

# IP Camera Integration to a Legacy FTI System

Eyup YAKA<sup>1</sup>, Dogukan CEVIK<sup>1</sup>, Tayfun SAHIN<sup>1</sup>, Aysu KIRDAR<sup>1</sup>, Yaren CEMAL<sup>1</sup>, Cenk OZDEMIR<sup>1</sup>,  
Mahmut Kursad ARPACIOGLU<sup>1</sup>

<sup>1</sup>Turkish Aerospace Inc. Fethiye Mah. Havacilik Bulvari No:16 Kahramankazan/Ankara/Turkiye  
[eyup.yaka@tai.com.tr](mailto:eyup.yaka@tai.com.tr), [dogukan.cevik@tai.com.tr](mailto:dogukan.cevik@tai.com.tr), [tayfun.sahin@tai.com.tr](mailto:tayfun.sahin@tai.com.tr), [aysu.kirdar@tai.com.tr](mailto:aysu.kirdar@tai.com.tr),  
[yaren.cemal@tai.com.tr](mailto:yaren.cemal@tai.com.tr), [cozdemir@tai.com.tr](mailto:cozdemir@tai.com.tr), [mkarpacioglu@tai.com.tr](mailto:mkarpacioglu@tai.com.tr)

## Abstract

In TA (Turkish Aerospace), video data is acquired through an analog camera integrated with the KAM500 data acquisition system during flight tests. However, low image quality and unsatisfactory clarity of the image are known issues of analog video acquisition. To eliminate the issues related with analog cameras, it is aimed to acquire video data using an IP camera which provides high definition images.

The integration of an IP camera to an existing KAM500 DAQ system (which includes EBM/102 module to get IP camera stream packages) is explained within a similar bandwidth usage on telemetry system compared to an analog camera video. To address the challenge, this study presents the development of specialized software tools operating independently on both the aircraft and the ground station. These tools parse and reconstruct the received ethernet packets into a suitable format and enabling the generation of a reliable video stream.

**Key words:** Camera, Flight Test Instrumentation, Ethernet, Video, Data Acquisition

## Introduction

TA is one of the Türkiye's leading companies in the defense and aerospace sector. The company operates in a wide range area including jet aircraft and helicopter design, production and modernization, UAV development, satellite and space systems, advanced research and development. TA has led major projects such as ANKA UAV, ANKA-3 UAV, Aksungur UAV, ATAK Helicopter, T-625 Gökbey Helicopter, Göktürk-3 Satellite, Hürkuş Trainer Aircraft, Hürjet Advanced Jet Trainer Aircraft and Fifth Generation Fighter Jet TF-X (KAAN). Flight tests are carried out during the aircraft development process. In these flight tests, various data are collected from the aircraft by the Flight Test Instrumentation teams in order to evaluate the aircraft design activities.

In aircraft flight operations, analog cameras are commonly used for the acquisition of video data within data acquisition systems. The use of analog cameras presents disadvantages in terms of image quality and clarity. During the downlink of data collected from the aircraft via telemetry, modules compatible with telemetry data transmission formats are used, and PCM (Pulse Code Modulation) broadcasting compliant with the CH4 standard is implemented using ENC (encoder) modules. In telemetry operations, in order to adjust the desired bitrate value, the occurrence and frequency values of

the analog camera video parameter data must be configured. Due to this configuration, achieving the desired bitrate for analog camera data may result in high bandwidth usage, as words that are not actually used in the generated telemetry packets still need to be transmitted.

For the reasons mentioned above, a solution has been targeted to improve the video data acquired during flight tests, ensuring that it maintains similar technical characteristics such as bitrate and frame rate, while providing better image quality, clarity, and continuity. For this purpose, the use of IP cameras has been selected.

In systems where live streaming is required, such as flight tests, IP camera enables the transmission of video data with a constant bitrate. The MPEG-TS (MPEG Transport Stream) format, developed for the live and stable transmission of multimedia data such as video, provides a packet structure that ensures a lossless and stable video stream. In MPEG-TS, packets are independent of each other, allowing the video to be reconstructed even in the event of a connection loss. In flight tests, low latency is critical in MPEG-TS streaming. Therefore, despite potential disadvantages such as packet loss, the system utilizes a TS over UDP architecture to benefit from low latency, transmitting video data acquired via IP cameras using the UDP protocol.

IP camera devices use digital video compression standards such as H.264, and by employing a constant bitrate, they provide superior performance compared to analog cameras in terms of quality and clarity.

### **Related Works**

The use of video systems in flight test instrumentation (FTI) has traditionally relied on analog cameras due to their simplicity and low latency characteristics. However, analog video acquisition systems are inherently limited in terms of resolution, signal quality, and noise immunity. These limitations become more critical in modern flight test environments where high-fidelity visual data is required for post-flight analysis and real-time monitoring. Recent aerospace instrumentation studies highlight that analog video systems are increasingly insufficient for high-resolution telemetry applications due to their susceptibility to electromagnetic interference and restricted bandwidth efficiency [1].

To overcome these limitations, IP-based camera systems have been widely adopted in aerospace and surveillance applications. IP cameras perform onboard digitization and compression of video streams, enabling high-definition (HD) or ultra-high-definition (UHD) transmission over Ethernet-based networks. This architecture eliminates the constraints of analog signal transmission and enables scalable integration into modern data acquisition systems. Aerospace-specific implementations demonstrate that Ethernet-based video systems can be integrated into flight test environments with minimal modification to existing telemetry architectures, particularly when using standardized compression formats such as H.264 and H.265 [2].

Recent developments in flight test data acquisition systems show a strong trend toward Ethernet-centric architectures. Systems such as modular DAQ platforms (e.g., KAM-series architectures) increasingly support IP-based video ingestion through dedicated interface modules (e.g., EBM/102 or equivalent Ethernet bridge modules). These modules enable the acquisition of packetized video streams and their synchronization with other flight test parameters. Studies in aerospace instrumentation emphasize that Ethernet-based video integration allows unified data handling across multiple sensors while maintaining deterministic timing constraints required in flight test operations.

However, the transition from analog to IP-based video systems introduces new challenges in terms of data handling and real-time processing. Unlike analog signals, IP camera outputs are transmitted as discrete Ethernet packets, which must be correctly parsed, reordered, and reconstructed to form a continuous video stream. Therefore, specialized software tools are required both onboard and at ground stations to process raw Ethernet streams into usable video formats.

Furthermore, bandwidth management remains a critical concern in flight test telemetry systems. Although IP cameras provide higher resolution, they also introduce higher data rates compared to analog systems. Recent aerospace studies indicate that careful compression configuration and adaptive streaming techniques are necessary to ensure that video transmission remains within telemetry bandwidth constraints without degrading other critical sensor data streams [2], [3]. This is particularly important in integrated FTI architectures where video shares network resources with telemetry and health monitoring data.

Overall, the literature clearly indicates a transition from analog to IP-based video acquisition in aerospace applications. While analog systems are still used in legacy configurations, modern flight test environments increasingly rely on Ethernet-based video systems due to their superior image quality, scalability, and integration capability. However, successful implementation requires robust packet processing, synchronization mechanisms, and careful bandwidth optimization to ensure reliable video delivery in real-time flight test operations.

## **Materials and Methods**

### ***Integration Problem***

Digital video compression standards such as H.264 transmit image data in frame structures. These frame types are I, B, and P.

I-Frame (Intra Frame) is a key frame that contains the entire image independently without requiring any previous or subsequent frames. It has the highest image quality and the largest data size, and serves as a reference point in the video stream.

P-Frame (Predicted Frame) is a frame that references the preceding I or P frame and stores only the changed regions. This reduces the data size; however, if the referenced frame is corrupted, the P-frame is also affected.

B-Frame (Bidirectional Frame) is generated through prediction using both previous and subsequent frames. It provides the highest compression ratio, but introduces higher computational load and latency.

To stabilize the bitrate of the IP camera video stream, technical parameters such as GOP, MTU, and frame type must be selected in accordance with the bitrate requirements of telemetry operations.

After these adjustments are made and the IP camera stream is successfully obtained, the Ethernet-based transmission will be collected by the data acquisition system. This will be performed using the KAM500 data acquisition device, which is obtained from Curtiss-Wright, with EBM/102/B module (ethernet bus monitor module), which is capable of accessing Ethernet packets on the ethernet network.

The IP camera video data collected via the KAM500 data acquisition device and the EBM/102/B module must be captured completely without any degradation or interruption. Therefore, the MPEG-TS stream generated by the IP camera has been analyzed.

An example of MPEG-TS stream packages are shown in Fig. 1. and Fig. 2.

| No. | Interval | Time     | Source         | Destination | Protocol | Length | Frame | Info                                       |
|-----|----------|----------|----------------|-------------|----------|--------|-------|--------------------------------------------|
| 216 | 0.000121 | 0.305161 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P21] fragment of a reassembled packet     |
| 217 | 0.000123 | 0.305284 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P22] fragment of a reassembled packet     |
| 218 | 0.000131 | 0.305415 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P23] fragment of a reassembled packet     |
| 219 | 0.000131 | 0.305536 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P24] fragment of a reassembled packet     |
| 220 | 0.000143 | 0.305669 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P25] fragment of a reassembled packet     |
| 221 | 0.000150 | 0.305798 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P26] fragment of a reassembled packet     |
| 222 | 0.000124 | 0.305907 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P27] fragment of a reassembled packet     |
| 223 | 0.000121 | 0.306053 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P28] fragment of a reassembled packet     |
| 224 | 0.000122 | 0.306155 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P29] fragment of a reassembled packet     |
| 225 | 0.000126 | 0.306255 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P30] fragment of a reassembled packet     |
| 226 | 0.000123 | 0.306378 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P31] fragment of a reassembled packet     |
| 227 | 0.000137 | 0.306495 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P32] fragment of a reassembled packet     |
| 228 | 0.000129 | 0.306638 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 1358   | ✓     | [P33] fragment of a reassembled packet     |
| 252 | 0.015517 | 0.342367 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 794    | ✓     | [0000] = 0000 Len=752 video-stream         |
| 277 | 0.019051 | 0.352853 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 794    | ✓     | [0000] = 0000 Len=752 video-stream         |
| 304 | 0.040826 | 0.423044 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 982    | ✓     | [0000] = 0000 Len=940 video-stream         |
| 330 | 0.019962 | 0.462006 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 1170   | ✓     | [0000] = 0000 Len=1138 video-stream        |
| 357 | 0.040809 | 0.501288 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 794    | ✓     | [0000] = 0000 Len=752 video-stream         |
| 383 | 0.019994 | 0.542098 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 794    | ✓     | [0000] = 0000 Len=752 video-stream         |
| 408 | 0.040234 | 0.582332 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 794    | ✓     | [0000] = 0000 Len=752 video-stream         |
| 438 | 0.019080 | 0.622137 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 1170   | ✓     | [0000] = 0000 Len=1138 video-stream        |
| 459 | 0.019911 | 0.662048 | 192.168.28.193 | 225.8.8.8   | PPS PES  | 794    | ✓     | [0000] = 0000 Len=752 video-stream         |
| 482 | 0.040996 | 0.701844 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [0000] = 0000 Len=1316 Program Association |
| 483 | 0.000000 | 0.701844 | 192.168.28.193 | 225.8.8.8   | PPS TS   | 1358   | ✓     | [P27] fragment of a reassembled packet     |

Fig. 1. An Example of MPEG-TS Stream Package

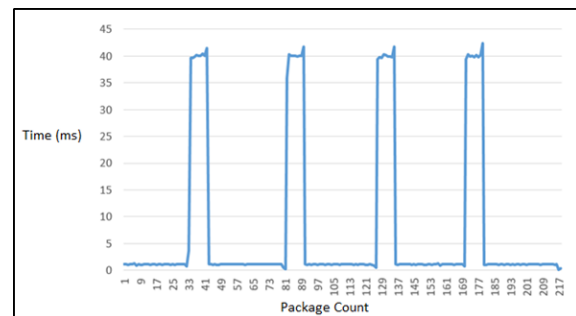


Fig. 2. Time Interval Between Consecutive Packages

Although the IP camera stream is configured to transmit at a constant bitrate, the packets do not arrive at fixed intervals, as can also be observed in the Fig.1. and Fig. 2. . In this case, in order for the Data Acquisition System to capture the MPEG-TS stream from the IP camera with a constant bitrate completely and without loss, very high bandwidth usage is required. The reason of this problem is due to the nature of the MPEG-TS streaming structure (Fig. 3.), each frame is generated with different sizes. Therefore, there is no fixed time interval between the transmission of the packets that compose the frames.

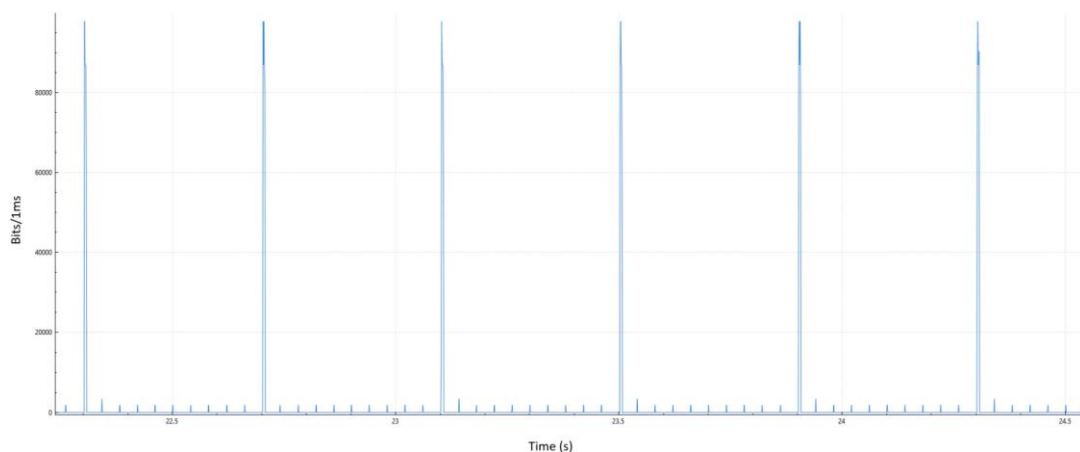


Fig. 3. Bitrate Graph of MPEG-TS Stream in Time-Domain

Data acquisition systems operate with fixed sampling frequencies. To ensure complete data capture, the sampling frequency of the KAM500 device must be adjusted based on the minimum time difference between packets. In this case, however, excessive sampling by the KAM500 device will occur, leading to unnecessary bandwidth usage.

Besides that, this problem was solved by Chapter 7 Packet Telemetry Downlink. Chapter 7 main objective is to multiplex different data types into a single telemetry downlink stream. All legacy data types that are analog sensor data, bus data, discrete data in addition to newer data types such as Gigabit Ethernet, High-Definition video can be packed into one single data stream. The biggest advantage of Chapter 7 is extremely

quick and easy system setup without the necessity to build-up time division multiplexed PCM frames nor to conduct its challenging validation. [5]

To overcome this asynchronously inserting data from one or more data streams into telemetry problem [6], the overall flight test instrumentation and telemetry system must be changed with equipments that are compitable with Chapter 7 Packet Telemetry Downlink. This process comes with it's huge cost. The overall system cost is approximately 130.000 USD.

This paper focuses on IP Camera video stream (which is over gigabit ethernet) integration to the legacy FTI systems with lower cost compared to the popular solutions for this integration problems.

In the flight operations carried out within Turkish Aerospace, the Curtiss-Wright KAM500 data acquisition system and the EBM/102/B Ethernet module are widely used. In order to obtain IP camera video data completely using the KAM500 data acquisition system equipped with the EBM/102/B Ethernet module, the timing irregularities present in the MPEG-TS stream must be resolved.

Each UDP packet transmitted over Ethernet in the MPEG-TS stream consists of seven pieces of H.264 data packets, each of 188 bytes, based on the applied configurations. With these defined parameters, the number of packets transmitted per second by the IP camera for video data

produced equivalently to analog camera output is examined in Fig. 4. . As an average value, 128 was deemed appropriate in accordance with the Nyquist theorem and selected as the packet transmission frequency. In other words, MPEG-TS packets from the IP camera should be transmitted sequentially at fixed time intervals of 8 ms.

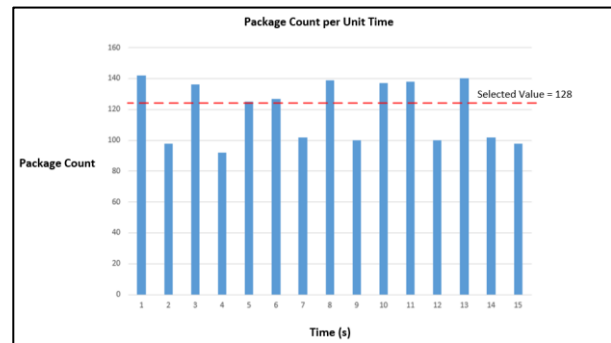


Fig. 4. Number of Packets Transmitted Per Second by the IP Camera

### Traffic Shaping Algorithm

There is no fixed time interval between the packets in a MPEG-TS stream. With the software to be developed, MPEG-TS stream packets should be captured and retransmitted at fixed time intervals. The representation of original MPEG-TS stream and desired MPEG-TS stream in time domain shown in Fig. 5. . The algorithm shown in Fig.6. has been used to perform the specified operation.

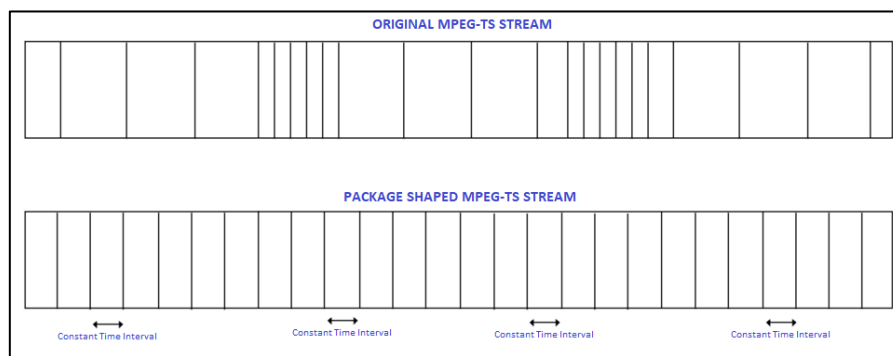


Fig. 5. Original MPEG-TS Stream and Desired MPEG-TS Stream in Time Domain

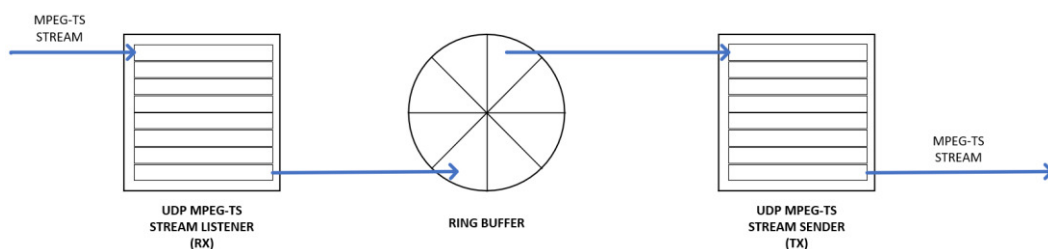


Fig. 6. Traffic Shaping Algorithm

### Evaluation of the Traffic Shaping Algorithm

In Fig. 7., the blue color represents the IP camera stream, while the black color represents

the output of the traffic shaping application. As can be seen in figure, a truly constant bitrate structure has been achieved.

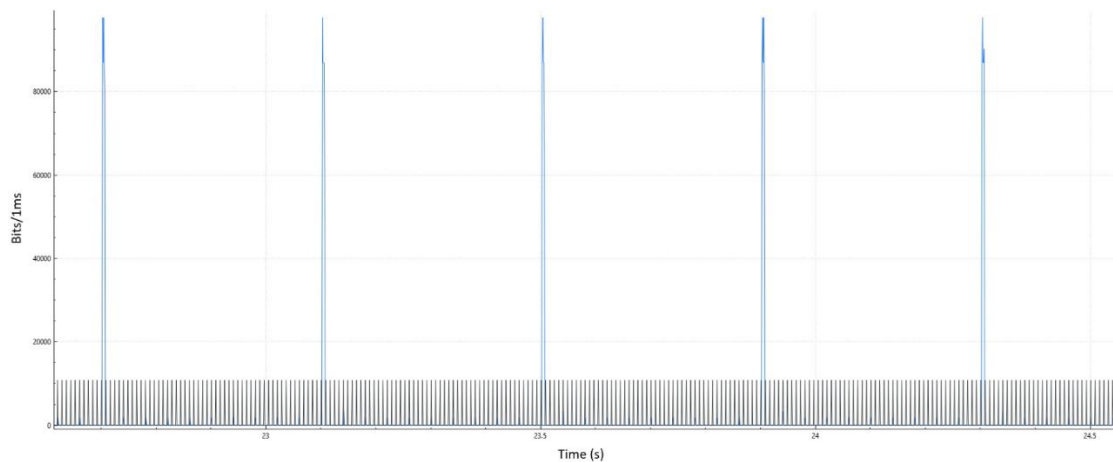


Fig. 7. Bitrate Graph of Original MPEG-TS Stream and Desired MPEG-TS Stream in Time Domain

Fig. 8. shows the time difference between consecutive packets graph of the MPEG-TS stream packets generated by the IP camera. As observed, the time difference between packets varies non-uniformly between 0 and 40 ms. Fig. 9. shows the inter-packet time graph of the MPEG-TS stream packets generated by the traffic shaping application. With the use of the traffic shaping application, the video data has been made suitable to be captured completely using a fixed sampling frequency by the KAM500 data acquisition device equipped with the EBM/102/B module, without incurring unnecessary bitrate usage. Without using traffic shaping application and approximately 1 Mbps bitrate usage condition, KAM500 data acquisition unit sampling points shown as red line in Fig. 10. Both in Fig. 10. and Fig. 11., KAM500 sampling rate is same (128 Hz).

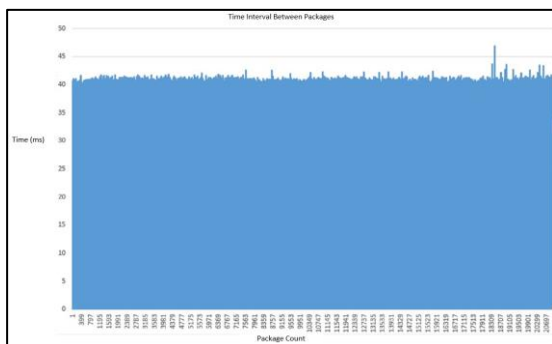


Fig. 8. Graph of Time Difference Between Consecutive Packages of Original MPEG-TS Stream

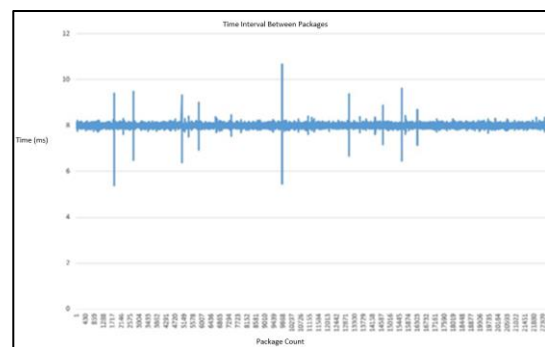


Fig. 9. Graph of Time Difference Between Consecutive Packages of Desired MPEG-TS Stream

As shown in Fig. 10., if aim is to achieve 1 Mbps video data stream bitrate usage, in this case KAM500 device misses a lot of packets. So, IP Camera stream cannot be gathered from this way.

With using traffic shaping application and approximately 1 Mbps bitrate usage condition, KAM500 data acquisition unit sampling points shown as red line in Fig. 11.

In Fig. 11., red lines represent KAM500 sampling points in time domain. With using traffic shaping application, KAM500 device do not miss any video packet that coming from IP Camera.

|     |                   |                |           |          |        |                                          |
|-----|-------------------|----------------|-----------|----------|--------|------------------------------------------|
| 174 | 0.000200 0.302206 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 1750 ✓ | 68458 - 8888 len1128 video-stream        |
| 176 | 0.000744 0.303840 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | 68458 - 8888 len1128 Program Association |
| 197 | 0.000000 0.303840 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 198 | 0.000154 0.303154 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 199 | 0.000024 0.303154 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 200 | 0.000129 0.303147 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 201 | 0.000129 0.303466 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 202 | 0.000118 0.303564 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 203 | 0.000187 0.303771 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 204 | 0.000148 0.303919 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 205 | 0.000021 0.303942 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 206 | 0.000236 0.304178 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 207 | 0.000022 0.304190 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 208 | 0.000115 0.304305 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 209 | 0.000154 0.304439 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 210 | 0.000118 0.304557 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 212 | 0.000121 0.304678 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 213 | 0.000121 0.304799 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 214 | 0.000128 0.304919 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 215 | 0.000121 0.305040 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 216 | 0.000121 0.305161 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 217 | 0.000121 0.305284 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 218 | 0.000131 0.305415 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 219 | 0.000111 0.305526 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 220 | 0.000143 0.305669 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 221 | 0.000181 0.305770 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 222 | 0.000122 0.305892 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 223 | 0.000121 0.306013 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 224 | 0.000122 0.306135 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 225 | 0.000128 0.306255 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 226 | 0.000121 0.306378 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 227 | 0.000121 0.306501 | 192.168.28.193 | 225.8.8.8 | MPEG TS  | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 228 | 0.000129 0.306630 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 1750 ✓ | video-stream                             |
| 232 | 0.015517 0.321167 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 794 ✓  | 68458 - 8888 len752 video-stream         |
| 277 | 0.019951 0.324984 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 982 ✓  | 68458 - 8888 len508 video-stream         |
| 284 | 0.000026 0.322084 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 1170 ✓ | 68458 - 8888 len508 video-stream         |
| 158 | 0.019994 0.462806 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 1170 ✓ | 68458 - 8888 len128 video-stream         |
| 257 | 0.000000 0.501168 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 794 ✓  | 68458 - 8888 len752 video-stream         |
| 163 | 0.019994 0.542050 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 794 ✓  | 68458 - 8888 len752 video-stream         |
| 418 | 0.046214 0.583132 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 794 ✓  | 68458 - 8888 len752 video-stream         |
| 418 | 0.019985 0.423217 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 1750 ✓ | 68458 - 8888 len1128 video-stream        |
| 419 | 0.019994 0.462806 | 192.168.28.193 | 225.8.8.8 | MPEG PES | 794 ✓  | 68458 - 8888 len752 video-stream         |

Fig. 10. 10 KAM500 Sampling Points

|     |                   |                |           |         |        |                                          |
|-----|-------------------|----------------|-----------|---------|--------|------------------------------------------|
| 158 | 0.000021 0.222007 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | 68272 - 9999 len1316 NULL packet         |
| 153 | 0.000017 0.230184 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | 68272 - 9999 len1316 NULL packet         |
| 158 | 0.007969 0.238073 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | 68272 - 9999 len1316 NULL packet         |
| 153 | 0.007968 0.246061 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 168 | 0.007922 0.253983 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 172 | 0.007558 0.262163 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 178 | 0.007954 0.270037 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | 68272 - 9999 len1316 NULL packet         |
| 183 | 0.008002 0.278039 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 187 | 0.007999 0.286137 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 191 | 0.008059 0.294096 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 195 | 0.008068 0.302164 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | NULL packet                              |
| 231 | 0.007951 0.310115 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | 68272 - 9999 len1316 Program Association |
| 236 | 0.008016 0.318131 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 241 | 0.008070 0.326201 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 246 | 0.008012 0.334213 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 251 | 0.007954 0.342167 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 257 | 0.007867 0.350034 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 262 | 0.007999 0.358036 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 267 | 0.008001 0.366035 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 272 | 0.007972 0.374087 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 278 | 0.008011 0.382019 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 283 | 0.007990 0.390008 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 288 | 0.007996 0.398004 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 293 | 0.008000 0.406004 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 299 | 0.007993 0.414087 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 305 | 0.007951 0.422084 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 310 | 0.008010 0.430054 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 315 | 0.008217 0.438271 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 320 | 0.007746 0.446017 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 325 | 0.008044 0.454061 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 331 | 0.007945 0.462086 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 336 | 0.008050 0.470071 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 341 | 0.007949 0.478028 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 346 | 0.008020 0.486048 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 352 | 0.007979 0.494023 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 356 | 0.008006 0.502019 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 362 | 0.008014 0.510083 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 367 | 0.008086 0.518165 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 372 | 0.008003 0.526187 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 377 | 0.008020 0.534217 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 382 | 0.007971 0.542050 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |
| 388 | 0.008073 0.550121 | 192.168.28.244 | 239.8.8.8 | MPEG TS | 1750 ✓ | [MP2T fragment of a reassembled packet]  |

Fig. 11. KAM500 Sampling Points

## Acquisition and Reconstruction of MPEG-TS Stream Via KAM500 After Capture and Transmission

The MPEG-TS stream obtained through the traffic shaping application is collected using the EBM/102/B module. This data acquired via the KAM500 data acquisition device can then be retransmitted in iNet-X format using the BCU (bus controller unit) module as shown in Fig. 12, or the video data can be broadcast as a PCM stream compliant with the CH4 standard using the ENC (encoder) module as shown in Fig. 13.

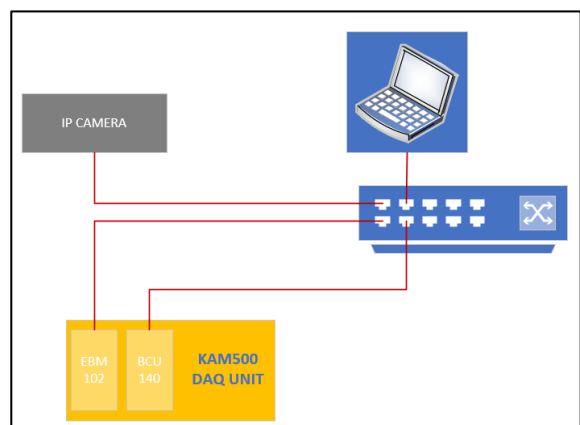


Fig. 12. Retransmitting Architecture of Desired MPEG-TS Stream in iNet-X Stream

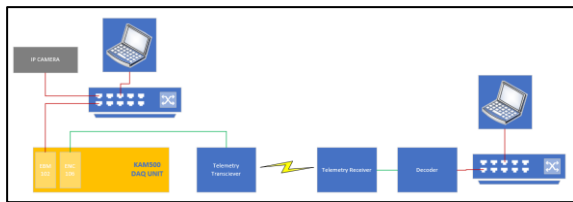


Fig. 13. Retransmitting Architecture of Desired MPEG-TS Stream via Telemetry

### Creation of MPEG-TS Stream From iNet-X Stream

The video data transmitted in the iNet-X streaming format using the BCU module must be converted into the MPEG-TS streaming format completely and without any loss in order to be viewable by users.

The video packet collected using the EBM/102/B module and retransmitted in iNetX format via the BCU module is shown in Fig. 14. MPEG-TS packet headers can be observed within the packet. An example for MPEG-TS package headers in UDP packages which is in iNet-X streaming format, shown in Fig.15.

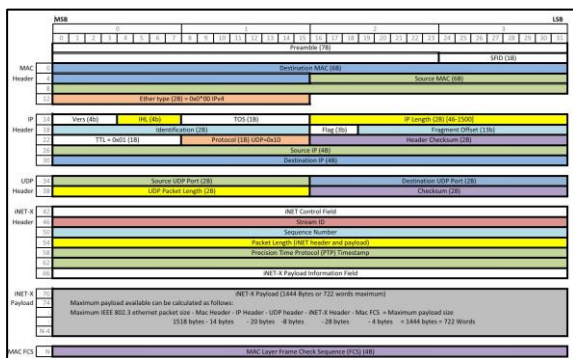


Fig. 14. iNet-X Streaming Format

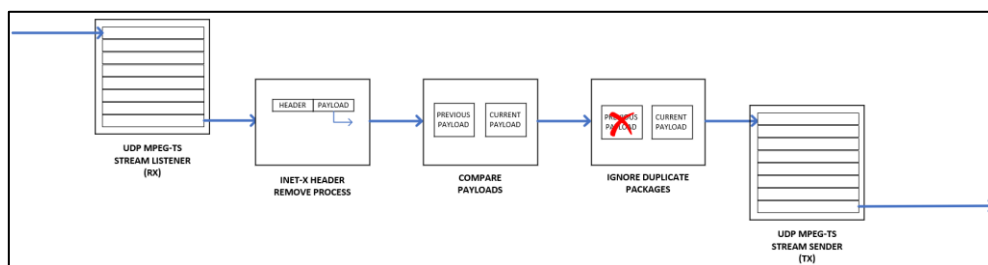


Fig. 16. The Architecture of Creation of MPEG-TS Stream

With the developed application, the MPEG-TS video data contained within the iNet-X packets received via the BCU module has been retransmitted in MPEG-TS format in a playable form, without any loss and without disrupting the sequence.

As can be observed from the Fig. 17., the raw video data was first completely captured using the traffic shaping application and retransmitted

|      |                         |                         |      |                         |                         |
|------|-------------------------|-------------------------|------|-------------------------|-------------------------|
| 0000 | 01 00 5e 00 00 01 00 0c | 4d 00 00 42 08 00 45 00 | 0138 | 05 03 d8 d6 20 9e a0 f5 | 00 34 13 84 72 e1 35 d6 |
| 0010 | 85 86 a5 f5 40 00 ff 11 | c7 d3 c0 a8 1c f3 eb 00 | 0140 | 70 e1 92 05 1c 06 0f 0e | 06 11 09 15 30 d9 f2 e0 |
| 0020 | 00 01 03 ff 07 6a 05 72 | 00 00 00 00 00 00 00 00 | 0150 | a1 ee 7a 53 fc 30 07 9f | f3 99 04 a0 f3 c2 81 c6 |
| 0030 | 00 20 03 d3 d1 fc 00 00 | 15 5a f9 1e af 7a 3a 4e | 0160 | a0 d3 0a c4 3a 20 41 53 | 20 38 9c 04 b4 f7 0a 62 |
| 0040 | 5e f7 00 00 00 00 01 00 | 5a 00 00 00 00 00 c1 76 | 0170 | 3b 5e f3 4c dd 7e 81 f9 | 40 7e 54 fb 1b 81 e7 48 |
| 0050 | 8b 11 08 00 45 00 05 40 | b7 a8 00 00 04 11 25 58 | 0180 | f9 93 13 be f0 af c8 4f | d5 f8 cd 99 63 a9 c2 64 |
| 0060 | c0 a8 1c f4 ef 08 08 08 | f9 68 27 0f 05 2c fc 44 | 0190 | 27 d4 a8 57 92 95 fb 94 | 85 9d 83 da 00 b0 14 c7 |
| 0070 | 07 00 2a 10 4e 25 4a    | 83 a5 17 5a 2d 30 c1 3e | 01a0 | 8a 9a 70 f9 0d 08 12 3e | da 07 02 81 0b 13 77 cd |
| 0080 | 8c c3 3f c4 f9 87 03 4c | 0a 8c 95 a3 ad 26 6c c5 | 01b0 | 5 c 02 d0 1b 40 47 e3   | 90 ef 6a 07 2a 70 99 72 |
| 0090 | 00 0e 6a 3c 1a 1d 01 44 | 6a 33 06 4b b0 2d 18 a3 | 01c0 | 2d 19 1b 47 65 d1 88 4f | 55 44 3b 00 83 28 33 d9 |
| 00a0 | c1 84 e9 29 3f 51 f7 21 | 8c 51 bc ff d9 72 8b ee | 01d0 | ce 6e cb 42 35 2f 5d 66 | a1 10 7b c4 16 63 6d bd |
| 00b0 | 09 dd c2 f9 f7 4c cb 63 | c1 79 ab 56 fd a9 04 1d | 01e0 | 93 dd a2 56 fa ab 41 9e | 07 00 21 1b 3e 1a f3 11 |
| 00c0 | 13 44 54 0e 11 b0 6f f9 | 81 40 23 20 2a fe f3 a6 | 01f0 | 05 1e c5 0e 07 79 95 1b | 03 50 6d 00 06 31 b0 66 |
| 00d0 | 95 37 f3 b0 12 06 04 d3 | 36 aa 4a 9c 9a c9 cc 2f | 0200 | 26 33 73 d4 1b 57 c7 04 | fa 54 30 38 01 01 f3 fd |
| 00e0 | 78 cd 75 06 07 b3 01 1b | 76 8d 92 06 58 9c 99 fa | 0210 | 02 4c dd 70 2b a9 6f 41 | 0e ed 2e a7 17 a8 4a be |
| 00f0 | a0 7c 25 a7 07 db 36 f5 | bc 51 3c 57 5c 98 7c 2a | 0220 | c3 ab cd d3 51 ed d7 f3 | a3 60 5f dd 44 d7 64 89 |
| 0100 | 05 4e 07 00 21 e0 41 25 | 75 07 7e a9 c0 ca 0c 01 | 0230 | 3b 9e a6 1c de 08 cf 66 | 1f c2 62 43 58 c3 46 ea |

Fig. 15. Example of MPEG-TS Package Headers in iNet-X Streaming Format

The MPEG-TS packets transmitted within the iNet-X structure must be retransmitted in a playable format as pure MPEG-TS, free from the iNet-X format. To achieve this, the headers identifying the iNet-X format within each packet must be removed, and the video data contained in the iNet-X structure must be properly transmitted. Additionally, in cases of undefined synchronization errors or when no packets remain in the traffic shaping application and no packets are sent, the same MPEG-TS packet may be retransmitted. When the same packet is sent again, video playback applications encounter continuity errors, preventing smooth video playback. An algorithm has been developed to address these issues. The architecture of creation of MPEG-TS Stream algorithm shown in Fig. 16.

contained within the iNet-X stream back into an MPEG-TS stream is shown in red on the Fig.17.

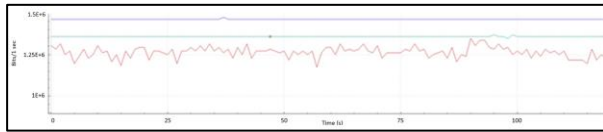


Fig. 17. Bitrate Graph of Desired MPEG-TS Stream, iNet-X Stream and Final MPEG-TS Stream Created from iNet-X Stream

With these processes:

- The IP camera MPEG-TS stream was first regularized according to time domain
- It was fully captured using the KAM500 data acquisition system
- It was retransmitted within iNet-X stream.
- An MPEG-TS stream was reconstructed from the video data within the iNet-X stream.

#### Creation of MPEG-TS Stream from Ch4 Pcm Broadcast

During flight operations, image data is downlinked from the aircraft to the ground via the Telemetry System. The Telemetry System receives the PCM stream generated by the ENC module located on the KAM500 data acquisition system. The structure of the PCM stream produced by the ENC module is shown in Fig. 18.

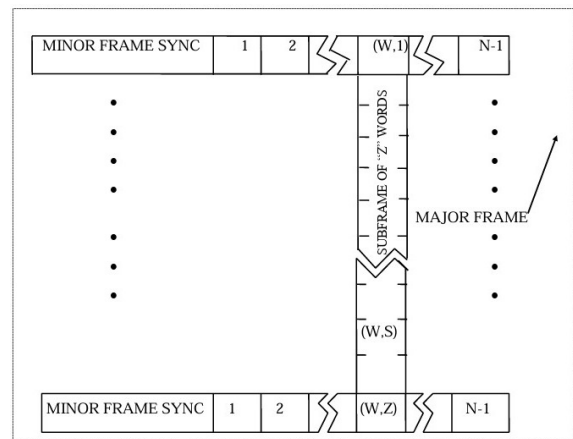


Fig. 18. Ch4 PCM Stream Format

In order for the video data transmitted via telemetry using the Chapter 4 PCM streaming format to be viewable by users, it must be converted into MPEG-TS streaming format completely and without any loss.

Chapter 4 PCM streaming is constructed using technical parameters such as "Words Per Minor Frame," "Minor Frames Per Major Frame," and "Sync Word." The PCM stream structure is formed in accordance with telemetry requirements.

The video data acquired from the aircraft is downlinked to the ground via the telemetry system during flight operations. There are significant differences between Chapter 4 PCM streaming and iNet-X streaming. Therefore, a new study has been conducted to generate an MPEG-TS stream from Chapter 4 PCM streaming. The operational model of the algorithm is shown in Fig. 19.

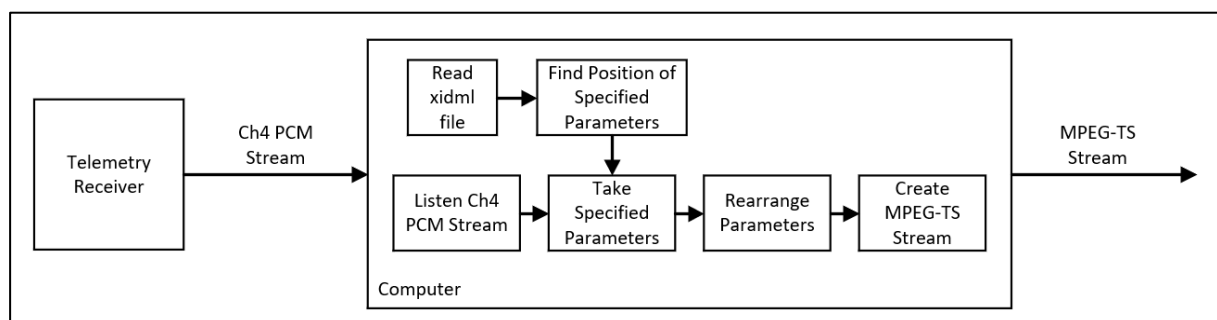


Fig. 19. CH4 PCM to MPEG-TS Stream Algorithm

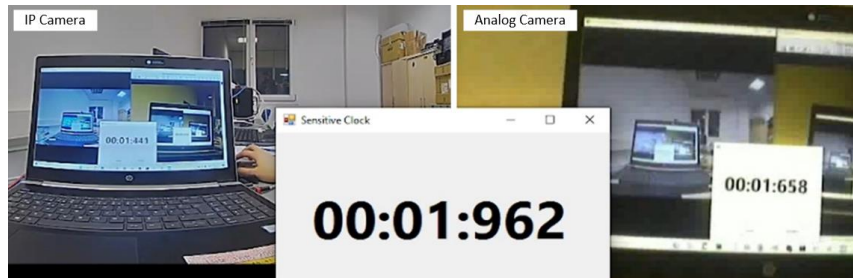
The configurations of the KAM500 data acquisition system are performed via the DAS Studio application. As a result of the configurations made through DAS Studio, a file with the ".xidml" extension is generated. The characteristics of the Chapter 4 PCM stream

produced by the KAM500 data acquisition system are contained within this file. In order to generate an MPEG-TS stream from the Chapter 4 PCM stream, the ".xidml" file must be analyzed. In this way, the positional data where

the parameters required for MPEG-TS stream generation are located can be identified.

With the developed application, the Chapter 4 PCM stream received via the Telemetry Receiver is monitored. The parameters specified by the user are extracted, and the MPEG-TS stream is reconstructed using these parameters.

The stability and behavior of the system have been evaluated through long-term tests. The results obtained from these measurements are presented in *Fig.20.* and Table 1. The table also includes the latency values of both the currently used system and the developed system. As can be seen in Table 1, compared to the system in which image data is obtained via an analog camera, the image data has been improved with approximately 200 ms less delay.



*Fig. 20. Measured Time*

*Table 1 Time Difference Table*

| Current Timestamp (ms) | Analog Camera Timestamp (ms) | IP Camera Timestamp s(ms) | Time Difference Between Analog Camera and IP Camera(ms) | Time Difference Between Analog Camera and Real Time(ms) | Time Difference Between IP Camera and Real Time(ms) |
|------------------------|------------------------------|---------------------------|---------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------|
| 1962                   | 1658                         | 1441                      | 217                                                     | 304                                                     | 521                                                 |

## Future Works

Future studies may focus on further improving bandwidth efficiency and reducing latency in IP camera based telemetry systems. Adaptive traffic shaping algorithms can be developed to dynamically adjust packet transmission intervals according to varying telemetry loads and network conditions during flight tests. In addition, advanced video compression standards such as H.265/HEVC or AV1 may be investigated in order to achieve higher image quality at lower bitrate values.

Another important research area is the enhancement of transmission reliability under harsh airborne communication conditions. Packet loss recovery mechanisms, forward error correction techniques, and synchronization improvements can be integrated into the developed software infrastructure to provide more robust and uninterrupted video streaming. Finally, the compatibility of the proposed architecture with next-generation telemetry systems and higher bandwidth communication infrastructures can be evaluated to support ultra-

high-definition and low-latency video transmission requirements in future aerospace applications.

## Results

Currently, within Turkish Aerospace Industries, the low image quality obtained from analog cameras and the lack of a constant bitrate negatively affect telemetry operations. When a configuration that achieves the same bitrate values is applied to IP cameras, a lossless and stable video stream can be achieved through the packet structure of the MPEG-TS streaming format, which is designed for the live and stable transmission of video data.

Without using developed traffic shaping application, IP Camera video stream could be taken without package loss by configuring KAM500 sampling rate. This sampling rate must be chosen according to minimum time difference between packages. In scenario that is mentioned in this paper, the minimum time difference is approximately 100 us. To achieve this sampling time period, sampling rate must be 10 kHz or

more than 10 kHz. Efficiency of this process according to bitrate usage calculated as follows;

*IP Camera Bitrate (Actual bitrate)  $\approx 1$  Mbps*

*KAM500 data stream bitrate  $\approx 1.5$  Mbps*

$$\text{Efficiency} \approx \frac{1}{100} \times 100 \approx \%1$$

In this case, bitrate usage efficiency is approximately %1. This means that %99 percent of bitrate is actually wasted.

Without using developed traffic shaping application, IP Camera video stream packets could be taken from traffic shaping application output video stream. KAM500 sampling rate must be chosen according to minimum time difference between packages. In scenario that is mentioned in this paper, the minimum time difference is approximately 8 ms. To achieve this sampling time period, sampling rate must be 128 Hz or more than 128 Hz. Efficiency of this process according to bitrate usage calculated as follows;

*IP Camera Bitrate (Actual bitrate)  $\approx 1$  Mbps*

*Traffic Shaping Output bitrate  $\approx 1.4$  Mbps*

*KAM500 data stream bitrate  $\approx 1.5$  Mbps*

$$\text{Efficiency} \approx \frac{1}{1.5} \times 100 \approx \%66.67$$

In this case, bitrate usage efficiency is approximately %66.67. This value means that traffic shaping application removes wasting bitrate usage.

In this system where IP cameras are used, it has been observed that image quality improves and bitrate values remain stable in a way that does not affect telemetry operations. This provides high-quality and uninterrupted video streaming to test observers during flight tests.

The improvements obtained in video data through the use of IP cameras are as follows:

- It has been observed that images obtained from IP cameras have higher light sensitivity.
- It has been observed that the edges and characteristic features of objects in IP camera images are more distinct.
- No pixel degradation or freezing has been observed during sudden scene changes.

A comparison of video data obtained using analog and IP cameras is shown as examples in Fig. 21, Fig. 22.

As can be seen in Fig. 21., the use of an IP camera has resulted in better performance in technical aspects that directly affect video viewing quality, such as image quality and clarity.

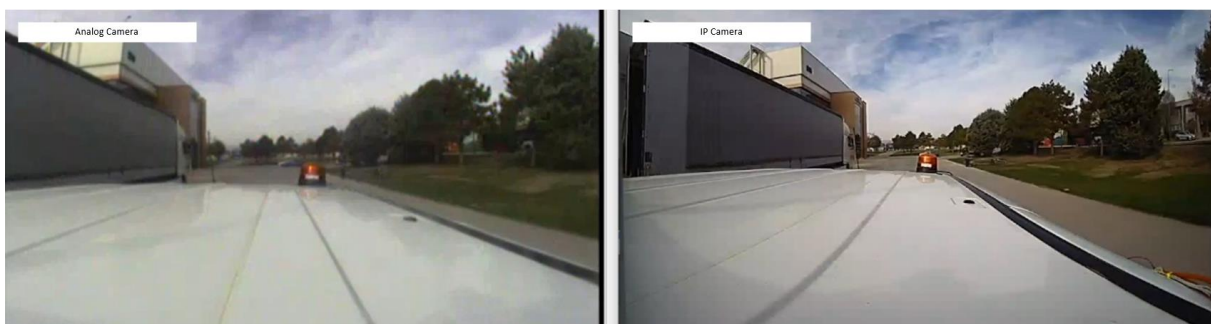


Fig. 21. Comparison of IP Camera and Analog Camera-1



*Fig. 22. Comparison of IP Camera and Analog Camera-2*

## References

- [1] M. Tavana et al., "Wireless Power Transfer for Aircraft IoT Applications: System Design and Measurements," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11834–11846, 2021.
- [2] D. Han, Y. Dai, and H. Minn, "Modeling Ground-Air Wireless Connectivity: Continuous Connection Probability Analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3611–3625, 2021.
- [3] S. Tahir et al., "A Novel Approach to Reduce Breaches of Aircraft Communication Data," *Electronics*, vol. 12, no. 1, 2023.
- [4] D. Jiang et al., "Research Progress of High-Speed Data Transmission Technology for Lightweight UAV Platforms," *Applied Sciences*, vol. 14, no. 6, 2024.
- [5] Safran, "Why IRIG 106 Chapter 7?" A Technical Introduction to Chapter 7
- [6] "Chapter 7 Packet Telemetry Downlink" [Online]. Available: [irig106.org/docs/106-20/chapter7.pdf](http://irig106.org/docs/106-20/chapter7.pdf)