

CYRANO protocol:

Ethernet fencing protocol for communication between computers and piste apparatuses

Version 1.0, December 05, 2008

1. Introduction

This protocol aims at describing the communication between software managing fencing competitions (shortly named Software below), and piste apparatus (shortly names Apparatus below) using the Ethernet IP protocol. Communication occurs in two directions: Software sends matches to devices; devices send scores, stopwatch, lights, priority, etc., to Software.

The technical name of the protocol is **EFP1**. The use name is **CYRANO**.

History

In April 2008, Favero contacted Nicaud telling him that he intends to create a new Apparatus which will display the fencer names and nations and will communicate with Software with Ethernet IP messages if a protocol can be defined for that. Nicaud said "yes" and sent a first version of a protocol. In May, Favero and Nicaud elaborated this protocol with more details and decided to make it public and also to discuss with FIE and other manufacturers in order to eventually improve the protocol. Discussions occurred in May and June and result in this document.

The discussion group was composed of:

- FIE (Dos Santos <jecdossantos@gmail.com>, Ioan Pop <pop.fie@gmail.com>, Nathalie Rodriguez. <nr@fie.ch>),
- Engarde (Jean-Francois Nicaud <Jean-Francois.Nicaud@imag.fr>, PALFI Georges <georges.palfi@free.fr>)
- Favero (Gino Favero <orders@favero.com>)
- Allstar (Frank Messemer <f.messemer@t-online.de>)
- EscrimeTec (Frédéric Marciano <EscrimeTec@aol.com>)
- Spanish manufacturer (Diaz <fer@cib.csic.es>, Miguel Ojeda <m.ojeda@telefonica.net>)
- Swiss Timing (Cuenin, Thierry <Thierry.Cuenin@swisstiming.com>)
- Kabcom (KABCOM <kabcom@wanadoo.fr>)
- Ophart (German software Derkum <s.lange-derkum@fechten.org>)
- Tesis (Italian software Tedeschi <tesis@tesis.org>)

2. Sort of IP protocol

The UDP protocol is used: it is light, it sends information without establishing connection and without reception acknowledge. When acknowledge is required, it is done inside the messages.

See <http://tools.ietf.org/html/rfc768>

In French: http://fr.wikipedia.org/wiki/User_Datagram_Protocol

In Italian: http://it.wikipedia.org/wiki/User_Datagram_Protocol

In English: http://en.wikipedia.org/wiki/User_Datagram_Protocol

in Deutsch: http://de.wikipedia.org/wiki/User_Datagram_Protocol

The IP addresses used are not dynamic so that Software can have a table with all the IP addresses and the piste identifiers of a competition.

The port number used is 50100. It is in the range of Dynamic and/or Private Ports which is from 49152 through 65535 (see: <http://www.iana.org/assignments/port-numbers>).

The idea is to always use it or to use it except for special reasons. Software will allow changing the port number. It is suggested that Apparatuses also allow changing the port number.

3. Content of messages

In individual competitions, a **match** occurs between two individuals and may have several **periods** with a 1 minute pause between two periods.

In team competitions, a **match** occurs between two teams and is composed of **rounds** between individuals.

The period number and the round number are generally displayed by the Apparatuses using the same area.

HELLO message

The HELLO message tells to the Apparatus that Software is online; it contains only the compelling fields (see below). Software sends an HELLO message to each piste each 15 seconds.

NEXT message

The NEXT message sent by the Apparatus, asks to Software the data for next round/match.

PREV message

The PREV message sent by the Apparatus, asks to Software the data of previous round/match.

DISP message

The DISP message sent by Software providing a match/round to the Apparatus and the INFO message sent by the Apparatus providing information to Software contain the identification of the match and round. There is a fencer identifier for left and right fencers. Software uses this identifier and not the fencer name because the name can change at any time (correction of a spelling error - change of the parameter for producing names for devices). Software has to take care of the homonymous names. It is worth to use the Olympics convention for 10 characters names with first letter of first name possibly parameterized to allow larger names (10, 12, 15, etc.).

If the referee changes the right/left position of the fencers on the piste and the Apparatus, the Apparatus exchanges the values of the right and left fencers (Id, name, nat, etc.) in order to correspond to the piste. Software use the fencer Id to give the correct score, card, etc. to the correct fencer.

INFO message

The Apparatus sends an INFO message each time there is a change in its state. The content of this message can be limited to the compelling fields and the changed elements. In addition, the Apparatus sends a full INFO message each time it received an HELLO message (see below).

Here is a list of the possible INFO message:

- partial INFO message: it contains all the compelling fields and only the changed fields;
- full INFO message: it contains all the compelling fields and the ones that can change on the Apparatus;
- incomplete INFO message: it contains only the known data (Protocol, Com, Piste, Compe, score, stopwatch, cards, status...) and the unknown fields are empty or have default values.

Note that the INFO message (finished) has the field 14 (state) to the value 'E' (**Ending**).

ACK message

The ACK message sent by Software, tells to the Apparatus that the end of match/round has been approved.

NAK message

The NAK message sent by Software, tells to the Apparatus that the end of match/round has not been approved.

(**) means "compelling field" for all messages.

(*) means "compelling field" for DISP and INFO messages except incomplete INFO when waiting without match

| # | Fieldname | Description, remarks | Values, explanations, remarks | Examples | Max length |
|-----|---------------|--|---|---|------------|
| 1 | Protocol (**) | Name of the protocol | EFP1 for Ethernet fencing protocol number 1 | EFP1 | 4 |
| 2 | Com (**) | Name of the command | HELLO: tells that Software is online (Software → Apparatus) DISP: to display the match (Software → Apparatus) ACK: acknowledges reception of the end of a match or round (Software → Apparatus) NAK: indicates the reception of an incorrect end of a match (Software → Apparatus) INFO: score, stopwatch, light, etc. (Apparatus → Software) NEXT: asks Software to send the next round or the next match (Apparatus → Software) PREV: asks Software to send the previous round or the previous match (Apparatus → Software) | HELLO DISP ACK NAK INFO NEXT PREV | 6 |
| 3 | Piste (**) | Piste identifier | Can be a number, "podium", "rouge", "vert", etc. | 17 | 8 |
| 4 | Compe (**) | Competition identifier | Allows several competitions | efj-eq | 8 |
| 5 | Phase (*) | Phase number | First round of poules is 1 when it exists; second round of poules is 2 when it exists; tableau is "last round of poules number" plus 1 | 1 | 2 |
| 6 | PoulTab (*) | Identifier of the poule or the tableau | Number of the poule or identifier of the tableau | 3 A32 | 8 |
| 7 | Match (*) | Match number | Number of the match inside the poule or the tableau | 12 | 3 |
| 8 | Round (*) | Round number | Number of the round for team competitions (1-9), of the period for individual competitions (1-3) | 6 | 2 |
| 9 | Time | Time to start the match | The time to start the match sent by Software can be displayed by Apparatuses | 13:15 | 5 |
| 10 | Stopwatch | Value of the stopwatch | In INFO messages, Apparatuses put here the value of the stopwatch. In DISP messages, Software generally makes this field empty. It can put here a value of the stopwatch when going back to a match ended by error. | 3:00 | 5 |
| 11 | Type | Type of competition | I: individual T: team | I | 1 |
| 12 | Weapon | | F: fleuret E: épée S: sabre <i>Can be used by Apparatuses to adapt to the weapon</i> | S | 1 |
| 13 | Priority | Priority | N: none R: right L: left | N | 1 |
| 14 | State | State of the Apparatus | F: Fencing H: Halt P: Pause W: Waiting E: Ending | H | 1 |
| 15 | RefId | Referee identifier | Can be ignored by Apparatuses | 132 | 8 |
| 16 | RefName | Referee name | Can be ignored by Apparatuses | J. Smith | 20 |
| 17 | RefNat | Referee nation | Can be ignored by Apparatuses | GBR | 3 |
| R1 | RightId (*) | Right fencer Id | | 28 | 8 |
| R2 | RightName | Right fencer name | | P. Martin | 20 |
| R3 | RightNat | Right fencer nation | | FRA | 3 |
| R4 | Rscore | Right fencer score | | 8 | 2 |
| R5 | Rstatus | Right fencer status | U: Undefined V: Victoire D: Défaite A: Abandonment E: Exclusion | V | 1 |
| R6 | RYcard | Right fencer yellow card | Number of yellow cards for right fencer (0-1) | 0 | 1 |
| R7 | Rrcard | Right fencer red card | Number of red cards for right fencer (0-15) | 1 | 2 |
| R8 | Rlight | Right fencer light | 0: Right light off 1: Right light on | 1 | 1 |
| R9 | RWlight | Right fencer white light | 0: Right white light off 1: Right white light on | 0 | 1 |
| R10 | RMedical | Right fencer medical interventions | Number of medical interventions on right fencer | 0 | 1 |
| R11 | RReserve | Right reserve fencer introduction | N: none R: introduce the reserve for current round | N | 1 |

| | | | | | |
|-----|------------|-----------------------------------|--|-----------|-----|
| L1 | LeftId (*) | Left fencer Id | | 32 | 8 |
| L2 | LeftName | Left fencer name | | B. Panini | 20 |
| L3 | LeftNat | Left fencer nation | | ITA | 3 |
| L4 | Lscore | Left fencer score | | 6 | 2 |
| L5 | Lstatus | Left fencer status | U: Undefined V: Victoire D: Défaite A: Abandonment E: Exclusion | D | 1 |
| L6 | LYcard | Left fencer yellow card | Number of yellow cards for Left fencer (0-1) | 0 | 1 |
| L7 | LRcard | Left fencer red card | Number of red cards for Left fencer (0-9) | 1 | 1 |
| L8 | Llight | Left fencer light | 0: Left light off 1: Left light on | 0 | 1 |
| L9 | LWlight | Left fencer white light | 0: Left white light off 1: Left white light on | 0 | 1 |
| L10 | RMedical | Left fencer medical interventions | Number of medical interventions on left fencer | 0 | 1 |
| L11 | LReserve | Left reserve fencer introduction | N: none R: introduce the reserve for current round | N | 1 |
| | | | | | 167 |

4. Behaviour of Software and Apparatuses

An Apparatus has several states:

- 1) **Fencing**: fencers are fencing, the stopwatch is running;
- 2) **Halt**: between “halt” and “aller”, the stopwatch is stopped;
- 3) **Pause**: between two periods, the stopwatch is running;
- 4) **Ending**: between the end of a match (or round for Team competitions) and the acknowledge receipt;
- 5) **Waiting**: a match/round is finished (still displayed on the Apparatus) or a match/round has not yet started (already displayed on the Apparatus) or there is no match/round. The stopwatch may indicate the starting time of the match. The scores may be blank.

Let us call **Active** the union of the four first states (all except waiting).

When the Apparatus receives an HELLO message, it sends a full INFO message to Software if it is in **Active** state; if it is in waiting state, it sends an incomplete INFO message with the piste number and the **Waiting** state and other available information (maybe none or the complete information if it has already received a DISP message).

The Apparatuses have a yellow light indicating when the Software connection is on (it turns on when it receives an HELLO message - it turns off after 40s without receiving an HELLO message).

The Apparatuses have a red light indicating when the Apparatus is active.

The Apparatuses have the following buttons: “**BEGIN**”, “**END**”, “**NEXT**” and “**PREVIOUS**”. “**END**” can be named “**END + final approval**” in order to precise its action.

Behaviour of the Apparatus

| State of the Apparatus | Event | Behaviour of the Apparatus | New state of the Apparatus |
|------------------------|------------------------------------|---------------------------------------|----------------------------|
| Active | Change on the Apparatus | sends an INFO (not finished) message | Active |
| Active | Reception of HELLO message | sends a full INFO message | Active |
| Active | Hit on the “ END ” button | sends an INFO (finished) message | Active (Ending) |
| Active | Reception of ACK message | changes state → sends an INFO message | Waiting |
| Active | Reception of DISP message | ignore | Active |
| Waiting | Reception of HELLO message | sends an INFO message | Waiting |
| Waiting | Reception of DISP message | Install the match/round | Waiting |
| Waiting | Hit on the “ BEGIN ” button | changes state → sends an INFO message | Active (Halt) |
| Waiting | Reception of ACK message | ignore | Waiting |

Hitting the “**NEXT**” and “**PREVIOUS**” button does nothing when the state of the Apparatus is active (the Apparatus sends nothing to Software).

In individual competitions, the “**END**” button indicates the end of the match. The situation is correct for the end of the match if:

- 1) The status of both fencers is normal and
 - a. the scores are different
 - b. or the scores are equal and there is a priority
- 2) or the status of one fencer or both fencers is abandonment or exclusion.

The situation of two fencers with **normal status, equal scores and no priority** is an incorrect end of match.

In team competitions, the “**END**” button indicates the end of the round and, in the case of the last round, the end of the match. The end of the match is correct / incorrect like in individual competitions. The end of a round which is not the last one is always correct.

Software takes care of fencers’ status only when the state is **Ending**.

If Software receives an incorrect end of match, it does not send an ACK message but sends an NAK “not acknowledge” message, so the apparatus can show to the referee a message as for example “END not accepted” or “Incorrect END”.

The matches/rounds sent by Software may have scores to be displayed. This is the case for team rounds, except the first one. This is also the case when the Apparatus asks for the previous match/round. The rule is: when it receives a DISP message, the Apparatus displays the score it contains and the “**BEGIN**” button must not reset the scores.

Detailed behaviour of the buttons

- **“NEXT”** button. It works only if the Apparatus is on the **Waiting** status. It requires Software to send to the Apparatus the next match/round information. The referee will see the new fencer names etc. on the LCD display but the Apparatus continues to be on the **Waiting** state.
- **“PREVIOUS”** button. It works only if the Apparatus is on the **Waiting** status. It requires Software to send to the Apparatus the previous match/round information. It is useful to modify a mistake. To modify a value it is necessary to push also the **“BEGIN”** button, make the modifications, and at the end push **“END + final approval”**.
- **“BEGIN”** button. It works only if the Apparatus is on the **Waiting** status. The **“BEGIN”** button will put the Apparatus on the **Active** status: it means that it will send continuously to Software all the changes that happen on the match/round. A red light on the apparatus will signal when it is on the active status.
- **“END + final approval”** button. It works only if the Apparatus is on the **Active** status. It is better to use this long name to emphasize that it is sent the conclusive results. The Apparatus will go to the **Waiting** status only if it will receive an ACK message.

It is suggested that each one of the 4 above buttons is accepted only if it is pushed together to an **“ALT”** button, to avoid wrong button actions. Anyway, any manufacturer can decide if want or not to include a security system.

Behaviour for the reserve fencer

The value ‘R’ (RESERVE) in the “R11” or “L11” field, sent by Apparatus, is accepted from Software only for Team competitions, when there is a reserve fencer and the reserve fencer has not yet been used. It can be cancelled during the round it has been set; subsequently to the insertion of the reserve fencer, Software will insert the reserve fencer (id, Name, Nation) in the following rounds.

5. Breakdown, lost messages and errors

UDP runs without reception acknowledge. Reception acknowledge messages (ACK and NAK) have been inserted for a particular situation.

Lost messages are solved this way

- 1) In the case of the lost of an HELLO or an INFO (not finished) message, another message of the same kind will be sent later.
- 2) In the case of the lost of an INFO (finished) message, Software will not send the ACK or NAK message. The active red light will not go off, the piste assistant will hit again the **“END”** button.
- 3) In the case of the lost of an ACK or NAK message, the active light will not go off, the piste assistant will hit again the **“END”** button.
- 4) In the case of the lost of a NEXT, a PREV or a DISP message, the new match/round will not be displayed on the Apparatus, the piste assistant will hit again the **“NEXT”** or **“PREVIOUS”** button.

Errors

- 1) The piste assistant hits the **“END”** button in an inadequate situation (two fencers with normal status, equal scores and no priority). The Apparatus receives the NAK message and displays “Incorrect END”.
- 2) The piste assistant hits the **“END”** button in a formally adequate situation but the match/round is not finished (there is still fencing to do). The piste assistant hits the **“PREVIOUS”** button to ask Software to put the previous match/round to unfinished and to send it again to the Apparatus (DISP message).
- 3) The piste assistant hits the **“END”** button with a wrong score. The piste assistant uses the **“PREVIOUS”** button to correct the score or asks the Software manager to correct the score with Software.

6. Syntax of the content

A two levels CSV syntax is used. There are 3 mains areas separated by “%”, each one having several values separated by “|”.

The first area is the general area. The second is the right fencer area. The third is the left fencer area.

Values are given with their natural length (no filling with blanks or zeros to reach a particular length).

Example:

```
|EFP1|DISP|8|fm-eq|2|B64| ... |%|28|P. Martin|FRA||U|...|%|32| B. Panini|ITA||U|...|%|
```

An empty value is given by two consecutive “|”.

If from a position of an area to the end of this area all the values are empty, they can be omitted.

