networks—operate under intermittent or time-constrained connectivity. Ground stations may only have visibility into a satellite during short contact windows, and even within those intervals, link conditions can vary significantly due to geometry, mobility, atmospheric effects, and changing signal conditions.

These variations are commonly described using SNR (Signal-to-Noise Ratio), which reflects the strength of the signal relative to background noise. Noise, in this context, includes atmospheric effects, interference, and system-level distortions that reduce the receiver’s ability to correctly interpret the transmitted signal.

As SNR degrades, packet loss becomes more likely. This degradation is not binary. A communication link does not simply transition from reliable to unusable; instead, it operates along a continuum. During a single pass, some portions of the transmission may be received cleanly, while others may experience elevated loss. The receiver therefore observes a mixture of correctly received packets and missing data.

The practical consequence is that data is rarely received as a complete and contiguous sequence. Instead, it is received in fragments. A file transmitted during a satellite pass may be partially received, with additional portions arriving during subsequent passes or through alternate delivery paths. This fragmented

reception is not an edge case—it is the expected operating condition.

A delivery system designed for these environments must therefore operate under a different set of assumptions. It must assume that reception will be incomplete, that loss will be uneven and often bursty, and that reconstruction must occur across time rather than within a single transmission interval.

II. Fragmented Reception in Hybrid Architectures

Fragmented reception becomes more pronounced in hybrid architectures that combine GEO, low Earth orbit (LEO), and terrestrial networks. These systems introduce variability not only in signal quality, but also in latency, path availability, and receiver conditions.

Data may traverse multiple delivery paths, each with distinct characteristics. A GEO link may provide broad coverage but higher latency. A LEO link may offer lower latency but shorter contact windows and more frequent handovers. Terrestrial paths may provide high throughput but may not be universally available. In practice, these paths are often used in combination, with data moving dynamically between them depending on availability and conditions.

Receivers in such systems are rarely homogeneous. Some may be fixed ground stations with stable connectivity and favorable reception conditions. Others may be mobile, intermittently connected, or operating under constrained environments. As a result, different receivers often capture different subsets of the same transmission.

Even a single receiver may accumulate data over time. One contact window may deliver a portion of a file, while subsequent windows deliver additional portions. These fragments may be separated by periods of no connectivity and may arrive out of sequence.

This leads to a fundamental asymmetry between transmission and reception. From the perspective of the transmitter, the data may have been sent in full. From the perspective of any individual receiver, however, the received data set may be incomplete, out of order, and assembled over multiple intervals.

In multicast and broadcast systems, this asymmetry becomes more significant. A single transmission is intended to serve many receivers, yet each receiver experiences different loss patterns. One receiver may

miss packets due to interference, another due to mobility, and another due to timing. A delivery model that assumes uniform reception across receivers is therefore not aligned with the operational reality.

The delivery problem is not simply one of transmission reliability. It is one of reconstruction from partial and uneven reception over time.

III. Conventional Delivery Models and Their Limitations

Traditional delivery models address packet loss through retransmission, redundancy, or repetition. Each approach is effective under certain conditions, but each relies on assumptions that become difficult to satisfy in fragmented multicast environments with asymmetric or constrained return paths.

Retransmission-based approaches rely on feedback from the receiver. ARQ (Automatic Repeat reQuest) allows a receiver to request specific packets that were not successfully received. Transport-layer protocols such as RIST (Reliable Internet Stream Transport) and QUIC (Quick UDP Internet Connections) extend this approach by incorporating buffering, selective retransmission, congestion control, and flow management.

These mechanisms are effective in bidirectional networks where feedback is available and latency constraints are manageable. However, in SmallSat environments, the return path may be unavailable, intermittent, or constrained in capacity. Even when a return path exists, the time required to request and receive missing packets may exceed the duration of the contact window. A receiver that detects loss late in a pass may not have sufficient time to request recovery before the link is lost.

Retransmission also introduces structural challenges in multicast systems. Each receiver may experience a different loss pattern. Supporting retransmission for multiple receivers requires maintaining receiver-specific state and potentially transmitting different packets to different receivers. This reduces the efficiency advantages of broadcast delivery and introduces complexity in scheduling and resource allocation.

Forward error correction (FEC) provides an alternative by adding redundancy to the transmitted data. In most implementations, FEC operates over

fixed groups of packets. Recovery is possible if enough packets from a group are received.

This approach is effective for random or low-level loss but is less effective under burst loss conditions. If multiple packets within a group are lost in a short interval, the available redundancy may be insufficient to recover them. In addition, recovery is confined to the packets within that group. Packets received later do not contribute to recovery, even if they contain useful information.

Repetition, or carousel delivery, attempts to improve reliability by transmitting the same data multiple times. While this increases the probability that receivers will eventually obtain missing packets, it is inefficient. Receivers may repeatedly receive data they already have while still missing other portions. The approach also scales poorly when receivers have different loss patterns.

These approaches share a common limitation: they depend on the eventual reception of specific packets. In fragmented environments, this dependency becomes increasingly difficult to satisfy.

IV. Link-Layer and Application-Layer Forward Error Correction

Forward error correction is applied at multiple layers of a communication system, and its effectiveness depends on where it is implemented.

Link-layer FEC operates at or near the physical transmission layer. It is typically implemented within the modem, radio, chipset, or transmission waveform itself, and is commonly integrated into standards such as DVB-S2/S2X, CCSDS channel coding, 5G NR, and ATSC 3.0. As a result, link-layer FEC is usually tightly coupled to modulation and waveform behavior and operates before higher-layer IP or application processing occurs.

Its purpose is to correct errors introduced during transmission over a noisy channel. By adding redundancy at the bit or symbol level, link-layer FEC enables receivers to recover from errors caused by noise, fading, or interference.

This form of FEC is essential for maintaining link reliability. When the signal is present but degraded, link-layer FEC improves the probability that packets are received correctly. However, it operates within the temporal bounds of a transmission interval. It

assumes that packets arrive, even if imperfectly, and focuses on correcting errors within those packets.

Link-layer FEC does not address the absence of packets. If packets are not received at all—due to coverage gaps, mobility, or the end of a contact window—it cannot recover them.

Application-layer FEC operates above the transmission layer and protects the data itself rather than the transmission waveform or individual packets. Unlike link-layer approaches, it is generally software-defined and applies redundancy at the object level, allowing packets received across different sessions, paths, or transmission intervals to contribute to the same reconstruction process. Data that would otherwise be unusable because it is incomplete can instead be retained and combined with future reception.

The difference between these two approaches matters in practice. Link-layer FEC preserves transmission integrity during a transmission interval. Application-layer FEC preserves data recoverability over objects such as files or extended video sequences across intervals.

This becomes especially important in environments where packets may be lost entirely across contact windows or delivery intervals, since link-layer FEC can only correct errors within packets that arrive, while application-layer FEC can help recover data when entire packets or groups of packets are missing.

V. Accumulation-Based Recovery Models

A. From Segment Completion to Information Accumulation

Traditional delivery models treat data as a sequence of segments that must each be received. Missing segments create gaps that must be filled for reconstruction to succeed.

Under these conditions, this model becomes inefficient. A receiver may collect most of the data but still be unable to reconstruct the complete object because a small number of specific segments are missing. Another receiver may be missing a different set of segments, leading to inconsistent completion across the receiver population.

An alternative approach is to treat delivery as a process of information accumulation.

In this model, each received packet contributes to the total information required to reconstruct the data. Missing packets reduce the available information but do not create fixed dependencies on specific segments. Recovery becomes possible once sufficient information has been accumulated, regardless of which specific packets were received.

This changes the delivery model to match how fragmented networks actually behave.

Delivery is no longer measured by completing a transmission interval, but by accumulating enough information to reconstruct the data.

This shift has practical implications. It reduces the importance of packet identity, allows receivers to progress independently, and enables delivery systems to make use of partial reception rather than discarding it.

B. Interleaving and Time-Distributed Redundancy

Packet loss in satellite communication systems is often bursty rather than random, with losses occurring in correlated intervals caused by fading, interference, or transitions in coverage.

In block-based FEC systems, packets are grouped and protected together. If a burst loss affects multiple packets within the same group, the redundancy associated with that group may be insufficient to recover the loss.

Interleaving addresses this problem by redistributing packets before transmission. Instead of transmitting related packets together, it spreads them across time. Packets that depend on each other for recovery are deliberately separated so that they are unlikely to be affected by the same burst of loss.

This can be understood in terms of exposure. Without interleaving, a short impairment may remove several packets from the same recovery group, rendering that group unrecoverable. With interleaving, the same impairment removes packets from different groups, allowing each group to retain enough information for recovery.

Time-distributed interleaving extends this concept by allowing redundancy relationships to span multiple transmission intervals. Recovery is no longer tied to a single moment in time. If packets are lost during one interval, the information needed to recover them may be received earlier or later.

This introduces time diversity. Because channel conditions vary, spreading redundancy across time reduces the likelihood that a single impairment will eliminate all of the information needed for recovery.

C. Rateless Erasure Coding

Rateless erasure coding extends the concept of information accumulation by removing the dependency on fixed packet groupings.

In a rateless system, each packet is an encoded combination of the original data set. The encoder can generate a continuous stream of such packets, each providing additional information about the original content.

Reconstruction depends on receiving a sufficient number of independent packets rather than specific packets, allowing recovery to continue across varying reception conditions and transmission intervals.

This approach differs fundamentally from both block-based FEC and repetition-based delivery. In block-based FEC, recovery depends on receiving a sufficient subset of packets from a specific group. If too many packets from that group are lost, recovery fails, even if additional packets are received later. In repetition-based delivery, recovery depends on eventually receiving the specific packets that were missed.

Packets received during different transmission intervals can contribute to the same reconstruction process. A later packet can compensate for one lost earlier, even though the two were not part of the same original grouping.

Packet identity and ordering become less critical to recovery. Losing a specific packet does not create a unique gap that must be filled, and packets received out of sequence can still contribute to reconstruction. Instead, recovery depends on accumulating sufficient independent information across reception intervals.

In multicast environments, this approach allows different receivers to reconstruct the same data even if they receive different subsets of packets. The system does not depend on per-receiver feedback or retransmission, and each receiver progresses based on the data it has successfully received.

VI. Delivery Control and Operational Tradeoffs

Application-layer recovery does not operate in isolation. Its effectiveness depends on how data is packetized, scheduled, prioritized, and transmitted relative to the operational needs of the mission.

For file-based delivery, the primary objective is usually completion: the receiver must eventually reconstruct the full object. For streaming or near-real-time workflows, the objective may differ. A receiver may need usable segments as early as possible, even if the full object is not yet complete. This creates a tradeoff between latency, redundancy, and recovery depth.

Smaller recovery windows can reduce delay because the receiver needs fewer packets before attempting reconstruction. However, smaller windows may also reduce resilience under burst loss. Larger recovery windows improve tolerance to loss and fragmented reception but can increase the time before reconstruction is possible. The appropriate balance depends on the use case.

For example, a SAR collection may tolerate delayed reconstruction if the complete data product is eventually recovered. A time-sensitive ISR workflow may prioritize earlier availability of partial products or selected data segments. A software update distributed to many receivers may prioritize efficient completion across the fleet rather than low latency for a single receiver.

Application-layer delivery control allows these tradeoffs to be managed above the transport layer. Data can be organized by object, priority, time sensitivity, or mission relevance. Redundancy can then be applied in a way that reflects the value and urgency of the data rather than treating all packets identically.

This application-level visibility matters in hybrid networks. Transport paths may differ in latency, reliability, and availability, but the application layer maintains knowledge of the data object being delivered. That allows recovery to be structured around the mission product rather than the behavior of any single link.

VII. Multicast Scaling and Receiver Independence

A central challenge in broadcast and multicast delivery is that receivers experience different loss patterns under the same transmission. Some may receive most packets during the first contact window, while others may miss large portions due to mobility, obstruction, interference, or timing.

Retransmission-based recovery becomes difficult in this setting because the sender must respond to many different receiver states. If each receiver requests different missing packets, the system either becomes a collection of individualized unicast recovery flows or must repeatedly transmit data that many receivers already possess. Both outcomes reduce the efficiency that motivated multicast delivery in the first place.

Application-layer FEC and rateless encoding change this behavior. The sender does not need to know exactly which packets each receiver missed. It continues transmitting encoded information, and each receiver uses whatever it successfully receives. Receivers progress independently toward reconstruction without requiring the sender to maintain detailed per-receiver recovery state.

This is especially relevant for distributed ground segments, mobile terminals, maritime platforms, airborne receivers, and other environments where receiver populations are large, moving, or operationally diverse. In such networks, uniform reception is unrealistic. Receiver independence becomes a scaling requirement.

The practical result is that multicast delivery can remain efficient even when reception is uneven. The system does not require every receiver to share the same packet history. Each receiver only needs to accumulate enough recoverable information to reconstruct the data object.

VIII. Operational Implications of Accumulation-Based Recovery

The practical effect of accumulation-based recovery is that packet loss changes from a blocking event into a slowdown in reconstruction. In traditional systems, missing packets create gaps that must be repaired before reconstruction can continue. In accumulation-based systems, recovery can continue as long as enough recoverable information keeps arriving.

This behavior is especially useful in hybrid architectures with multiple paths or intermittent reception. Data received over one path, session, or

contact window can be combined with data received later, allowing the system to make progress without depending on a single clean transmission interval.

IX. Implications for Ground Segment Design

As SmallSat systems mature, the ground segment becomes more than a passive endpoint for downlinked data. It becomes an active part of the delivery architecture. Ground segment software can determine how data is packetized, how redundancy is applied, how priorities are assigned, and how receivers reconstruct data under loss.

This is significant because many resilience improvements at the physical layer require changes to radios, antennas, payloads, or spectrum usage. Application-layer techniques can often be introduced above existing transport and link infrastructure. They do not eliminate the need for strong RF design, link-layer coding, or sound network engineering, but they add protection at the layer where the system understands the mission data itself.

This creates a useful separation of responsibilities. Link-layer mechanisms help maintain the integrity of the communication channel. Transport mechanisms move packets across available paths. Application-layer recovery protects the data object across gaps, paths, and time. In hybrid architectures, all three layers may be necessary, but they solve different problems.

For SmallSat operators, this layered approach is practical. It allows missions to use existing satellite links, terrestrial paths, and multicast distribution models while improving recoverability at the data-object level. It also supports future architectures where GEO, LEO, terrestrial IP, and 5G NTN paths may be combined in ways that are operationally useful but difficult to make perfectly continuous.

The broader implication is that resilience is not only a link property. In hybrid and intermittently connected architectures, it also depends on whether the delivery system can reconstruct usable mission data from incomplete reception over time.

X. Conclusion

SmallSat and hybrid space-ground communication systems operate under conditions where reception is often inherently fragmented.

By shifting to accumulation-based models and applying application-layer FEC, interleaving, and rateless encoding, delivery systems can operate effectively under intermittent and one-way conditions.

These techniques do not eliminate packet loss. They make it manageable by aligning the delivery model with the behavior of the communication environment.