TMP: Time Modulation Protocol

Dalia Papuc, Benjamin Lion, Hans-Dieter Hiep

October 27th, 2022

1 Introduction

In this article we describe a novel technique for *increasing the effective capacity* or *increasing the end-to-end privacy and security* of communication channels using high-resolution clocks[1]. Under specific conditions more data may be sent on channels than is traditionally the case, by means of advanced synchronization and packet delivery scheduling algorithms. This may alleviate the need for capital investment for extending bandwidth for Tier 2 or Tier 3 network operators, who are purchasing Internet Protocol (IP) transit to connect with remote networks [Win06]. Alternatively, this technique may be used by end-users of the network to increase the end-to-end confidentiality beyond classical encryption techniques, without the need for adapting the underlying transit networks.

So, let us jump ahead and give the protocol realizing our novel technique a name: Time Modulation Protocol, or TMP in short. The essence of TMP is this: instead of sending data (a sequence of bits) contained in packets over the Internet from one party to another, we instead ensure that the two parties have synchronized clocks and transmit data by merely sending a signal and reading off the clock value at the moment of reception of the signal. No longer data is present in the transmitted message (as a bit sequence), but instead is transmitted at the moment a signal is received: the value of the clock on the receiving side then is the value of the message. This ensures that the content of the communication can no longer be sniffed by a man-in-the-middle *if* such an attacker does not know the parameters of the clock synchronization between the parties, but it also ensures that metered links under-report the information content of the message.

The purpose of this paper is expository: to explain the main ideas behind our novel technique, we introduce a formal mathematical model of time-sensitive channels that we analyze theoretically, and relate it to existing concepts in the scientific literature (Section 3). We then sketch out a conceptual design of the protocol, including its architecture, deployment, and security considerations (Section 4). But, first we give motivation and explain the technique in more detail before diving into formal theory (Section 2).

[1]This technique was discovered during projects supported by the European Union's Next Generation Internet (NGI) initiative, as part of the Horizon 2020 Research and Innovation Programme, under grant agreements No. 825310, No. 871528, No. 957073.

2 Motivation and Potential

To whom is TMP potentially useful, why does it deserve attention, and what potential applications does TMP have?

To address the first issue, we consider the following two use-cases:

- network operators may benefit by increasing the effective capacity of channels: we shall discuss what is *effective capacity* (i.e. the end-to-end bitrate) and how it differs from *capacity* (i.e. the bitrate of metered/throttled channels) and we explain how and under what circumstances one may increase the effective capacity of a channel beyond its capacity;
- end-users may benefit from increased end-to-end security of channels: we shall discuss how our technique realizes a *covert channel* on top of any existing *overt channel* and under what conditions such a covert channel remains secret: when is it difficult to recover information for attackers, and how can this be used to increase the security of classical encryption.

Although it is important to have a solid theoretical understanding of the technique underlying TMP, we also want to progress in making the technique useful in practice. Thus, we sketch out how it could be realized, by proposing an architecture of a new and open Internet protocol. We recognize that TMP is most useful when it can be implemented by any software/hardware vendor in such a way that allows for interoperability, and this article is the first step towards accomplishing such goal.

To address the second issue, why TMP deserves attention, this article gives input to discussions concerning the questions:

- Why is there a tension between the quality of service of Internet service providers and precise timing requirements from end-users?
- Is it possible, in a general and easily implementable way, to circumvent any packet filtering and other censorship techniques on the transport layer?

Finally, since the ideas underlying the technique presented in this article are so simple, it is necessary to explain them using a simplified mathematical model and demonstrate prior art, to make sure the technique is in the public domain and cannot be patented.

What are the main ideas underlying our technique? The essence is that we manage the scheduling of sending signals and control timing of the reception of signals. After parties are synchronized (i.e. the internal clocks of the parties and the timing characteristics of the channel between them are predictable) it is possible to communicate by merely sending signals. The signals themselves do not directly contain information. To recover an information channel, the sending party first predicts the time in which a signal is expected to be received by the receiving party and schedules transmission until it can be guaranteed that the receiver's clock has the right value upon reception of the signal. Upon receiving the signal, the receiver looks at the clock value: the clock value encodes the data that has been transmitted (see Example 1).

Example 1. *Consider two parties, A and B, and both have a wall clock. A can communicate with B using a button that lights up a bulb on the side of B. Their clocks are synchronized (with a resolution of seconds) and the transmission delay of a signal is 2 seconds. Now A wishes to transmit a bit sequence to B. If A wishes to transmit a one, she waits until the second-hand on her clock points to an odd number and then press the button. If she wishes to transmit a zero, she waits until the second-hand points to an even number and press the button.*

For example, if A wishes to transmit 1011*, the following may happen:*

Clock value at A	*Button*	*Signal*	*Clock value at B*
57	*push*		57
58	*push*		58
59	*push*	*light*	59
0		*light*	0
1	*push*	*light*	1
2			2
3		*light*	3

At B, each time the bulb lights up, the evenness/oddness of the clock value determines the bit transmitted: an odd clock value represents a one, an even clock value represents a zero. Thus, B received the bit sequence 1011*.*

Clocks can be realized by hardware devices which provide a high-resolution clock, or by some global time signal such as that provided by a Global Positioning System (GPS). Two clocks are synchronized between two parties if both parties know that both clocks have the same value and tick at the same speed, and the timing characteristics of the channel between them is predictable. Interesting challenges are the difficult task of keeping clocks synchronized, for instance as caused by clock drift or by mobile processes, and what security assumptions are needed to ensure communication over the time domain remains secure. It seems there is a need for an analysis to know under what conditions, such as obliviousness of the prior synchronization of two parties, a man-in-the-middle is unable to uncover information transmitted through timing channels.

We now address the third issue, the potential applications of TMP.

It is not difficult to imagine more complicated modes of communications: between the two extremes of merely sending signals on the one hand, and sending messages in packets in the traditional sense on the other hand, the sender can also choose to send a parameterized number of bits through the time domain and let the remaining bits of the message present in the packet. The packet then contains partial information of the message, but it also acts as a signal. For an attacker to know the actual content of a packet, it is then required to know who are the involved parties, their precise clock values, clock resolution, how to encode/decode clock values to data, the overall transmission delay, and other timing characteristics. These variables are difficult to discover from within the network, since in principle any two parties on the Internet can be synchronized.

Alternatively, the technique also has the potential to be used to pass messages through highly censored networks that only allow specific traffic patterns to pass firewalls: the time domain is in principle unrestricted and can be used

for free communication. More specifically, if the packet is simply a signal and contains no data, or is a packet that can be hidden in usual Internet traffic, we can ensure to pass those strict firewall rules or content filters. Moreover, middleware that tries to prevent free communication through the time domain (e.g. by introducing unpredictable delays, jitter) also severely negatively affects other applications on the network.

Speculating even further, this technique may give fine-grained control over the trade-off between privacy and transmission speed: for example, by setting the ratio between the part of a message that is sent using the time domain (covert channel) and the part of the message that is sent in the usual way (overt channel). Using encryption on top of this may make it more difficult to uncover the information transmitted: using the time domain to transmit (a part of) an encrypted message causes an exponential increase (in the number of bits sent over the covert channel) in the difficulty of cracking the used cryptography method, since an attacker has to guess the value of the bits that are missing from the overt channel. Another potential benefit of this approach is that in metered channels or throttled channels, only the amount of transmitted bits is measured. But by using signals, we significantly reduce the *measured* bit count, causing the metered connection to under-report the actual amount of data, that is instead transmitted through the time domain. Depending on the timing quality of the underlying transit, the effective transmission speed can be acceptable for some applications, allowing for a (dynamic) trade-off in bandwidth costs versus transmission speed.

3 Formal Model

We now turn our attention to a more formal approach, to give understanding how the technique described above actually works. We first recognize the components involved in communication between two parties, and find a suitable mathematical abstraction to describe their behavior.

In his seminal work, Shannon introduced a communication system to reliably transmit information over a noisy channel [Sha48]. As depicted in Figure 1, a communication system comprises five main components (information source, transmitter, channel, receiver, destination). In Shannon's setting, the source and channel are given: the question is how to reliably transmit messages from the source to the destination, while the channel may be interfered by noise. Thus, the main question in his setting is how to design transmitters (that converts messages to sequences of bits) and receivers (that converts sequences of bits back to messages) that are reliable. This question crucially depends on the concept of *capacity*: the quality of the channel is determined by its capacity C (bits per second). The capacity of a channel is an upper bound on how much information can be reliably transmitted. Shannon's main theorem states that, for a noiseless channel, given a source of information with entropy H (bits per message), there exists a transmitter and a receiver that can transmit $C/H - \epsilon$ messages per second, for arbitrary small ϵ.

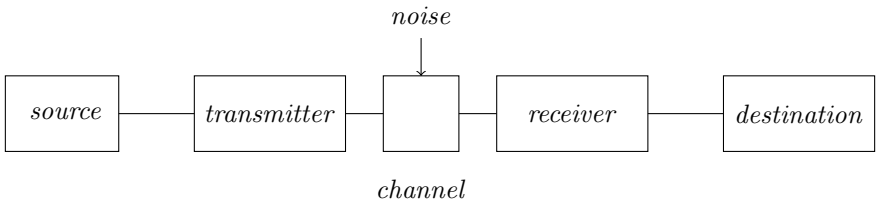

Figure 1: Diagram representing the different components involved during communication. The *source* is the information source that produces messages. The *transmitter* operates on the source and converts messages into a suitable form to be transmitted over the channel. The *channel* transports signals from the transmitter to the receiver, possibly interfered by *noise*. The *receiver* operates on the received signals and converts it back to messages, and delivers the messages to the *destination*.

That is, according to Shannon's theory, it is not possible to transmit more than C/H messages per second.

Example 2. *If the source has an entropy of* 1 *(bits per message), i.e. every message can be encoded as a single bit, then it seems obvious that one cannot transmit* more *than n messages on a channel with a capacity of n bits per second.*

Now, if a channel has a lot of noise, its capacity decreases: one may use a transmitter and receiver that perform error detection and correction, but the overhead required to compensate for the noise is the cause for a decrease in capacity. In this setting, the capacity of a channel indicates how fast one can reliably transmit information, but this notion of capacity is a static notion and time insensitive. That is, Shannon's theory says nothing about another aspect of quality: namely, what is the delay and the unexpected variation of delay (called jitter) of a channel. When comparing two channels, both having the same capacity, but one has lower delay and less jitter, we would like to recognize that one as having a higher quality than the other, with higher delay and more jitter.

Remark 1. *Often the quality of Internet connectivity is measured by capacity (bits per second), also called bandwidth. We later introduce the additional quality measure of jitter (and explain its unit: seconds).*

We investigate the same components as in Figure 1, but use a different mathematical model to explain the *time-sensitive* behavior of these components. The mathematical theory of communication we present accounts for this quality concern, by recognizing another source of noise, which we call *time noise*: in Section 3.1, we give a description of a class of time-sensitive transmitters and receivers, for which the meaning of a signal on the channel is dependent on time.

We show that such a sophisticated view of noise is necessary, by giving a construction of a noiseless channel that is seemingly able to transmit more

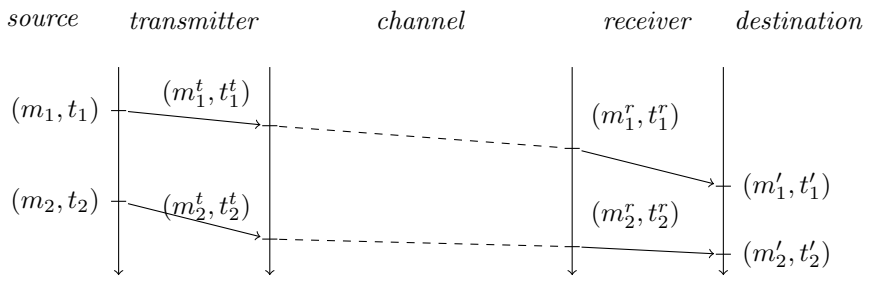

Figure 2: Diagram of a communication system using time-sensitive components.

messages than Shannon's theory predicts in Section 3.2, thus showing that the notion of capacity of a channel (bits per second) is no longer appropriate: although we still have the capacity of a channel (bits per second, being the same as signals per second), we also introduce a new concept, called *effective capacity*, which measures the bits per second that can be obtained by time dependent but synchronized transmitters and receivers. The transmitter and receiver perform encoding/decoding in bits per signal. By letting our signals have time dependent meaning, our analysis leads to interesting insights.

Finally, in Section 3.3, we show what prior art and related work exists, and why it is interesting in relation to our technique.

3.1 Components

Time and uncertainty There are different ways to mathematically conceptualize time, e.g. *real time* and *logical time*. In real time, we associate to events a real-valued timestamp: the time in which the event happened. In logical time, we abstract from the timestamp and we consider the order in which events happen. However, since logical time can always be reduced to real time, e.g. by assigning to each logical event some real-valued timestamp such that the order of events is consistent with the logical order of events. We shall thus consider real time.

In real time, for any two events that happen, there always exists a timestamp that lies precisely in the middle of the two events. Something similar can be found true for logical time: it is conceivable that for two events that happen in logical time, there could have happened a third event in between, such that the third happens after the first event and before the second event. As a consequence, theoretically, the amount of events that can fit a time interval is unbounded.

In practice, however, such modeling of time is unrealistic due to limited precision. No physical clock provides a real-valued timestamp: clocks are always bound by their precision. And even with logical time, it is inconceivable to be able to observe unbounded many events, so there must be a practical bound on how many events can be potentially interspersed between two observed events.

Now, stating that an event happens exactly at a time t would require us to measure with absolute precision the duration of t, which cannot practically be attained. Instead, the delayed time would range in an interval $[t-\epsilon; t+\epsilon]$ where ϵ becomes arbitrary close to 0 depending on the available precision.

To keep our theory simple, we do model events using real time: but we shall account for the imprecision of measuring time by introducing the concept of *time noise*.

An event is modeled as a pair of a *message* and a real-valued *timestamp*. The message is the information that is overtly present: for that, we use an arbitrary non-empty set D called the message domain. We assume the set D to be enumerable (that is, messages are digitally encodeable). It is possible that D contains only a single element: in that case an event can be considered just a timestamp. The timestamp of an event is the time at which the event is observed.

We also account for uncertainty in transmitting information through a channel: if a transmitter transmit a message m, but the receiver receives a potentially different message m' then *data noise* may cause the messages m and m' to differ. Similarly, if a transmitter transmits a message at time t, but the receiver receives that message at time t' then *time noise* may cause t and t' to differ. We thus consider events to represent whatever is observed at a component.

As a special case may an event have no significant message (if the message domain D contains only a single element) and we use the symbol $*$ to stand in the place of the message. We shall call such events *signals*. The only thing observed is the timestamp in which the signal became apparent. Note that within each time interval there are two possibilities: either some signal was present (an event was recognized within the time interval, with $*$ as message), or no signal was present.

Information source An information source produces digital messages, each at some time instant, to the transmitter. Formally, we model an information source as a subset of partial functions $\mathbb{R} \rightharpoonup D$, where D is the set of messages produced by the source. Each partial function $\mathbb{R} \rightharpoonup D$ is called a *stream of events* (with message domain D). Let $s : \mathbb{R} \rightharpoonup D$ be a stream of events, then $s(t) = d$ means that an event with message d is present at time t, which we also write $(d, t) \in s$. Messages are always generated at some time instant, and $s(t)$ is undefined when no message is produced by the source at time t.

For instance, Figure 2 displays a source that emits a stream of events in which we see message m_1 at time t_1, and message m_2 at time t_2.

Remark 2. *A stochastic source that would have some probability distributions over the set of messages could also be seen as a temporal source, where the distribution of message over time corresponds to the probability of the message from the stochastic source.*

Transmitter A transmitter operates on a stream of events and schedules every event for transmission over the channel. More formally, a transmitter is

an operator $\mathcal{T}$ that maps the stream of events $\mathbb{R} \rightharpoonup D$ to another stream of events $\mathbb{R} \rightharpoonup D'$. The transmitter changes the domain of messages: typically it reduces the message domain D by choosing D' to be $\{*\}$ (to produce a stream of signals, i.e. events containing only $*$ and a timestamp). Lifting to sets, given an information source $S \subseteq \mathbb{R} \rightharpoonup D$, the result of applying the operator to it, $\mathcal{T}(S)$, is again a set of streams of events, i.e. $\mathcal{T}(S) \subseteq \mathbb{R} \rightharpoonup D'$.

The transmitter of Figure 2 scheduled message m_1 to be sent at time t_1^t with message m_1^t, and scheduled message m_2 to be sent at time t_2^t with message m_2^t: what is depicted are particular streams of events. Note that if $m_1^t = *$ and $m_2^t = *$ would be the case (and so D' is $\{*\}$), then the transmitter reduced the stream of events (from the information source) to a stream of signals (to be transmitted by the channel).

Remark 3. *Typically, the domain D is a set of bit sequences, that are operated on by the transmitter before being sent over the channel. Note that a transmitter can still encode the information of the source entirely within the time domain (i.e. choosing D' to be $\{*\}$), as shown later. An idealized transmitter is a function, but practically a transmitter also introduces internal time noise or jitter. Indeed, it is possible to have a class of transmitters $\mathcal{T}$ that get as parameter a random variable sampled from a distribution, and transmits the signal accordingly. Instead, and without loss of generality, we consider the channel as the only source of noise: any internal noise produced by transmitters are amortized and modeled as noise from the channel.*

Channel A channel carries messages from transmitter to receiver through some spatial and temporal medium. A channel is a relation $\mathcal{C}$ that relates streams of events to streams of events, i.e. $\mathcal{C} \subseteq (\mathbb{R} \rightharpoonup D') \times (\mathbb{R} \rightharpoonup D')$. A signal channel is a channel in which D' is $\{*\}$: it does not carry messages but merely signals.

Different classes of $\mathcal{C}$ would capture different kinds of noise on a channel. For instance, a (data and time) noiseless channel is the identity relation. In Figure 2, only a single pair of two streams that are related by the channel are shown (by the dashed lines).

Remark 4. *Same as for a transmitter, the input/output response of a channel may depend on some probabilistic variables. Channels of such type would take as parameter a noise variable, whose valuation may impact the time at which a signal is received on the other end of the channel. For example, a signal channel that introduces a constant delay is parameterized by a (constant) value that determines the delay for the signal to propagate through the channel. Other examples are channels that work on arbitrary message domains that introduce a delay as a function of the message it carries: allowing us to model channels in which 'short' messages are transmitted faster than 'long' messages.*

Receiver A receiver is modeled as an operator $\mathcal{R}$ that maps streams of events to streams of events, i.e. $\mathcal{R} : (\mathbb{R} \rightharpoonup D') \to (\mathbb{R} \rightharpoonup D)$. Again we lift receivers to

sets of streams of events, so that given some such set $R \subseteq (\mathbb{R} \rightharpoonup D')$, we have $\mathcal{R}(R) \subseteq (\mathbb{R} \rightharpoonup D)$.

Remark 5. *In general, a receiver acts as the inverse of the operation performed by a transmitter with respect to the message. However, the original timestamp of the source may no longer be recovered: typically, there is a delay involved in transmitting and receiving a message (caused internally by the transmitter and receiver). In the case of a noisy transmission, the receiver should be able to recover the message reliably.*

Destination A destination receives digital messages from the receiver, and is similar to the source: a set of partial functions $\mathbb{R} \rightharpoonup D$.

Model of communication Fix a message domain D and take a source $S \subseteq \mathbb{R} \rightharpoonup D$. Now, given a transmitter $\mathcal{T} : (\mathbb{R} \rightharpoonup D) \to (\mathbb{R} \rightharpoonup D')$, a channel $\mathcal{C} \subseteq (\mathbb{R} \rightharpoonup D') \times (\mathbb{R} \rightharpoonup D')$, and a receiver $\mathcal{R} : (\mathbb{R} \rightharpoonup D') \to (\mathbb{R} \rightharpoonup D)$, we model the destination as all those partial functions d such that there is some $\sigma : R \rightharpoonup D'$ such that $\mathcal{R}(\sigma) = d$ and $(\mathcal{T}(s), \sigma) \in \mathcal{C}$ for some $s \in S$. Here D' is the message domain of the channel, which typically contains only a single element to model signal channels.

We may thus think of a communication system being the components: source, transmitter, channel, and receiver. In such a system the destination then is fixed: and we can compare the source with the destination to determine the qualities of the communication system.

Reliability We can now speak of a *reliable* transmitter/receiver pair (relative to a given channel) whenever it is the case that for any stream of events s in the information source, and any stream of events obtained at the destination d, their projections to a sequence of messages are equal. That is: the order and the contents of the messages are preserved.

More strict notions of reliability can be defined by also considering the timestamps of the corresponding messages in the source and destination (e.g. their difference in time is limited), but also weaker notions of reliability can be defined by discarding information about the order of messages when comparing source and destination.

3.2 Capacity, effective capacity and jitter

In the seminal paper of Shannon, the capacity of a channel is measured in bits per second and is a fixed constant of a communication system. As displayed in Figure 3, a visual description of the capacity is that no more than n bits can be transmitted in 1 second between the transmitter and receiver.

Ten years after his seminal work, in [Sha58], Shannon considers the case of a channel with side information, and shows that the capacity of the transmission can exceed that of the channel capacity. Here we already find the roots of our technique, namely we distinguish between the *capacity* of an underlying channel,

and the *effective capacity* of a synchronized system employing such underlying channel. We exploit the fact that the transmitter and receiver are synchronized to realize a side channel.

Concretely, in the case of a time-sensitive transmitter and receiver, a bit sequence is no longer transmitted through the channel, but is enumerated on both the transmitter and the receiver side as time passes by. As shown in Figure 4, a single signal transmitted on the channel can reliably encode for one of the enumerated sequences if the transmitter and the receiver are synchronized. In such case, the effective capacity (bits per second) may exceed the capacity of the channel (bits per second).

We discuss the idealized setting in which a transmitter and receiver are specific functions of time, and relate the effective capacity to the property of the synchronization taking place between the two communicating parties.

Transmitter and receiver A time-sensitive transmitter takes a stream of events as input and encodes the content of the messages as a stream of signals. The time encoding is parameterized by a *precision*, δ, of a time reading, where $\delta = 0$ refers to second precision, $\delta = 3$ refers to millisecond precision, $\delta = 6$ refers to microsecond precision, etc. The corresponding *time resolution* for a given δ is $10^{-\delta}$: it is the smallest discriminatable unit in time.

We define the time approximation $\lceil t \rceil_\delta$ which maps the time t to the smallest time $k \cdot 10^{-\delta}$ such that $k \cdot 10^{-\delta} \leq t < (k+1) \cdot 10^{-\delta}$ with $k \in \mathbb{N}$.

Since the data domain D is enumerable, there must exist some $\phi_D : \mathbb{N} \to D$ to enumerate D (there may be multiple choices possible). A time-encoding transmitter $\mathcal{T}_\delta$ with precision δ is defined, for all streams of events s and for all events $(d, t) \in s$, by $\mathcal{T}(s)(t) = (*, t^{tr})$ where $t^{tr} = k \cdot 10^{-\delta}$ is the smallest time such that $d = \phi_D(k)$ and $\lceil t \rceil_\delta < t^{tr}$.

Remark 6. *For clarity, we fix a class of transmitters that use a time precision in the decimal base, but other bases could be used without changing our results.*

Remark 7. *The transmitter and receiver are idealized in the sense of not having internal noise and being able to schedule a message at a precise timing. Instead, the internal noise is modeled by a noisy channel. (See also Remark 3.)*

Example 3. *Consider a source whose messages are taken from a data domain $D = \{0,1\}^n$ containing binary sequences of size n. There is a natural enumerator $\phi_D : \mathbb{N} \to D$ that maps the natural number to the n last digits of its binary encoding. For instance, if $n = 4$, $\phi_D(2) = \phi_D(18) = 0010$ and $\phi_D(0) = \phi_D(16) = 0000$. A transmitter with precision δ and using such enumerator can schedule the same message every $16 \cdot 10^{-\delta}$ seconds.*

Remark 8. *We may consider different classes of enumerators, such as periodic or aperiodic enumerators. An enumerator is periodic if the whole enumeration of all data repeats. An aperiodic enumerator may change the order in which data is enumerated over time, such that if two data elements are enumerated twice, the data element that follows may be different. In case of periodic enumerators,*

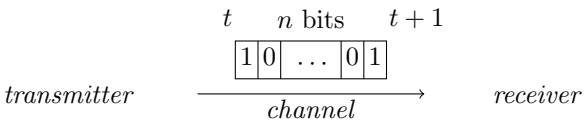

Figure 3: Sequence of n bits sent on a channel in 1 second.

the transmitter may schedule the same message at periodic time stamps. In case of aperiodic enumerators, such as enumerating natural numbers, the transmitter may schedule a signal for a one-time transmission. In such case, after the (unique) window of opportunity is missed, the same data can no longer be transmitted.

Remark 9. *In the case of a stochastic source, the occurrence of a message over time follows a distribution. Instead of enumerating every message once over a time interval, the repetition of a message at several time instants increases the likelihood of such message to be scheduled faster. Note that the precise order at which messages are enumerated does not change the probability, only the ratio of selecting the message faster.*

A time-decoding receiver $\mathcal{R}_\delta$ with time precision δ is defined, for all streams of signals $\sigma : \mathbb{R} \rightharpoonup \{*\}$ and for all occurrences of signal events $(*, t) \in \sigma$, by $\mathcal{R}(\sigma)(t) = (\lceil t \rceil_\delta, t)$.

Theorem 1. *For any precision δ, the transmitter $\mathcal{T}_\delta$ and the receiver $\mathcal{R}_\delta$ are reliable, i.e. inverse of each other with $\mathcal{R}_\delta \circ \mathcal{T}_\delta = id_{\mathbb{R} \rightharpoonup D}$ and $\mathcal{T}_\delta \circ \mathcal{R}_\delta = id_{\mathbb{R} \rightharpoonup \{*\}}$.*

Channel capacity The channel capacity as depicted in Figure 3 is the number of bits that can be sent within a second. Practically, the communication between the transmitter and the receiver is performed by sending signals that propagate through the channel. A bit takes two values, which also corresponds to the two states of an on/off signal: either there is a signal received or there is none. However, physical signals could carry more information (e.g. a photon has a phase, or an electrical signal has an amplitude), and therefore encode for more bits of information. For simplicity, we consider the simple encoding of an n bits message as a sequence of modulated pulses that takes the value 1 when the pulse is high, and 0 when the pulse is low.

The capacity of the channel is therefore giving an upper bound to the sampling frequency for a reliable transmission. We could recognize a pulse to be either the absence or the presence of a signal within a given time interval. Therefore, a channel with capacity n (bits per seconds) models that no more than n signals can be reliably transmitted within one second. If more signals could be reliably received within one second, a larger sequence of bits could then be reliably communicated and that would violate the channel capacity.

Remark 10. *There is a necessary synchronization between the transmitter and the receiver: the sampling rate at the receiver must be at the same frequency as*

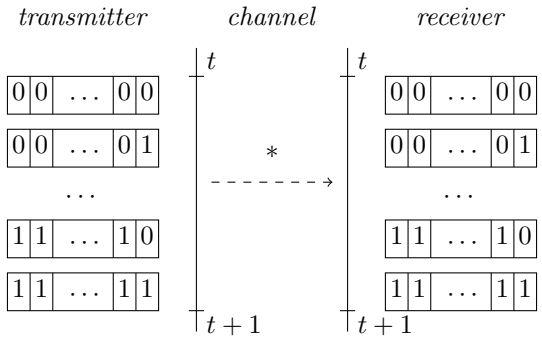

Figure 4: Alternatively, with a time-sensitive transmitter and receiver, the capacity of a channel is the number of signals that can be sent within a second, and the effective capacity of a transmission system is given by the number of bits that a transmitter and receiver can encode per signal.

the rate of emission of signals at the transmitter.

Effective capacity The capacity of a channel bounds the number of bits per second that can be reliably transmitted, but does not add any constraints on the time of arrival of the signals, which carries additional information. Assume now a pair of a time-sensitive transmitter and receiver as depicted in Figure 4. Both the transmitter and receiver enumerate, in lexicographic order and at the same speed, the set of bit sequences of size n. The channel is assumed to have a delay of 0 and a capacity of 1, which means that the transmitter cannot emit more than 1 signal within a second. In such setting, we observe that the amount of information transmitted by a *single* signal already exceeds the channel capacity.

Example 4. *Consider again Example 2: take a channel with capacity 1 (bits per second, thus signals per second). According to Shannon's theory, the number of messages that can be transmitted in one second is therefore at maximum* 1*. However, looking at Figure 4, we are able to transmit* n *messages (each message encoded by one bit) with only a single signal!*

Now take a channel of capacity of n, which means that the transmitter cannot emit more than n signals within a second. If the transmitter wants to communicate a digital message m of bit size n, then sending a single signal into the channel at the time of the enumeration is sufficient for the receiver to recover the message m, given that the channel has 0 delay. The effective capacity of the triple transmitter-channel-receiver exceeds the channel capacity, as n signals sent in 1 second reliably transmits n messages of size n (without giving information except that this is the case), $n-1$ signals sent in 1 second reliably transmits $n-1$ messages of size n (giving information about the message *not* transmitted), et cetera, all the way down to 1 signal sent in 1 second which

reliably transmits one message of size n (giving information about that message), and zero signals (without giving information except that this is the case).

Theorem 2. *For any channel capacity C, there exists a pair of a time-sensitive transmitter and receiver such that the effective capacity $C^{\mathit{eff}} > C$.*

Clearly, the effective capacity of a time-sensitive transmitter and receiver exceeds the capacity of the channel!

Remark 11. *Using a time encoding technique requires enumerating exponentially faster as the size of the set of messages increases. Thus, the limiting factor of employing this technique is the clock resolution that is available at the transmitter and receiver.*

Remark 12. *The assumption that the channel delay is* 0 *is unreasonable in practice. However, under a suitable synchronization protocol, if the delay between the transmitter and receiver is predictable, the transmitter and receiver can offset their enumeration such that the apparent delay is* 0*. The precision δ of the transmitter and receiver must be such that the delay of the transmission is predictable, and the enumerations on both side can be properly offset.*

By using time-sensitive transmitters and receivers, one can create a pair of time-sensitive transmitter and receiver with an effective capacity that tends to infinity, requiring clocks with a precision that tends to infinity. However, the conclusion that on any channel we could construct a channel with an unbounded effective capacity seems physically unfeasible (clocks with unbounded precision and digital circuitry that does not add any delay do not exists). Thus, this absurd situation shows that we need to model channels *with* time noise, and that time noise is a necessity in communication.

Channel with time noise As displayed in Figure 1, a noisy channel has an additional input from the environment that perturbs the content of the transmission. In the standard communication model, noises are classified with respect to the probability of altering a bit in the sequence that is sent, i.e. changing a 1 to a 0 or conversely, or altering the presence or absence of a signal. For a given class of noises, a transmitter and receiver can agree on an encoding and a decoding that enables recovery of possibly altered data.

In the case of a timed side channel, the content of a message is encoded as the time of arrival of a signal. The noise that affects the integrity of the communication is purely dependent on the variability of the channel delay. We call *jitter* the type of time noise in a channel, and introduce two equivalent definitions: one declarative and one operational.

The *jitter* is the smallest time resolution (seconds) that a transmitter and a receiver employs to communicate, such that the integrity of the messages is preserved. In other words, if the precision were made larger, the messages received at the destination would differ from the messages sent by the source. The jitter represents therefore the precision that maximize the effective capacity

of a channel while preserving the reliability of the communication. Formally, given a channel $\mathcal{C}$, the jitter is defined as being

$$\xi_1 = \min\{10^{-\delta} \mid \forall s \in \mathcal{S}.\ \mathsf{pr}(\mathcal{R}_\delta(\mathcal{C}(\mathcal{T}_\delta(s)))) = \mathsf{pr}(s)\}$$

where $\mathsf{pr}(s) : \mathbb{N} \to D$ is such that $\mathsf{pr}(s)(i)$ is the i-th message of the stream of events s. The definition above captures that, under the precision δ, all the messages from the source are successfully transmitted to the destination. The jitter is the smallest time resolution for such δ.

Alternatively, one can approximate the jitter on a channel as the observable variance resulting from a sequence of measurements. Consider a channel for which the transmitter sends to the receiver a message containing the time of emission. The receiver can therefore compare the received value with the time at reception and subtract to get the delay of the transmission. Repeating such experiments between the transmitter and the receiver leads to a series of measurable delays for which the variability of the measure can be inferred. In other words, the jitter can be approximated to be the smallest time resolution that approximates the delay Δ of a representative set $\mathcal{M}$ of experimental time pairs (t^{rec}, t^{tr}) where t^{rec} is the time measured at reception and t^{tr} the time measured at transmission:

$$\xi_2 = \min\{10^{-\delta} \mid \exists \Delta. \forall (t^{rec}, t^{tr}) \in \mathcal{M}. \frac{\lceil ((t^{rec} - \ t^{tr}) \cdot 10^{-\delta}) \rceil}{10^{-\delta}} = \Delta\}$$

Theorem 3. *The two definitions for the jitter are equivalent, i.e.* $\xi_1 = \xi_2$.

Remark 13. *We do not consider data loss for a point to point communication. In Section 4.3, we discuss reliable multi point communication with possible data losses.*

3.3 Prior art

In this section, we give an overview of related work we found in the scientific literature, to make a credible case that the technique we describe is already prior art. This increases the likelihood that the technique we describe is and cannot be patented, thus may be considered as part of the public domain. This overview is not complete, but intends to indicate that the technique underlying TMP is already applied in different fields.

In the seminal work of Shannon [Sha48], we find the definition of capacity of a discrete channel:

$$C = \lim_{T \to \infty} \frac{\log(N(T))}{T}$$

where $N(T)$ is the number of signals sent over a period of time T. Now, given that the entropy (bits per message) of a source S with distribution $(p_i)_{i \in S}$ is

$$H(S) = -\sum_{i \in S} p_i \log(p_i),$$

there exists an encoding of the source S such that transmission of messages can be done close to the rate C/H. According to this theory, it is not possible to transmit information at a higher rate. As we already seen in the previous section, this analysis is static and does not take into account synchronization between the transmitter and receiver to establish a side channel. The reason why is that in [Sha48], it is assumed that the time interval in which a signal can be sent has a fixed duration [Nyq24]. However, already in [Sha58], Shannon describes that the theoretical result no longer holds for channels with side information (i.e. the theory only holds in absence of what we now call timed side channels).

We are not the first in seeing limitations in the static analysis of Shannon. Also Dobrushin writes in [Dob61]: "One of the basic assumptions of the theory developed thus far was that the distribution of the input messages and the [transmitter] were regarded as given. However, this assumption is unreasonable in many real situations. Either because the statistical parameters of the channel *change rapidly with time* so that there is no way of obtaining the a priori distributions of these parameters, or because the same receiving-transmitting set must be adapted to operate under various conditions." (Emphasis is ours.) Dobrushin also identifies other problems with the theory of Shannon, but we shall not cover that here.

In the previous section we established the theoretical possibility of constructing a channel with unbounded effective capacity, but we are not the first in doing so. In [KS11], Khanna and Sudan investigate channels with unbounded capacity: "the potential reasons why its capacity (the number of bits it can transmit in a unit of time) might be unbounded [are] (1) (uncountably) infinitely many choices of signal strength at any given instant of time, and (2) (uncountably) infinitely many instances of time at which signals may be sent. However channel noise cancels out the potential unboundedness of the first aspect, leaving typical channels with only a finite capacity per instant of time. The latter source of infinity seems less extensively studied. A potential source of unreliability that might restrict the capacity also from the second aspect is 'delay': signals transmitted by the sender at a given point of time may not be received with a predictable delay at the receiving end." They examine this source of uncertainty by considering a discrete model of delay variance, and thus further investigate what we have previously called *time noise*.

In the context of communication networks, Gallager writes in [Gal76] that the arrival time and the transmission delay of packets on a packet-switched network can also carry information. In [AV96], Anantharam and Verdú consider a queuing model for communication where timestamps are used as well to transmit information. But also in [BGN17], the authors show their awareness of the significance of time, for example by writing that "interpacket delays of a packet stream, the reordering packets in a packet stream, or the resource access time of a cryptographic module" can be used to convey information.

From the physical point of view, there is scientific work that analyzes the thermodynamics and cost of time keeping [Pea+20]: there is a clear relation between the ticking of a (classical) clock and energy dissipated. In [Yam11], Yampolskiy writes that "theoretically silence based communication down to a

Planck time is possible. Such a form of communication is capable of transmitting a large amount of information in a very short amount of time, approximately 1043 bits/s. Because precision of time communication could be detected, time itself could be used as a measure of communication complexity valid up to a multiplicative constant with respect to a particular communication system." and that "the same idea could be implemented in a way which uses computation instead of relying on access to a shared clock. [...] This protocol is also subject to limitations inherent in the networking infrastructure and additional problems of synchronization."

From a security context, there is existing work related to the technique we described. In [KKK02], the authors present a novel key exchange protocol, where both parties eventually synchronize on a neural network, and use that neural network to generate a common time-dependent key: adversaries that knows the algorithm and observe the exchange of information are not able to recover the source and target keys. Covert timing channels can be constructed, as in [Sel+09]: "we design and implement a covert TCP/IP timing channel. We are able to quantify the achievable data rate (or leak rate) of such a covert channel. Moreover, we show that by sacrificing data rate, the traffic patterns of the covert timing channel can be made computationally indistinguishable from that of normal traffic, which makes detecting such communication virtually impossible." However, there are also applications to detect covert channels [BGC05]: "Covert timing channels use packet inter-arrival times, not header or payload embedded information, to encode covert messages. This paper investigates the channel capacity of Internet-based timing channels and proposes a methodology for detecting covert timing channels based on how close a source comes to achieving that channel capacity." Note, however, that the technique we described goes beyond inter-packet delay.

Another interesting aspect of the scientific literature is that of Kolmogorov complexity [Kol68]. 'Information' as defined by Kolmogorov and 'information' as defined by Shannon are related in interesting ways [GV03]. In [Kol68], the author proposes different approaches to the definition of information. The *combinatorial* description intuitively describes the entropy of a source as the size of the bit sequence necessary to transmit a message from that source. However, the *algorithmic* approach described by Kolmogorov tackles the following question: what is the quantity of information conveyed by an object x about an object y? In [Roo03], van Rooy emphasizes that Shannon capacity is about a *channel* of communication while Kolmogorov quantifies the information of a *word* to be transmitted, using the notion of algorithmic complexity. The definition of Kolmogorov complexity is based on the existence of a function ϕ where $\phi(p, x) = y$ evaluates a program p on the input x and returns a value y. Then, Kolmogorov complexity (of object y relative to x) is:

$$K(x \mid y) = \begin{cases} \min_{\phi(p,x)=y} l(p) & \\ \infty & \textit{if there is no } p \textit{ such that } \phi(p, x) = y \end{cases}$$

where $l(p)$ is some measure of the length of the program description p. It seems

an interesting direction to investigate how Kolmogorov complexity changes if $\phi(p, x)$ is time dependent (that is, the program p has access to a clock and the letters of the word x are associated to a timestamp).

4 Design of the Protocol

In this section, we sketch out the conceptual design of TMP, including its architecture, how to deploy it on current and future packet-switched networks, and security considerations.

The architecture we have in mind realizes a synchronization protocol that ensures the clocks of parties are synchronized (Clock Synchronization Protocol), and realizes time-sensitive encoding schemes for sending messages and retrieving messages from observed events (Token Exchange Protocol). Both protocols are combined and controlled to form the Time Modulation Protocol. This is discussed in more detail in Section 4.1.

To deploy our protocol, we envision different networking environments. First we consider the TMP protocol running on top of the Internet Protocol (IP), and discuss the downsides of a packet-switched network without much control over paths by peers. We motivate why the TMP protocol running on top of a candidate for Layer 3 replacement, the SCION protocol [Zha+11], is fruitful. This is discussed in more detail in Section 4.2.

Finally, we analyze the networking environment from an adversarial perspective and investigate what security guarantees can be given by employing the TMP protocol: namely, if an attacker cannot recover the parameters of synchronization, the information content communicated through the time domain is confidential. This is further discussed in Section 4.3.

4.1 Architecture

We first discuss two protocols, the Clock Synchronization Protocol, and the Token Exchange Protocol. Both protocols are combined and controlled to form the Time Modulation Protocol.

Synchronization is a procedure that leads to agreement by parties on the parameters needed for exchanging tokens: clock precision and offset, the enumerator (mapping clock values to messages), and hybrid mode (how much bits are present in the time domain versus the data domain of each token). In the Clock Synchronization Protocol, two parties exchange timing information for constructing a model that predicts the transmission delay. We can start out with a simplified model of a global clock (e.g. as obtained from a GPS signal, or from any other shared global clock) for which the transmission delay is some established constant and the jitter is a (minimized) variable, to find the best values of the other parameters. This setup allows one to experiment with changing the synchronization conditions such as caused by local clock drifts or dynamic changes in transmission delays. More complex models of the channel delay are imaginable and should be investigated later (but is out of scope of this article),

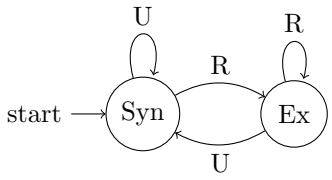

Figure 5: A simplified state machine describing the two modes of operation of TMP. The protocol starts in the Syn(chronize) state: it keeps in this state while the channel is U(nreliable). During synchronization, clock parameters are tweaked until it establishes a reliable channel. As soon as the channel becomes R(eliable), the protocol moves to the Ex(change) state: it may communicate messages as long as the channel is R(eliable). As soon as (incorrigible) errors are detected, the protocol moves back to the Syn(chronize) state.

such as delay dependent on packet content and networks providing congestion feedback.

Given that two parties are successfully synchronized, so we have obtained the parameters of the synchronization, we can choose a transmitter and receiver that will be employed during token exchange. The Token Exchange Protocol sends and receives tokens: tokens are either (on one extreme end) signals without any data contained in them but with significant timing information, or (on the other extreme end) messages in which timing information is insignificant. The Token Exchange Protocol must be designed in such a way to allow a dynamic trade off between these two extremes. This part of the protocol converts the reception of a token into a message. An important parameter is the precision, indicating what part of the clock value is significant. Increasing the precision may cause more interference by time noise in the underlying channel, while decreasing the precision causes more jitter since messages may be delayed for a longer time. The Token Exchange Protocol requires coding techniques to be applied to detect errors in transmission. Recoverable errors can be used to adapt the clock resolution dynamically, but unrecoverable errors can cause the channel to be disrupted temporarily and requires re-synchronization.

The Time Modulation Protocol combines the two protocols above, in an alternating fashion to ensure a reliable communication channel. It detects significant changes in transmission delays (leading to more errors during transmission) to trigger re-synchronization, see Figure 5.

Remark 14. *The choice of the enumerator may be used to implement some error detection and correction. For instance, by choosing a suitable mapping of data, or mapping some timestamps to errors. If a change in delay causes an error value to be detected on the receiver side, the time noise in the channel is detected.*

Error detection Due to the time sensitivity of TMP, and due to the volatility of time in, for instance packet-switched network, error detection and correction

are key considerations. We distinguish two categories of error detection:

- Active detection (on the side of the transmitter): transmit a (marked) message that is echoed by the receiver. Since the transmitter knows which message was originally sent, it can detect whether a different message was received. Active error detection can be used for discovering clock drift between the two parties, but also to discover different timing characteristics of the underlying channel. This process may be used for synchronization in the first place. Note that such active detection requires a symmetric channel (both parties are capable of transmitting and receiving messages).

- Passive detection: traditional error detection coding techniques can be applied over the data that is encoded in both the time domain and the data domain. Typically, errors are more likely in the time domain than in the data domain (since the link layer already provides error detection and correction capabilities for the data domain): thus the coding technique employed must favor detection of errors in the time domain.

Different strategies for error correction exist, and again we distinguish two categories for error correction:

- Active correction (on the side of the receiver): a message in which an error is detected at the receiver side can be requested from the transmitter to be sent again, in a similar way how TCP works. Alternatively, the detected error can be forwarded to the application and retransmission of the message can be managed by the application.

- Passive correction: traditional error correction coding techniques can be applied, where redundant information present in the time domain and data domain allows for recovering the original message that was transmitted. This technique is necessary in case of asymmetric communication. The overhead of adding redundant information causes a decrease in (effective) capacity. Having the ability to correct errors in this manner allows for a dynamic adjustment of the parameters of the token exchange, e.g. by letting the receiver signal the transmitter to use a different clock resolution.

4.2 Deployment on IP and SCION

In this section we describe how TMP can be deployed in existing packet-switched networks, such as those that support the Internet protocol or SCION [Zha+11].

To keep the design of the Clock Synchronization Protocol and the Token Exchange Protocol simple, it seems a reasonable choice to see TMP as an enveloping protocol, i.e. offering the same guarantees as the Internet protocol (or any other Layer 3 protocol). This means that error detection is relegated to higher-level transport protocols such as TCP or UDP. However, the detection of errors should be fed back to the protocol implementation of TMP, e.g. to trigger re-synchronization if a certain threshold of errors is reached.

Network Topology A directed path between two peers on a network corresponds to a channel between a transmitter and a receiver. A symmetric path between two peers is established if there exists a path in both directions between the two peers. TMP on IP should realize symmetric paths, since then active error detection and correction techniques can be employed.

Current practices in IP networks make the timing characteristics of the underlying channel non-deterministic and may introduce unpredictable jitter. Examples are: (1) network configuration changes transparently (to the peers) change route and thereby change the timing characteristics, (2) congestion cause packets to be dropped causing unreliable transmission of signals, (3) priority queues and unpredictable overtaking traffic cause sudden but temporary changes in timing characteristics. Having the ability for peers to chose the path for communication would increase reliability of channels between peers: having these capabilities seem a good argument to switch to alternative Layer 3 protocols such as SCION.

Remark 15. *Knowing the path between two nodes is sensitive for the security of the communication. If the two end points of the communication are aware of the topology and the nodes involved in their communication, both parties may fix a precision such that any additional node on the path could be detected, since it would result either in an increase of the delay or a change in the jitter. In that sense, time-sensitive components differentiate an intruder that spoofs communication by being directly on the path from an intruder connected to a node on the path, as in the former case, a delay is added.*

4.3 Security considerations

In [Sha49], Shannon considers the possibility of an attacker to observe signals exchanged between a sender and receiver. To protect against such scenario, the transmitter and receiver share a key with which messages are encrypted. The general framework is pictured in Figure 6.

The core of TMP is based on a synchronization between the transmitter and receiver. As detailed in Section 4.1, the synchronization involves multiple parameters: the delay Δ, the jitter ξ, the choice of an enumerator, which includes establishing the amount of data to be transmitted in the time domain, resp. data domain. The delay is a constant that is agreed by both parties, and consists of the fixed time value that the signal takes to reach the other end. The jitter ξ is variable and models the unpredictability of the channel.

One of the laws of modern cryptography, the Kerckhoffs principle, states that 'security should not rely on the secrecy of the cryptosystem itself'. This is currently the case with (a)symmetric encryption algorithms. The notion of security for a synchronization protocol is the inability for an attacker to recover the meaning of the signals, i.e. providing confidentiality, even though the synchronization protocol is publicly available. It may be the case that establishing synchronization requires a channel that provides confidentiality, authentication and integrity, as it is the case with exchanging keys in classical

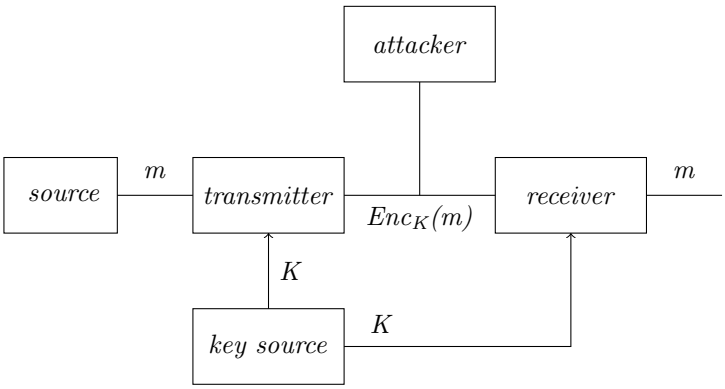

Figure 6: An attacker listens the encrypted communication $Enc_K(m)$ on a channel between a transmitter and receiver sharing the key K.

symmetric encryption.

As a first step, let us consider the following simple synchronization protocol:

- the transmitter and receiver exchange messages of the form $(\lceil t \rceil_\delta, t)$, for a δ initially fixed by the transmitter;
- at time of receiving, both parties compare their local time with the time $\lceil t \rceil_\delta$ for which the message that has been sent;
- after sufficient exchanges of messages, both the receiver and transmitter have a collection of pairs, for each message received, of its time of emission and time of reception. Considering the channel to be symmetric, the receiver and transmitter agree on the delay Δ, given by the subtraction of the timestamps, and samples of the jitter ξ given by the distance to the constant delay Δ. Note that if the time precision $10^{-\delta}$ is higher than the actual delay (i.e. $10^{-\delta} > \Delta$), then Δ will be computed to be 0 by both parties, as all durations lower than the time precision are identified to the same lower bound.

Furthermore, following the synchronization completion, we now consider a transmitter/receiver pair in which a message is encoded entirely in the time domain, and we let the choice of an enumerator open.

Remark 16. *List of questions:*

- *the encryption key of the communication seems to be (Δ, ξ), the parameter of the synchronization. When do the transmitter and receivers stop sending signals? What is a good approximation of the delay and the jitter?*
- *is it reasonable to assume channel to be symmetric from a time perspective?*
- *can the synchronization also change the precision δ to get as close as possible to the jitter (i.e. the uncertainty of the communication)?*

The follow up question is to consider whether the synchronization protocol is secure. For example, if an attacker can recover the encryption key (Δ, ξ) by observing the exchanges of messages and interacting with the receiver and transmitter by following the same synchronization protocol.

Remark 17. *A transmitter encrypts message m (plaintext) using the parameters of the synchronization protocol and it obtains a time t (ciphertext), at which the signal $*$ is scheduled for transmission. On the receiving side, a receiver decrypts time t'. It applies the inverse operation of the transmitter as mentioned in Section 3.1, to obtain the corresponding message m.*

We remark that the same security notions used in cryptography may be employed to assess the security of an instance of the synchronization protocol, such as security against distinguisher under chosen plaintext attacks (CPA), and chosen plaintext/ciphertext attacks (CPCA).

Note that the synchronization protocol differentiates an *internal* attacker that is on the path between the transmitter and receiver, and an *external* attacker that listens to messages forwarded from a peer on the path.

Theorem 4. *If an external attacker synchronizes with a transmitter and a receiver, the attacker cannot recover the two parameters of the synchronization between the transmitter and the receiver.*

Proof. Knowing (Δ_1, ξ_1) and (Δ_2, ξ_2), the two synchronization parameters between the attacker and the transmitter and receiver respectively, are not sufficient to deduce (Δ, ξ), the parameters from the transmitter and receiver, as $\Delta_1 + \Delta_2 = \Delta + x$ where x is the delay between the channel and the attacker. □

Through the simple example given above, we motivate the study and design of synchronization protocols that provide confidentiality, ensuring that only the sender and receiver are able to understand the meaning of a signal. Additional aspects that may be considered for security are: the proportion of information to send through the time domain and data domain, e.g. the message $m = 1011$ has 2^4 choices of bits to be sent in one of the two domains, and this choice may introduce a few bits to the 'key length'. This choice seems similar to the choice of an enumerator, e.g. for a given data domain D we could opt for a pseudorandom enumeration where the seed is shared by the transmitter and receiver and kept hidden.

Other important properties in providing a private and secure data transfer are integrity and authentication. Devising techniques to achieve the two are exciting research directions. Integrity ensures the content of the message is not altered. In TMP, this corresponds to a delay such that the signal is mapped back to the original message by the receiver (time domain only), and in hybrid mode of operation also ensuring the data is not modified (time and data domain). Authentication ensures the communicating party is the intended party. While there exist techniques to achieve these properties in the data domain, e.g. message authentication codes, we are wondering how time can influence the

field. An open question is: could synchronized clocks be exploited to guarantee security properties such as authentication?

In contrast to existing (unencrypted) transport protocols, collecting TMP signals without knowledge of the synchronization parameters, only reveals the timestamp at the observer, while generating signals corresponds to random data generation or an undefined value. To make the collection of TMP signals insignificant, consider the following protocol: let each communicating party send signals at random intervals to random peers where only one such signal to an intended peer carries significant data (an agreement between the communicating parties is indeed required). Now it is difficult for an attacker to guess which parties the sender is synchronized with: the redundant data complicates the analysis.

Finally we wish to address the question of how could one prevent two parties from successfully communicating via TMP. Due to (intentional or unintentional) errors in the network, messages may be (i) dropped, (ii) delayed, or (iii) altered. These challenges are also present in current packed-switched network communication environments.

In the case of multi point communication, the TMP protocol can suffer data loss: one peer drops the signal before transmission. Similarly to connection based protocols, an acknowledgment-based mechanism can be used on top of TMP to ensure retransmission and reliability of the communication. For instance, the Transmission Control Protocol (TCP) uses retransmission whenever a segment is lost, corrupted, or overly delayed [KR21].

In contrast to delaying a TCP segment, delaying the TMP signal may actually change the meaning of the intended message. We distinguish two kinds of delays: predictable and unpredictable. In the case of a predictable delay, the transmitter and receiver can agree on an offset that allows recovery of the message. Therefore, the communication is analog to the case of no delay. In the case of an unpredictable delay, the precision used by the transmitter and receiver can be renegotiated to account for the jitter. When this occurs often enough, it may lead to resynchronization between the two communicating parties, forcing them to decrease the precision of their clocks, which is undesirable. Sometimes, keeping the same precision may still be acceptable given suitable retransmission mechanism. (See also the discussion in the previous section.)

Last, a malicious node could prevent two parties from communicating by taking over their communication session. In a hijacking attack, a malicious node waits for a request and then races against the communicating party to produce a reply. An example is having a node trigger the execution of the re-synchronization protocol between two communicating parties and race with one of the legitimate parties to complete the protocol and consequently take over the connection. We depict this in Figure 7. The design of synchronization protocols must consider such situations and devise protective mechanisms to guard against hijacking.

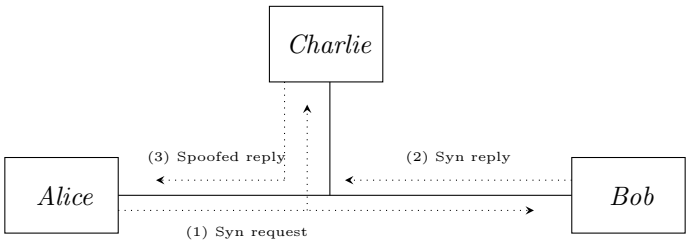

Figure 7: An example of TMP Hijacking. The idea is for Charlie to race against Bob when producing a TMP reply, for instance during re-synchronization. The intent of Charlie is to take over the session between Alice and Bob.

5 Call to action

We proposed the Time Modulation Protocol as an adapter to existing transport protocols, which transfer a message purely in the data domain. Now, with TMP, you could also opt for sending a message partly through the time domain and partly through the data domain, or purely through the time domain. TMP has two main benefits:

- TMP increases the effective capacity of a channel: a signal carries additional data in the time domain. We envision TMP as part of the new generation of Internet protocols.
- TMP provides confidentiality by design: without knowledge of the synchronization parameters between two communicating parties, an observer cannot understand the meaning of a signal.

If you believe TMP is a valuable communication protocol and would like to see it implemented, we encourage you to join us in developing it as an Internet standard, or by contributing (independent) research & development.

Here are a number of slogans:

> Want to communicate to the outside world even if your government imposes censorship on communication? Starts using TMP now!

> Want to transmit more data on a reduced bandwidth link? Consider TMP!

> What do you want? A 10Mbit/s channel with 5 second jitter, or a 50Kbit/s channel with 1 millisecond jitter? Figure it out by using TMP!

References

[Nyq24] H. Nyquist. "Certain Factors Affecting Telegraph Speed". In: *Bell System Technical Journal* 3.2 (1924), pp. 324–346. DOI: 10.1002/j.1538-7305.1924.tb01361.x.

[Sha48] Claude E. Shannon. "A mathematical theory of communication". In: *Bell Syst. Tech. J.* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[Sha49] Claude E. Shannon. "Communication theory of secrecy systems". In: *Bell Syst. Tech. J.* 28.4 (1949), pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[Sha58] Claude E. Shannon. "Channels with Side Information at the Transmitter". In: *IBM J. Res. Dev.* 2.4 (1958), pp. 289–293. DOI: 10.1147/rd.24.0289.

[Dob61] R.L. Dobrushin. "Mathematical problems in the Shannon theory of optimal coding of information". In: *Proc. 4th Berkeley Symp. Mathematics, Statistics, and Probability*. Vol. 1. 1961, pp. 211–252.

[Kol68] Andrei N. Kolmogorov. "Three approaches to the quantitative definition of information". In: *International Journal of Computer Mathematics* 2 (1968), pp. 157–168.

[Gal76] R. Gallager. "Basic limits on protocol information in data communication networks". In: *IEEE Transactions on Information Theory* 22.4 (1976), pp. 385–398. DOI: 10.1109/TIT.1976.1055588.

[AV96] Venkat Anantharam and Sergio Verdú. "Bits through queues". In: *IEEE Trans. Inf. Theory* 42.1 (1996), pp. 4–18. DOI: 10.1109/18.481773.

[KKK02] Ido Kanter, Wolfgang Kinzel, and E. Kanter. "Secure exchange of information by synchronization of neural networks". In: *EPL* 57 (2002), pp. 141–147.

[GV03] Peter Grünwald and Paul M. B. Vitányi. "Kolmogorov Complexity and Information Theory. With an Interpretation in Terms of Questions and Answers". In: *J. Log. Lang. Inf.* 12.4 (2003), pp. 497–529. DOI: 10.1023/A:1025011119492.

[Roo03] Robert van Rooy. "Quality and Quantity of Information Exchange". In: *J. Log. Lang. Inf.* 12.4 (2003), pp. 423–451. DOI: 10.1023/A:1025054901745.

[BGC05] Vincent H. Berk, Annarita Giani, and George V. Cybenko. "Detection of Covert Channel Encoding in Network Packet Delays". In: *Computer Science Technical Report TR2005-536*. Dartmouth Digital Commons, 2005.

[Win06] Mark Winther. "Tier 1 ISPs: What they are and why they are important". In: *IDC White Paper* (2006), pp. 1–13. URL: https://www.gin.ntt.net/wp-content/uploads/2020/01/IDC_Tier1_ISPs.pdf.

[Sel+09] S. H. Sellke et al. "TCP/IP Timing Channels: Theory to Implementation". In: *IEEE INFOCOM 2009*. 2009, pp. 2204–2212. DOI: 10.1109/INFCOM.2009.5062145.

[KS11] Sanjeev Khanna and Madhu Sudan. "Delays and the Capacity of Continuous-time Channels". In: *CoRR* abs/1105.3425 (2011). URL: http://arxiv.org/abs/1105.3425.

[Yam11] Roman V. Yampolskiy. "Efficiency Theory: a Unifying Theory for Information, Computation and Intelligence". In: *CoRR* abs/1112.2127 (2011). URL: http://arxiv.org/abs/1112.2127.

[Zha+11] Xin Zhang et al. "SCION: Scalability, control, and isolation on next-generation networks". In: *2011 IEEE Symposium on Security and Privacy*. IEEE. 2011, pp. 212–227.

[BGN17] Arnab Kumar Biswas, Dipak Ghosal, and Shishir Nagaraja. "A Survey of Timing Channels and Countermeasures". In: *ACM Comput. Surv.* 50.1 (2017), 6:1–6:39. DOI: 10.1145/3023872.

[Pea+20] A. N. Pearson et al. "Measuring the Thermodynamic Cost of Timekeeping". In: *arXiv: Mesoscale and Nanoscale Physics* (2020).

[KR21] James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach*. 8th. Pearson, 2021. ISBN: 9780135928615.