# Inter-protocol Steganography for Real-time Services and Its Detection Using Traffic Coloring

Florian Lehner[1], Wojciech Mazurczyk[1,2], Jörg Keller[1], Steffen Wendzel[3,4]

[1] University of Hagen, Germany
[2] Warsaw University of Technology, Poland
[3] Worms University of Applied Sciences, Germany
[4] Fraunhofer FKIE, Germany

# Goals

Steganography (hiding secret data) for streaming protocols is already known. For this paper, we had the following goals:

- Showing that inter-protocol steganography for streaming protocols is feasible, i.e. hiding secret data using the relationships between at least two protocols.
- Determining potential hiding methods.
- Evaluating these hiding methods.
- Performing first analysis on the detectability of the proposed methods.

# Introduction

**Why utilize IP telephony or video streaming for information hiding?**

- *Popularity of real-time services:* usage will not raise suspicions (such traffic will not be considered an anomaly).

# Introduction

**Why utilize IP telephony or video streaming for information hiding?**

- *Popularity of real-time services:* usage will not raise suspicions (such traffic will not be considered an anomaly).
- *Use of a variety of cooperating protocols:* provides several opportunities for data hiding in different network layers.

# Introduction

**Why utilize IP telephony or video streaming for information hiding?**

- *Popularity of real-time services:* usage will not raise suspicions (such traffic will not be considered an anomaly).
- *Use of a variety of cooperating protocols:* provides several opportunities for data hiding in different network layers.
- *High steganographic bandwidth:* e.g., during an G.711-based IP telephony call the RTP stream rate is 50 packets per second. Thus, even hiding only 1 bit in every RTP packet results in a bandwidth of 50 bit/s.

# Introduction

**Typical real-time service connection consists of two phases:**

- *Signaling phase:* messages are exchanged to set up and negotiate the connection parameters.
- *Conversation phase:* one or more real-time data streams are sent between the parties (e.g. for IP telephony there are two audio streams).

# Introduction

**Typical real-time service connection consists of two phases:**

- *Signaling phase:* messages are exchanged to set up and negotiate the connection parameters.
- *Conversation phase:* one or more real-time data streams are sent between the parties (e.g. for IP telephony there are two audio streams).

A popular protocol in this context is **RTP** (*Real-time Transport Protocol*). RTP is accompanied by **RTCP** (*Real-time Control Protocol*).

# Introduction

**Inter- and Intra-protocol Steganography:**

- *Intra-protocol Steganography:* utilizes covert channels inside *one protocol*, e.g. embedded into DNS, but no other protocol.
- *Inter-protocol Steganography:* utilizes relationships between multiple protocols.

**Inter-protocol Steganography Examples:**

- *PadSteg* embeds secret data into the Ethernet frame padding if certain types of higher-level protocol are embedded.
- *StegSuggest* utilizes suggestions presented during usage of Google Web search. Modifications of TCP (window size, timestamp options) influence modifications of HTTP (by adding suggestions which contain secret data).

# Introduction

**Similar Approaches**:[1]

- *Protocol Hopping Covert Channels* utilize several protocols within the same network packet and can change the utilized protocol for every new packet. Secret data can also be spread over multiple layers.
- *Protocol Switching Covert Channels* represent hidden information through the order of network protocols used in a flow.

[1] see: S. Wendzel & S. Zander: Detecting Protocol Switching Covert Channels, LCN 2012, IEEE, 2012. and: W. Mazurczyk *et al.*: Information Hiding in Communication Networks, Wiley-IEEE, 2016.

# Fundamentals

**RTP** provides end-to-end communication transmitting real-time data such as audio or video, via multicast or unicast.

- Usually encapsulated in UDP
- Packet frequency and size depends on negotiated multimedia codec.
- RTCP provides monitoring, controlling and identification of RTP streams (approx. 5% of the RTP session bandwidth).
- RTCP knows five message types, of which *Sender Report* (SR, for transmission and reception statistics from active senders) and *Receiver Report* (RR, like SR but for non-active senders) are the most frequently used.

# Fundamentals

**RTP** provides end-to-end communication transmitting real-time data such as audio or video, via multicast or unicast.

- Usually encapsulated in UDP
- Packet frequency and size depends on negotiated multimedia codec.
- RTCP provides monitoring, controlling and identification of RTP streams (approx. 5% of the RTP session bandwidth).
- RTCP knows five message types, of which *Sender Report* (SR, for transmission and reception statistics from active senders) and *Receiver Report* (RR, like SR but for non-active senders) are the most frequently used.

We tried to find unique RTP/RTCP relationships that can be utilized for steganographic purposes.

# Analysis of RTP/RTCP Related Fields

Several fields of the RTCP header cannot be used for a covert channel, e.g.:

- *Synchronization source identifier* (SSRC) remains static throughout the RTP session.
- *RTP timestamp, Extended highest sequence number received, Cumulative number of packets lost, Interarrival jitter* and *Sender's counts*: modifications would be conspicuous and easily detectable.

# Analysis of RTP/RTCP Related Fields

**Suitable Candidates**:

- *NTP Timestamp (RTCP):* 64-bit value; includes time when SR/RR was sent. Receiver can then calculate, by subtraction of the timestamp when this report was received, how long it takes to transmit packets. *Value varies from packet to packet*.

# Analysis of RTP/RTCP Related Fields

**Suitable Candidates**:

- *NTP Timestamp (RTCP):* 64-bit value; includes time when SR/RR was sent. Receiver can then calculate, by subtraction of the timestamp when this report was received, how long it takes to transmit packets. *Value varies from packet to packet*.
- *Fraction Lost (RTCP):* number of lost RTP packets since previous SR/RR; reset each time an SR/RR is received. Besides real lost RTP packets, a modified sequence number or holdup of RTP packets will also have an effect on this field.

# Analysis of RTP/RTCP Related Fields

**Suitable Candidates**:

- *NTP Timestamp (RTCP):* 64-bit value; includes time when SR/RR was sent. Receiver can then calculate, by subtraction of the timestamp when this report was received, how long it takes to transmit packets. *Value varies from packet to packet*.
- *Fraction Lost (RTCP):* number of lost RTP packets since previous SR/RR; reset each time an SR/RR is received. Besides real lost RTP packets, a modified sequence number or holdup of RTP packets will also have an effect on this field.
- *Delay since last SR (RTCP):* is part of SR/RR and should be rather constant with small variations (e.g. due to lost SRs/RRs; kept in units of $1/65536s$).

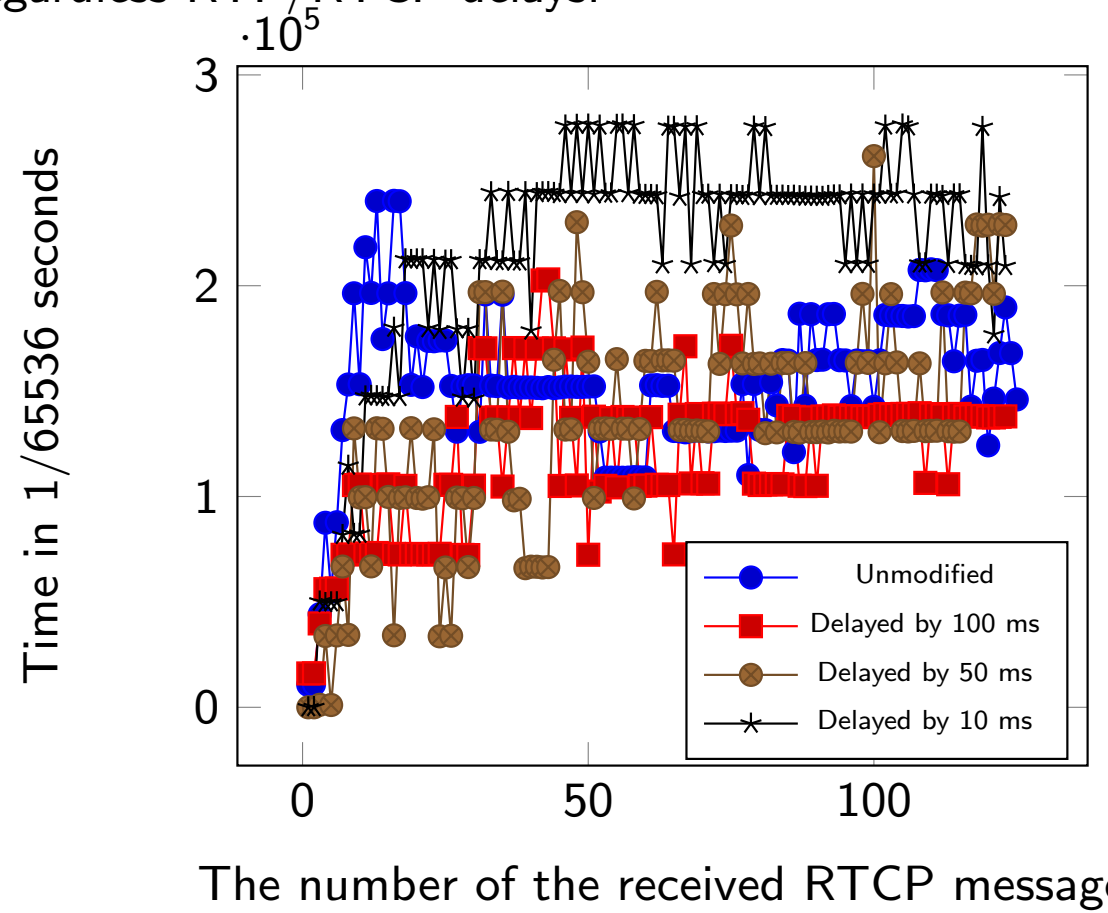
# Pre-evaluation of Delay Effect

Started with *analyzing the impact when RTP packets are subjected to delays* using *Opus codec* and speech payload for 20ms. Added delay on top of the network's delay; VoIP clients using 60ms jitter buffer.

Testbed using Debian 8 (Linux 3.16.7) and *Linphone* (an open source VoIP client) with accounts from *sip.linphone.org* to establish VoIP sessions. Communication between two clients was routed through the Internet.

# Pre-evaluation of Delay Effect

Impact of the delay on 'Delay since last SR/RR' values; time between two RTCP messages converged to a certain level after approx. 20 received RTCP messages, regardless RTP/RTCP delays.




**Result:** modifying RTP delay after RTCP packet 20 has an recognizable impact on RTCP messages that should allow covert signaling.

# Proposed Hiding Methods

**Module 1:** The *Fraction Lost* field is used in combination with the *Round Trip Time* (RTT) of the RTCP messages (influenced by dropped RTP packets).[2]

[2]The RTT is calculated as the time of arrival of the RTCP message subtracted by the *Delay since last SR/RR* and timestamp of the last SR/RR. These two elements have been selected to form the additional payload, as they are not relying on each other.

# Proposed Hiding Methods

**Module 1:** The *Fraction Lost* field is used in combination with the *Round Trip Time* (RTT) of the RTCP messages (influenced by dropped RTP packets).[2]

**Module 2:** The *Fraction Lost* field is only used for signaling purposes (to inform the covert receiver about the number of transmitted secret bits).
The hidden data relies on the timestamp of the last received RTP packet in combination with the value of the last RTP timestamp within a RTCP message. Both values are then compared. Equal values are interpreted as 0, otherwise 1.

[2]The RTT is calculated as the time of arrival of the RTCP message subtracted by the *Delay since last SR/RR* and timestamp of the last SR/RR. These two elements have been selected to form the additional payload, as they are not relying on each other.

# Three-phase Approach

**Initial/Measurement Phase:** To achieve a reliable data transfer, the covert communication does not start before the 20th RTCP message was received.

- Data from the last ten received RTCP messages is collected to calculate average values for Fraction Lost and RTT (continues till end of VoIP session).

# Three-phase Approach

**Signaling Phase:** informs the peer about the number of hidden bits to be transferred – basically a simple control protocol:[3]

- Is performed for both modules using the Fraction Lost field. Therefore, the sending side drops selected RTP packets. Afterwards, the receiving side will be informed about the *number of hidden bits* to be sent:
  - Module 1: encoded within the Fraction Lost field
  - Module 2: encoded in the difference between the timestamp of the last received RTP packet and the RTP timestamp within a received RTCP message

[3] see: S. Wendzel & J. Keller: Hidden and Under Control: A Survey and Outlook on Covert Channel-internal Control Protocols, in: Annals of Telecommunications (ANTE), Springer, 2014.

# Three-phase Approach

**Covert Communication Phase:** transfers the hidden information.

- Module 1: signals secret data by influencing the Fraction Lost field and calculated RTT values (caused by dropped RTP packets).
  Hidden data will be signaled alternating between Fraction Lost values (in our case: 2 bits) and RTT (1 bit).

- Module 2: same as in signaling phase (RTP/RTCP timestamp differences).

# Three-phase Approach

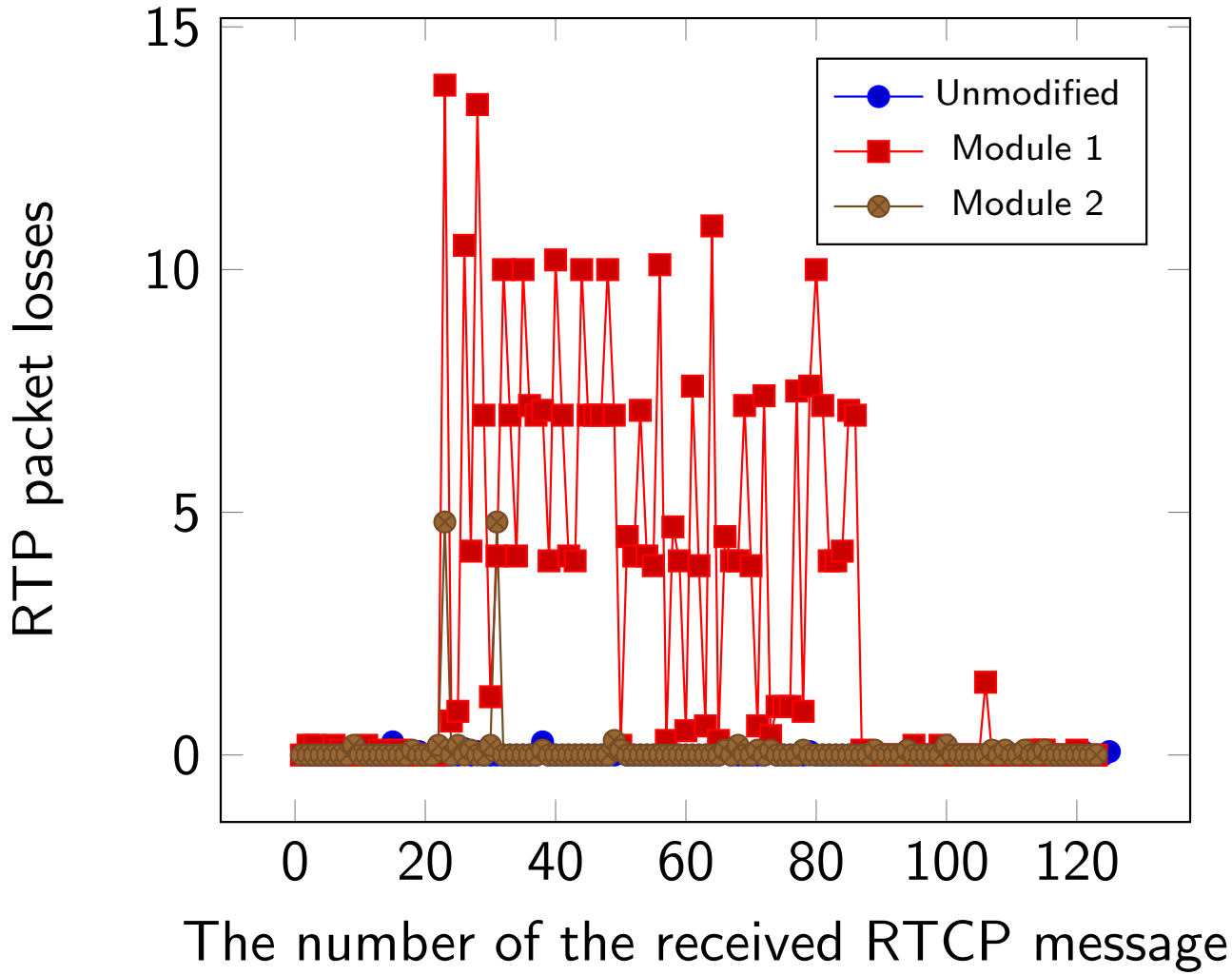
**Covert Communication Phase:** transfers the hidden information.

- Module 1: signals secret data by influencing the Fraction Lost field and calculated RTT values (caused by dropped RTP packets).
  Hidden data will be signaled alternating between Fraction Lost values (in our case: 2 bits) and RTT (1 bit).

- Module 2: same as in signaling phase (RTP/RTCP timestamp differences).

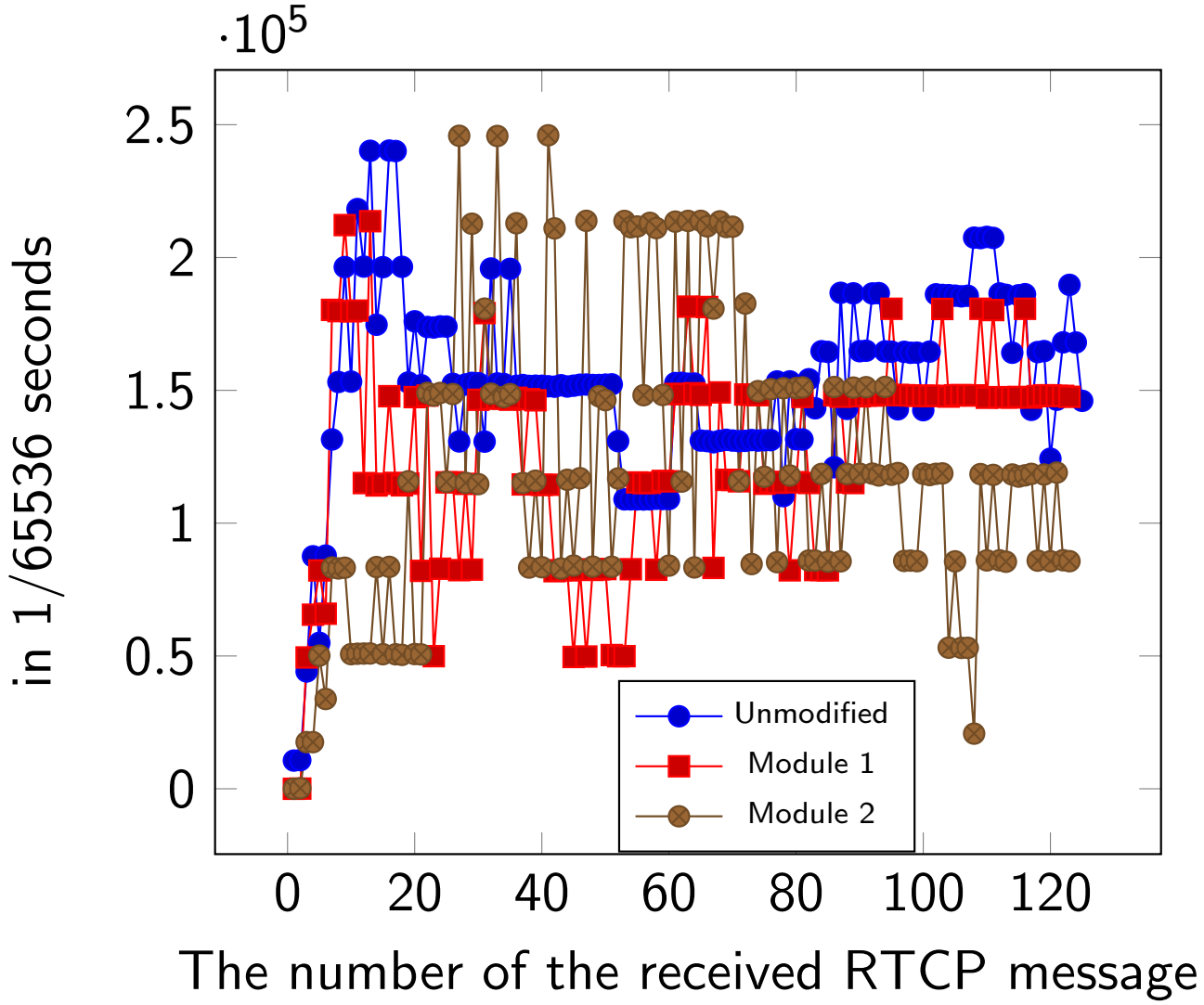
Implemented both modules as LKMs for Linux; they work independently of the used VoIP clients.

# Evaluation: Impact of M1+M2 on 'Fraction Lost' Field



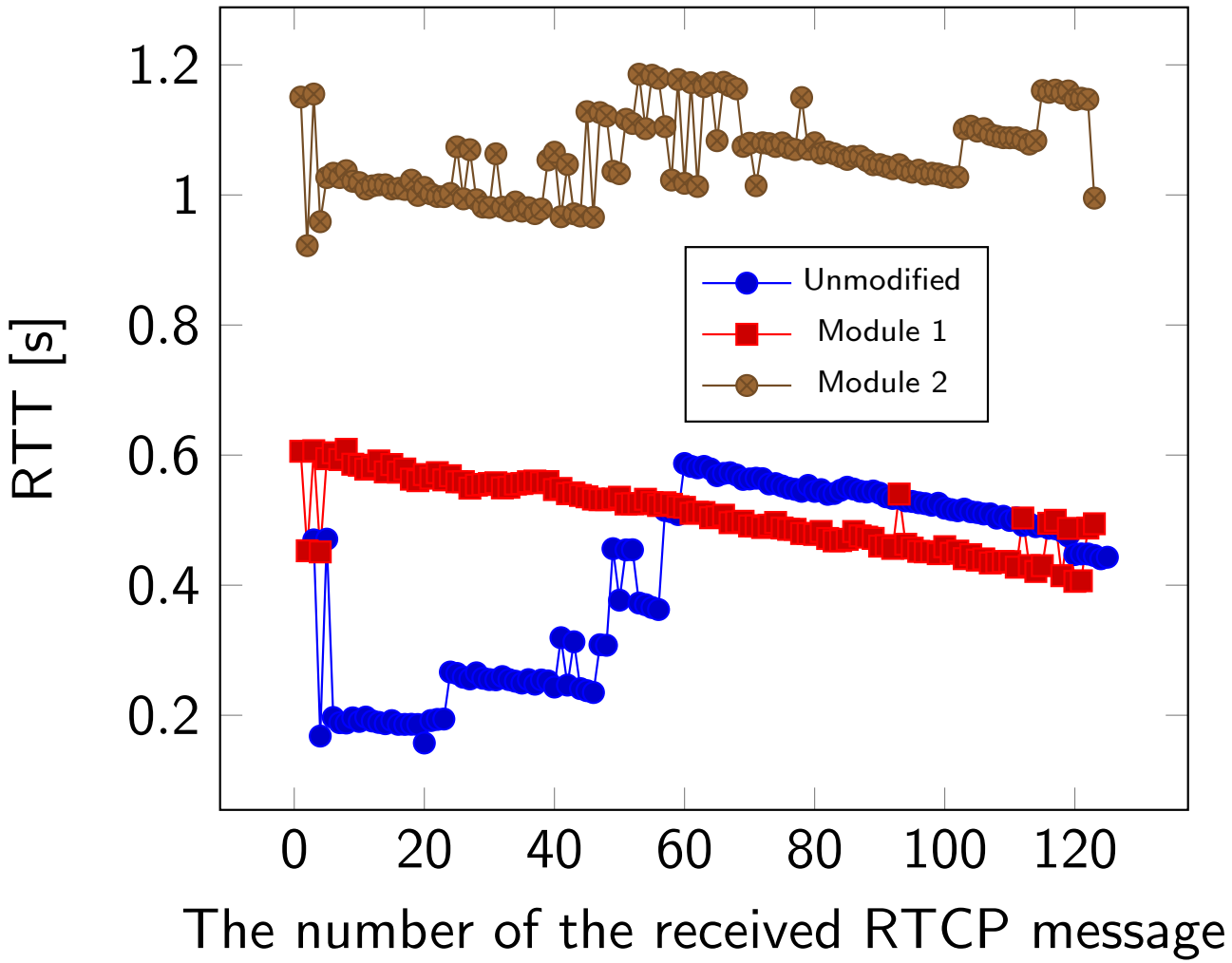

**Signaling:** Pkts. 23-28/31; **Transmission:** M1: 23-86 (identifiable; 5:55min for 175 bits), M2: not identifiable in Fig.; **Coding scheme:** identifiable for M1(+M2) (1,3,7,10,13).

# Evaluation: Impact on 'Delay since last SR/RR' Field




Transmission cannot be too easily identified for M1, however, some clearer differences for M2 after signaling phase.

# Evaluation: Impact of M1+M2 on the Round Trip Time




Although M2 does not signal using the RTT, its influence on the RTT is higher than in case of M1.

# Traffic Coloring Approach

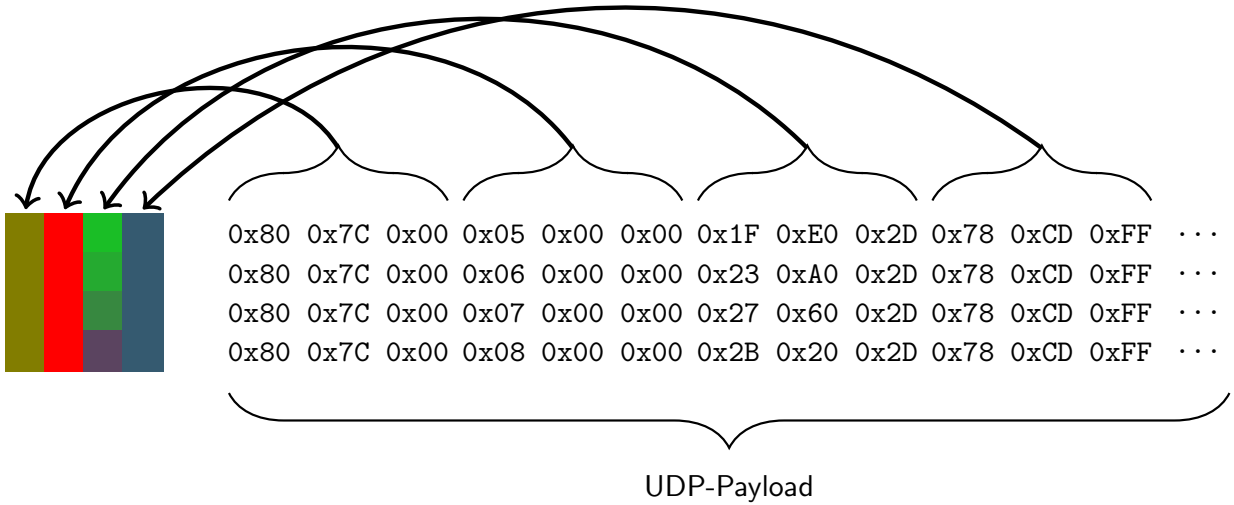



Fig.: Coloring of network traffic.

Focusing on VoIP, i.e. lower-level headers (UDP, IP) not included.
Incoming and outgoing traffic is colored separately.
Time-based steganography reflects on different intervals between network packets, whereas storage-based steganography changes the regular color structures.

# Traffic Coloring Approach

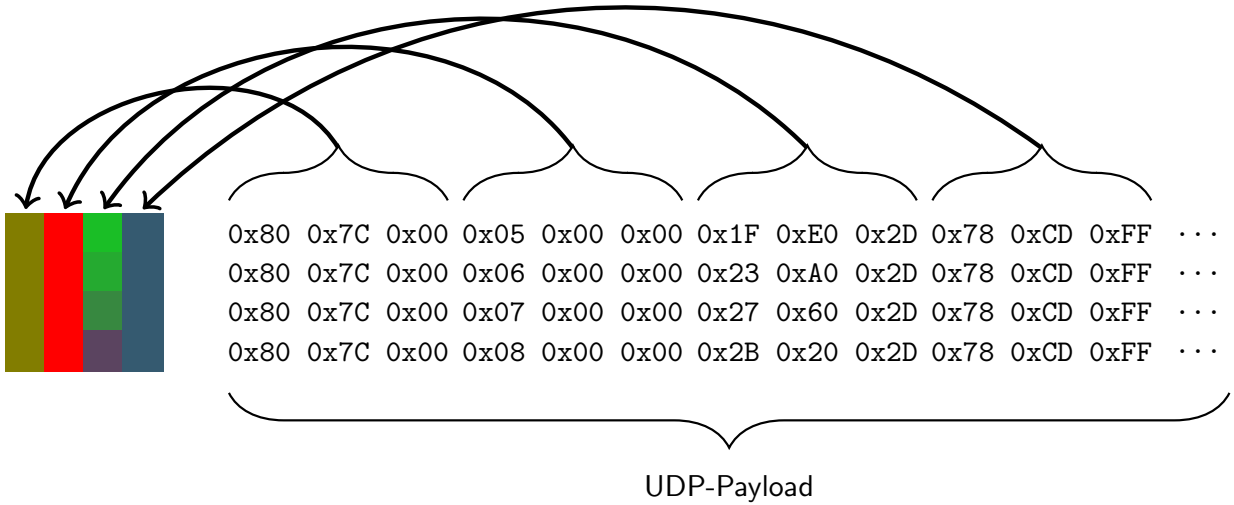
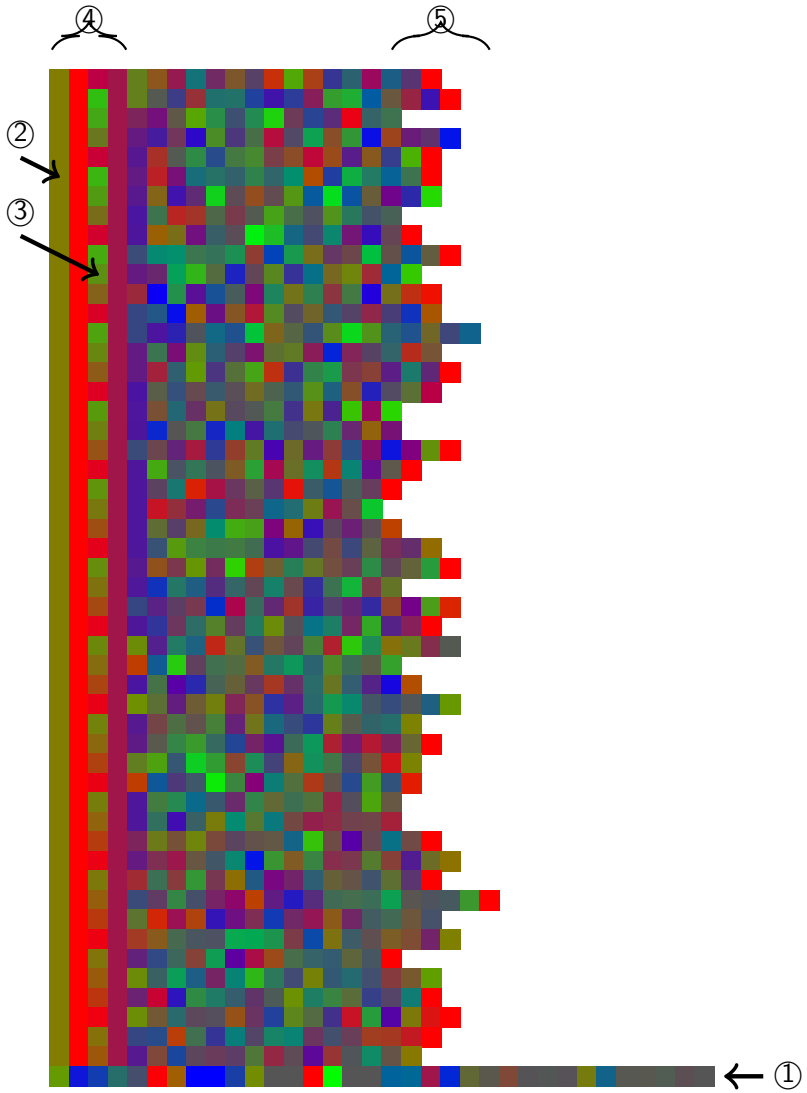



Fig.: Coloring of network traffic.

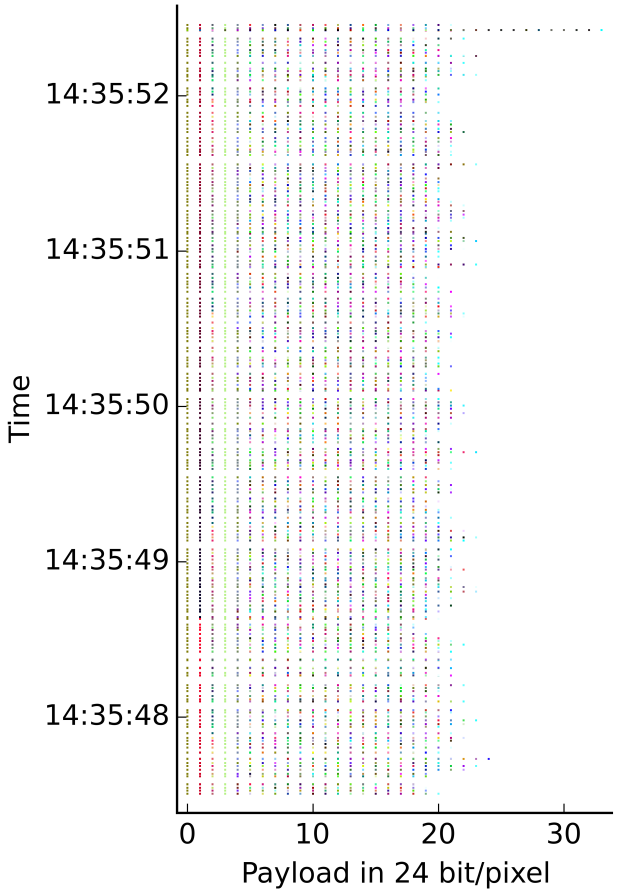
Used a script to extract relevant meta-data to generate images (e.g. 1 image/5$s$).
Data kept in SVG format (XML) to be machine-readable and ease image (meta data) processing.
Visual representation aids human analyst (visual analytics).

# Traffic Coloring Approach

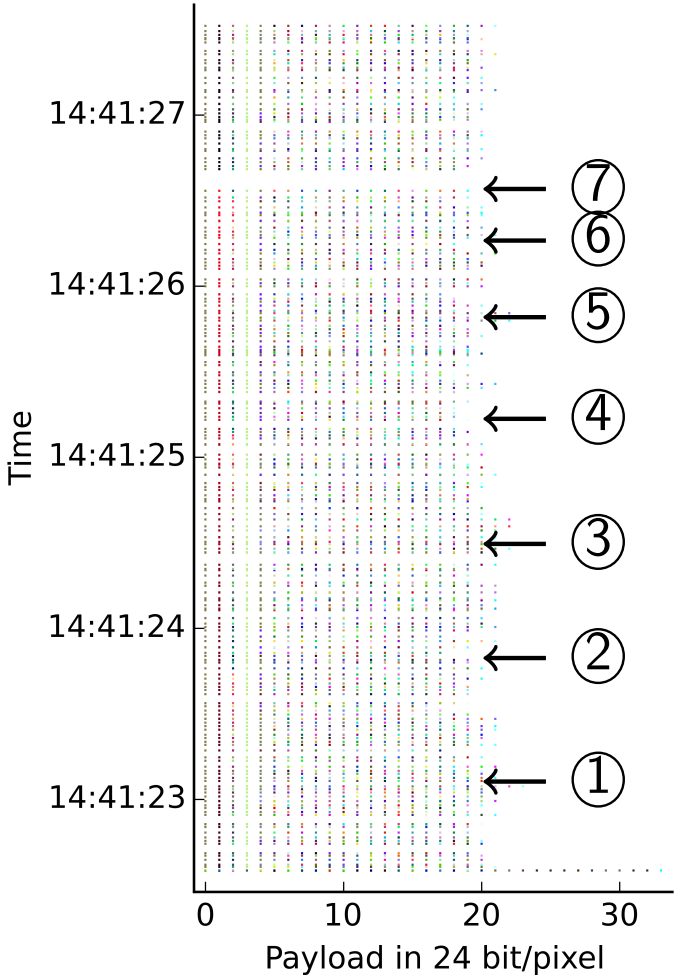



(1) RTCP packet, (2 & 3) re-occurring data – repeating color patterns, (4) packet header, (5) different packet lengths.

# M1: Visual Comparison of Altered and Unaltered Traffic



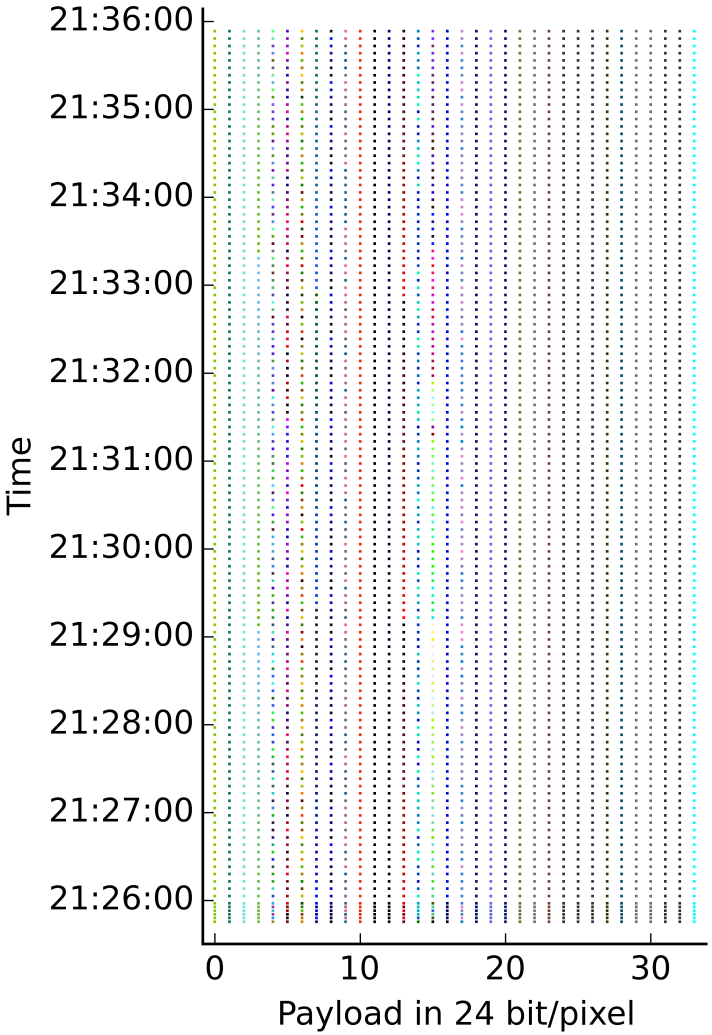

(a) Unmodified VoIP traffic



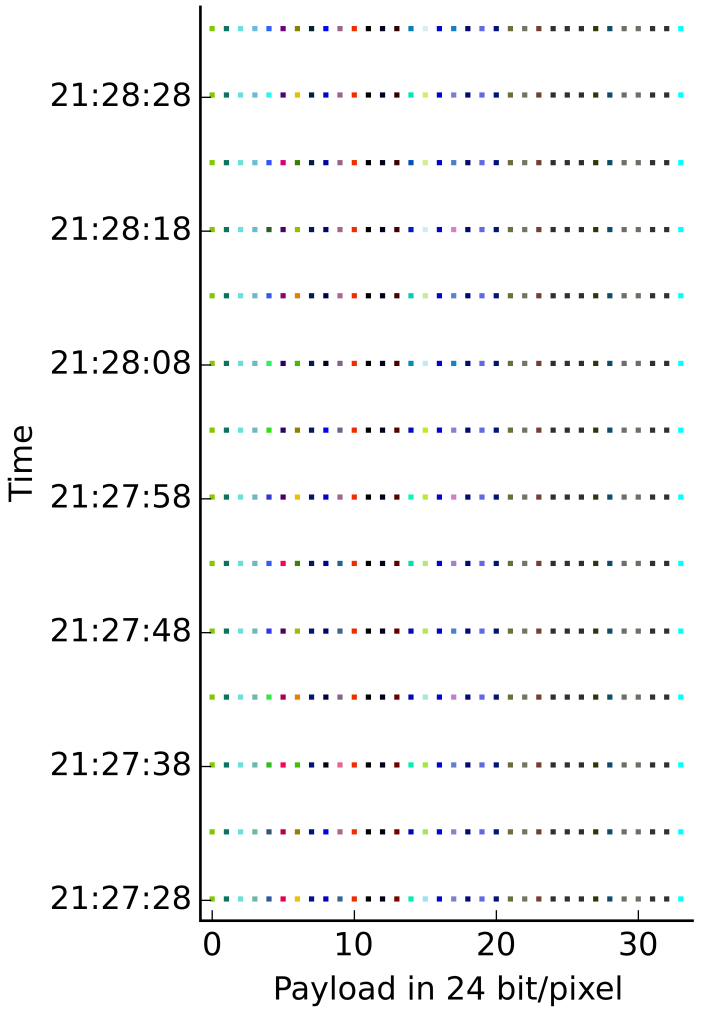

(b) Resulting VoIP traffic of M1

Images are stored in SVG format to allow easy application of image processing algorithms. / (1-7): unusual vertical spaces

# M2: Colored received RTCP messages

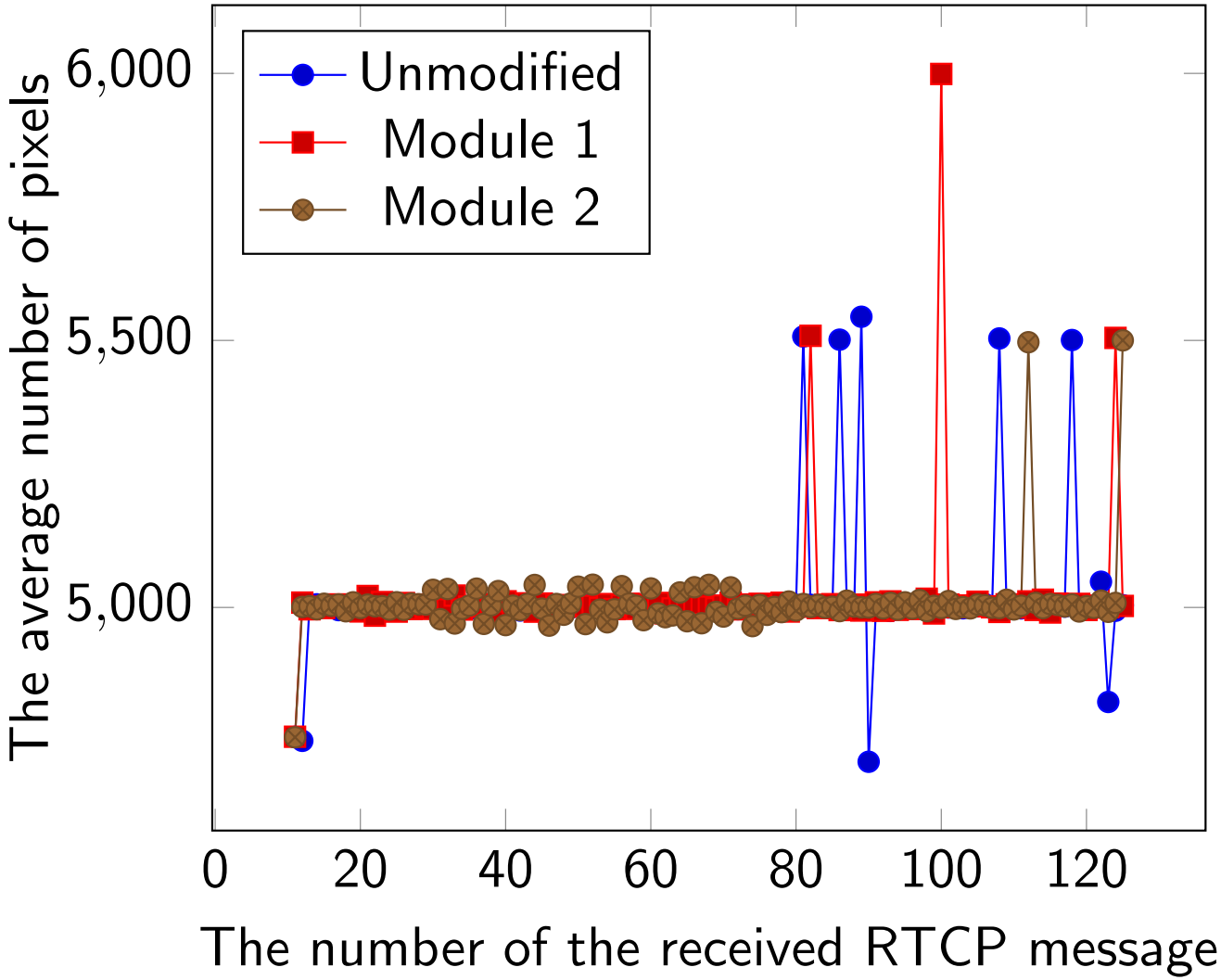



(a) Received RTCP messages from Module 2, no irregularity visible due to minimized delay



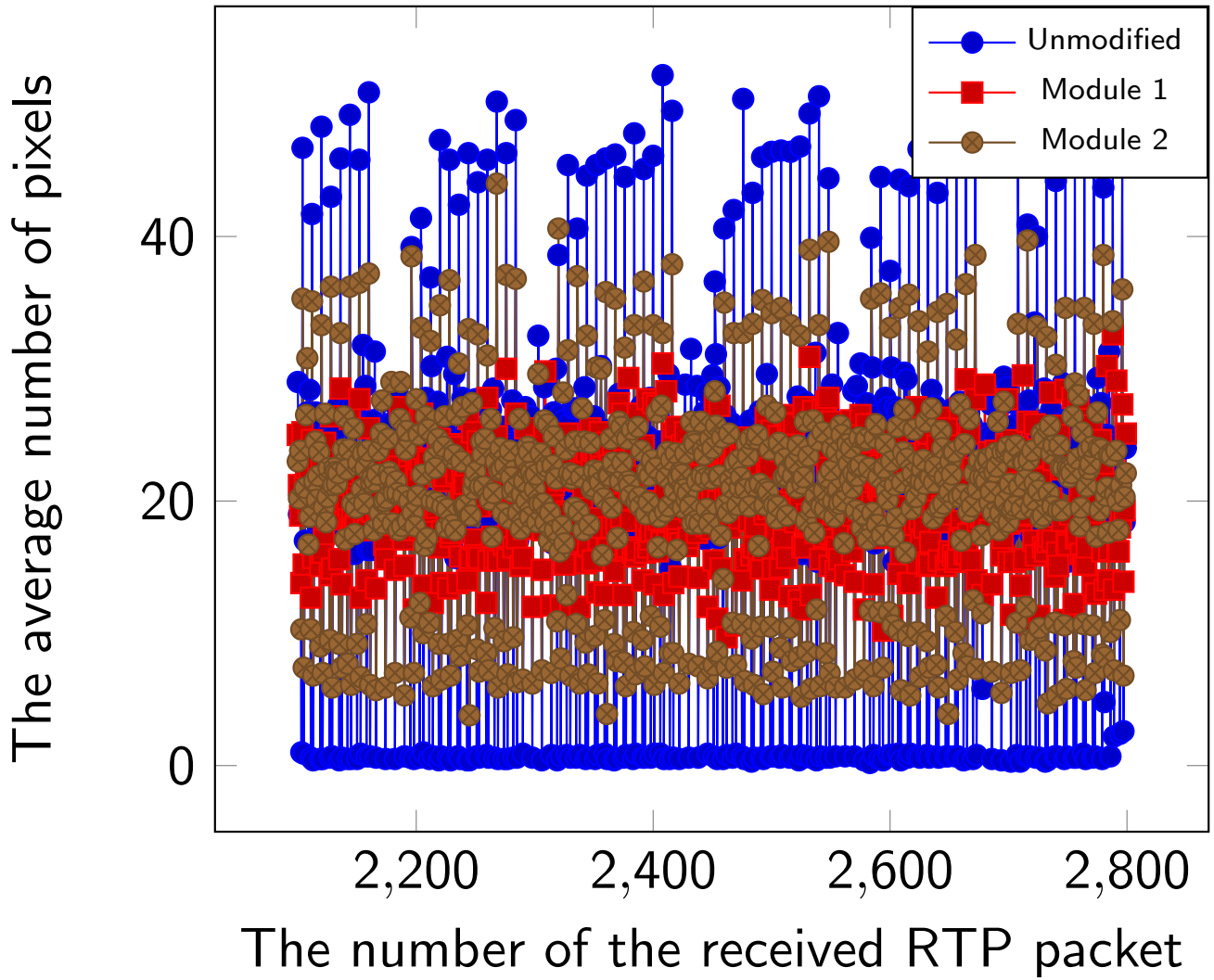

(b) Selected RTCP messages from Module 2: same situation despite higher resolution

# Colored pixels between two received RTCP messages




5,000px represent 1ms of traffic differences visible. Difference of M2 between 30-70th message considered detectable (larger than STDEV).

# Colored pixels between two received RTP messages




Influence of Inter-protocol Steganography visible.

# Initial Detection Approach

1. In each image: count number of colored RTP packets between two consecutive colored RTCP messages.
   - Colored RTP packets can be distinguished from colored RTCP messages by the length of the pixel row.
2. Identify start of potential steganographic transmission (based on above values).
3. Colored RTP packets and RTCP messages are evaluated separately:
   - For each generated image: determine time between two consecutive RTP packets.
   - This is done for RTCP messages accordingly.
4. A resulting value is classified as 'steganographic' if it varies from the expected one by more than the calculated standard deviation (built from the average of ten unmodified VoIP sessions).

# Initial Detection Results

| | **RTCP messages** | | **RTP packets** | |
|---|---|---|---|---|
| | Module 1 | Module 2 | Module 1 | Module 2 |
| TP | 175 (14.58%) | 278 (23.17%) | 90123 (32.82%) | 51847 (18.87%) |
| FP | 645 (53.75%) | 162 (13.50%) | 106677 (38.85%) | 53763 (19.58%) |
| TN | 269 (22.42%) | 595 (49.58%) | 45972 (16.74%) | 103281 (37.61%) |
| FN | 111 (9.25%) | 165 (13.75%) | 31838 (11.59%) | 65749 (23.94%) |
| Sensitivity | 61% | 63% | **74%** | 44% |
| Specificity | 29% | **79%** | 30% | 66% |
| FPR | 71% | 21% | 70% | 34% |
| FNR | 39% | 37% | 26% | 56% |

A high **sensitivity** corresponds to a high probability of detection and a high **specificity** corresponds to a high probability of absence of modifications.
However, both approaches need further investigation to provide convincing results.

# Summary

**Summarizing Characteristics of M1 & M2:**

**Steganographic bandwidth:** M1: 3 bits/RTCP message (which should not exeed 5% of the messages), M2: 1 bit/RTCP message. In our tests: 0.6 bit/s (M1) and 0.2 bit/s (M2).

# Summary

**Summarizing Characteristics of M1 & M2:**

**Steganographic bandwidth:** M1: 3 bits/RTCP message (which should not exeed 5% of the messages), M2: 1 bit/RTCP message. In our tests: 0.6 bit/s (M1) and 0.2 bit/s (M2).
**Robustness:** M1 & M2 are not suitable for networks where delays and packet losses vary widely.

# Summary

**Summarizing Characteristics of M1 & M2:**

**Steganographic bandwidth:** M1: 3 bits/RTCP message (which should not exeed 5% of the messages), M2: 1 bit/RTCP message. In our tests: 0.6 bit/s (M1) and 0.2 bit/s (M2).
**Robustness:** M1 & M2 are not suitable for networks where delays and packet losses vary widely.
**Detectability:** VoIP sessions only slightly modified, which aids covertness. However, detectability not excessively tested and, as usual for such methods, depends on the extend steganography is embedded.

# Summary

**Summarizing Characteristics of M1 & M2:**

**Steganographic bandwidth:** M1: 3 bits/RTCP message (which should not exeed 5% of the messages), M2: 1 bit/RTCP message. In our tests: 0.6 bit/s (M1) and 0.2 bit/s (M2).
**Robustness:** M1 & M2 are not suitable for networks where delays and packet losses vary widely.
**Detectability:** VoIP sessions only slightly modified, which aids covertness. However, detectability not excessively tested and, as usual for such methods, depends on the extend steganography is embedded.
**Effect on VoIP session:** Minimal: M1 Fraction Lost manipulation can have an effect on VoIP quality, thus was chosen so small that it is not noticeable. M2 has no impact on the mouth-to-ear delay, as only RTCP messages are delayed.

# Future Work

**Experiment with other codecs** than Opus.

**Improve detection method** to achieve better detection results.

**Continue development of Traffic Coloring**, i.e. extend the visual analytics component (e.g. fine-grained scrolling and zooming to allow inter-action with the detection algorithm) and perform a user study.