The Hermes Standard
for "M-to-M" in SMT Assembly

The Hermes Standard Change Proposal

CheckAlive Response

Voting meeting:
23th of April 2018 (NEPCON / Shanghai)

Requesting company:
ERSA GmbH

Version change:

Minor

Service description tag:

FeatureCheckAliveResponse

Description:

The connections are kept open all the time. As TCP by itself does not detect connection losses ("half-open connections" caused by e.g. process-/computer crash, unplugged network cables, …) both sides of a connection have to send cyclic CheckAlive messages. As not all TCP stacks recognize correctly the loss of connection when sending ~~telegrams~~ messages it is possible to extend the implementation of this functionality to an exchange of CheckAlive messages.

Use cases:

Especially for PLC that do not recognize "half-open connections", e.g. unplugged network cables.

Functionality / communication sequences:

One of the machines (in the figure the downstream machine but it could be also the upstream machine) send a (ping) CheckAlive message, that is a CheckAlive message with the attribute Type set to 1. The peer machine then responds immediately with a (pong) CheckAlive message, that is a CheckAlive message with the attribute Type set to 2 and the Id matching the Id of the (ping) CheckAlive message.
A missing response indicates a connection loss.

New / changed XML messages:

Extend the CheckAlive message with two optional attributes:

CheckAlive	Type	Range	Optional	Description
Type	int	1..2	yes	Ping / Pong message type
Id	string	any string	yes	Idenfier of the message

Proposed changes to standard:

2 Technical concept

…

2.3 Connecting, handshake and detection of connection loss

After booting, the downstream machine starts cyclic connection attempts to the configured upstream machines. When a connection is established, the downstream machine starts sending a ServiceDescription message whereupon the upstream machine answers with its own ServiceDescription. This ServiceDescription message contains the lane ID of the sending machine related to this TCP connection. It also contains a list of features which are implemented by the client. The features of the Hermes specification 1.0 have to be supported by any implementation and shall not be included explicitly.

If a downstream machine is already connected to the lane, this connection will be retained. A Notification message shall be sent to the new connection before it is closed.

After exchanging the handshake messages, both machines may begin to send BoardAvailable/ MachineReady messages (see section 2.4).

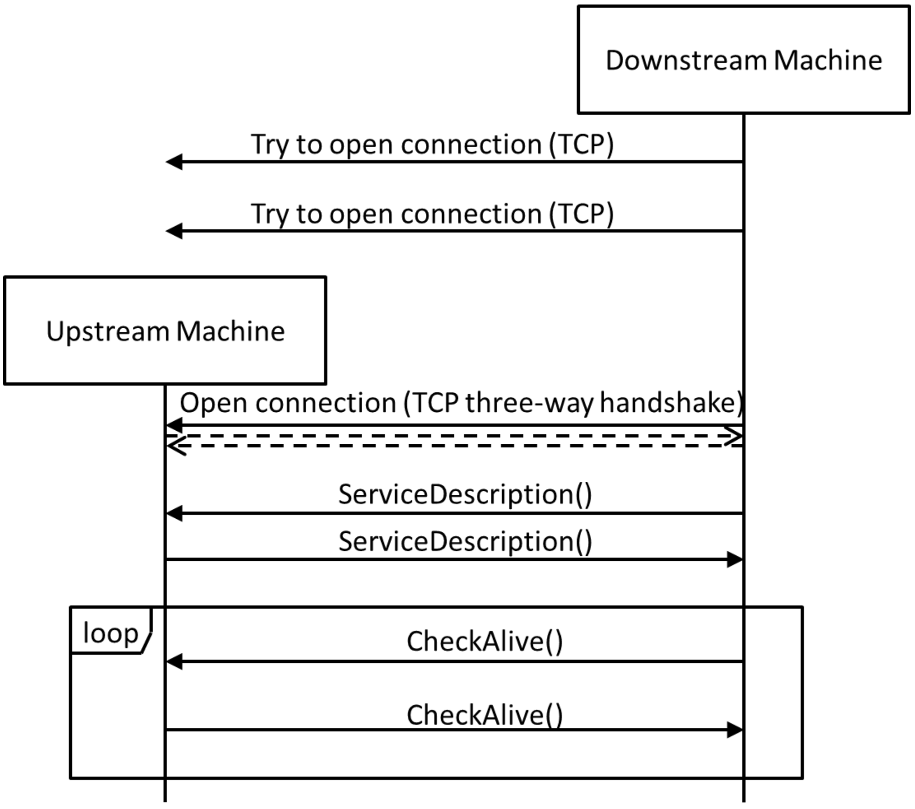

Fig. 2 Connection, handshake and connection loss detection

The connections are kept open all the time. As TCP by itself does not detect connection losses ("Half-open connections" caused by e.g. process-/computer crash, unplugged network cables …) both sides of a connection have to send cyclic CheckAlive messages. Those messages do not have to be answered by the

remote side – the TCP stack will detect a connection loss when trying to send the packet. If the server detects a connection loss, it ~~cleans up~~ ends the connection and waits for a new connection by the client. If the client detects a connection loss, it ~~cleans up~~ ends the connection and re-starts with the cyclic connection attempts. As not all TCP stacks recognize correctly the loss of connection when sending messages it is possible to extend the implementation of this functionality to an exchange of CheckAlive messages. Machines which have implemented this function do have the tag FeatureCheckAliveResponse in the ServiceDescription.
The exchange of CheckAlive messages then works like shown in Fig. 3.

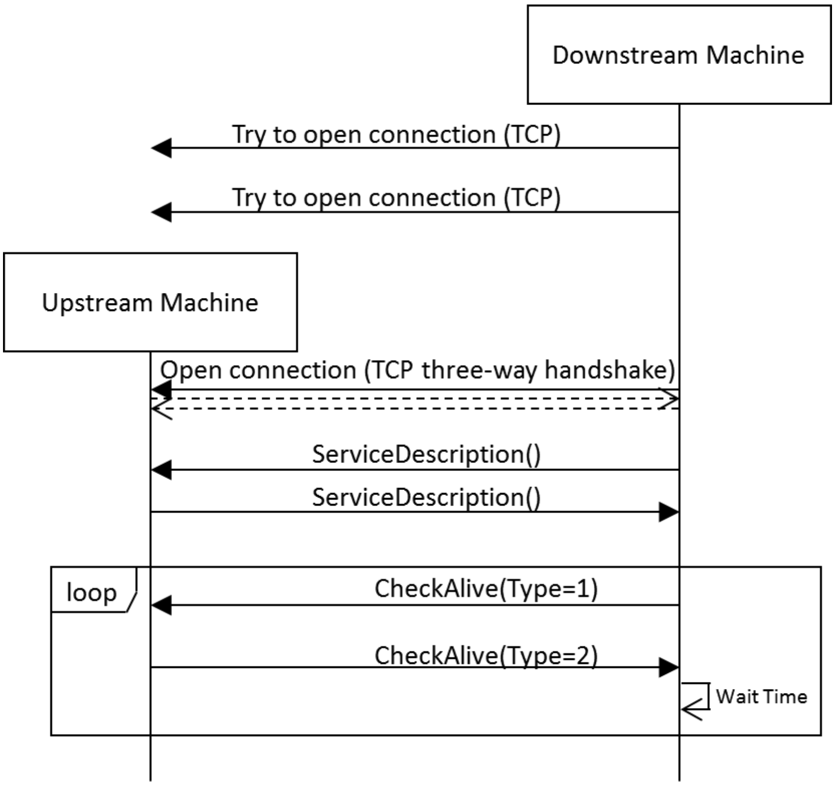

Fig. 3 Example for connection loss detection with FeatureCheckAliveResponse

One of the machines (in the figure the downstream machine but it could be also the upstream machine) send a (ping) CheckAlive message, that is a CheckAlive message with the attribute Type set to 1. The peer machine then responds immediately with a (pong) CheckAlive message, that is a CheckAlive message with the attribute Type set to 2 and the Id matching the Id of the (ping) CheckAlive message.
A missing response indicates a connection loss.

3 Message definition

…

3.3 CheckAlive

The CheckAlive message is used to detect connection losses. It therefore does not have to transport data and can be ignored by the receiver. Accordingly there is no response.

If a machine supports the FeatureCheckAliveResponse it must answer CheckAlive messages with Type set to 1 with a CheckAlive message with Type set to 2 and the same Id as the received CheckAlive message.

CheckAlive	Type	Range	Optional	Description
Type	int	1..2	yes	Ping / Pong message type
Id	string	any string	yes	Idenfier of the message

Type may be one of the following values:

1. Ping: CheckAlive request
2. Pong:CheckAlive response

The machine sending CheckAlive message with Type set to 1 chooses a unique for Id (e.g. GUID or time stamp). The machine responding with CheckAlive message with Type set to 2 has to answer using the same Id.

3.4 ServiceDescription

The ServiceDescription message is sent by both machines after a connection is established. The downstream machine sends its ServiceDescription first whereupon the upstream machine answers by sending its own ServiceDescription.

ServiceDescription	Type	Range	Optional	Description
MachineId	string	any string	no	ID/name of the sending machine for identifying it in a Hermes enabled production line.
LaneId	int	1 .. n	no	The sending machine's lane of this connection relates to Lanes are enumerated looking downstream from right to left beginning with 1
Version	string	xxx.yyy	no	The implemented interface version of the machine
SupportedFeatures	Feature[]		no	List of supported features (empty for version 1.0)

The features specified in version 1.0 of this protocol have to be provided by any implementation and thus are not listed in the SupportedFeatures list of the ServiceDescription explicitly.

Feature	Type	Range	Optional	Description
FeatureCheckAliveResponse	FeatureCheckAliveResponse		yes	Indication of CheckAliveResponse function implementation