Functional Specification for BlueTruth OutStation

SSL/4012/B/10.02.2015

Web: *www.simulation-systems.co.uk*
Email: *web@simulation-systems.co.uk*

Telephone: (01934) 838803
Fax: (01934) 876202

Simulation Systems Ltd
Unit 12 Market Industrial Estate
Yatton
Bristol, BS49 4RF
England

Contents

References

[1] BS EN ISO 9001:2008: Quality Management Systems - Requirements.

[2] TickIt Guide to Software Quality Management System Construction and Certification.

[3] SSL3753 – Functional Specification for the BlueTruth InStations v1.0

[4] SSL3820 – Interface Specification for BlueTruth OutStation v1.5

[5] Bluetooth Specification Version 4.0

Glossary: List of Standard Abbreviations

Term	Description
GPRS	General Packet Radio Service
NTP	Network Time Protocol
SSH	Secure Shell

1 Introduction

Bluetooth technology is embedded in billions of electrical devices world wide. Many devices (e.g. mobile phones, tablet computers, laptop computers etc) are carried by vehicle drivers or are embedded within a vehicle's system. Each Bluetooth device emits a unique signature that can be harvested and time stamped by an SSL OutStation. These signatures can then be made available to an InStation.

Analysing the collected signature information on an InStation enables the creation of origin and destination information for each detected Bluetooth signature including the time taken (i.e. journey times) between OutStations.

The purpose of this document is to describe an OutStation in terms of its capabilities. The target audience is internal to SSL.

2 System Overview

2.1 Main Components

The System consists of the following components:

- OutStation; located roadside or wherever the user wishes to obtain the Bluetooth ID's from.
- InStation; there can be multiple occurrences of the InStation desktop applications as well as a Web Portal for data visualisation.

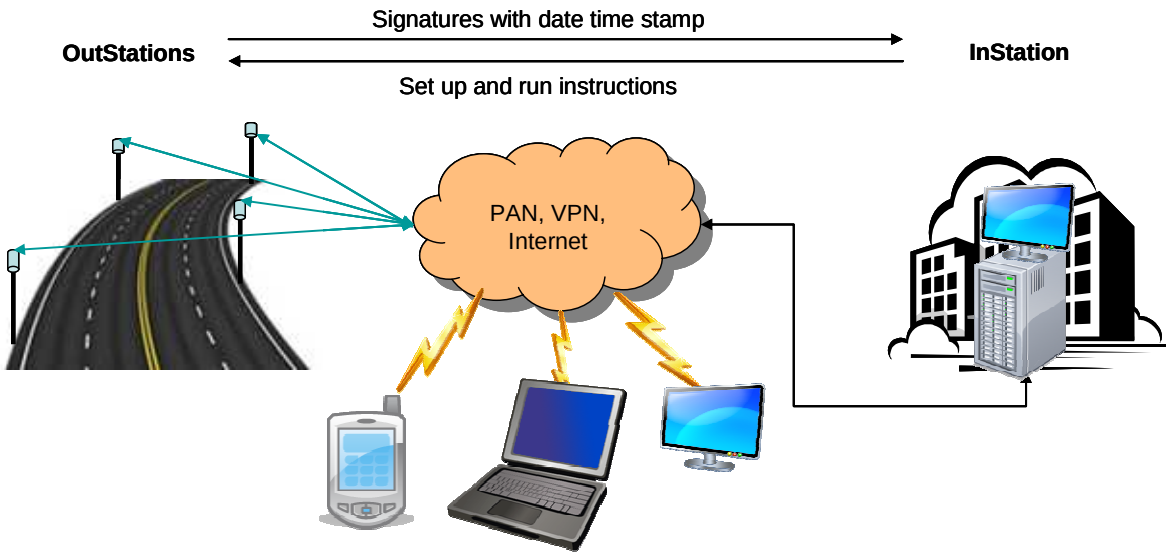

Figure 1: Overview of SSL BlueTruth System

2.2 Functionality

The main functions of the OutStation are:

- To provide Bluetooth device statistics.

The secondary / maintenance functions of the OutStation are:

- To provide SSH reverse communication to enable maintenance and future software upgrades.
- To provide time synchronisation by means of NTP.

To provide Bluetooth device statistic the OutStation will perform regular inquiries (scans) for the neighbouring devices using a Bluetooth adapter with a long range antenna. After each scan a raw journey times and congestion (queue) statistics report containing unique signatures of neighbouring devices (BT_ADDR) can be sent to the InStation for further data processing. The signature of the same device observed by two OutStations from different locations can be used to derive information about average journey time.
The configuration parameters of the OutStation will be provided in the functional configuration downloaded from the InStation.
In the case of a failure an alert and status report will be sent to the InStation.

The main functionality of the BlueTruth system is shown in Figure 2.
The main activities of the OutStation are shown in Figure 3 (version 3.0) and Figure 4 (version 4.0).

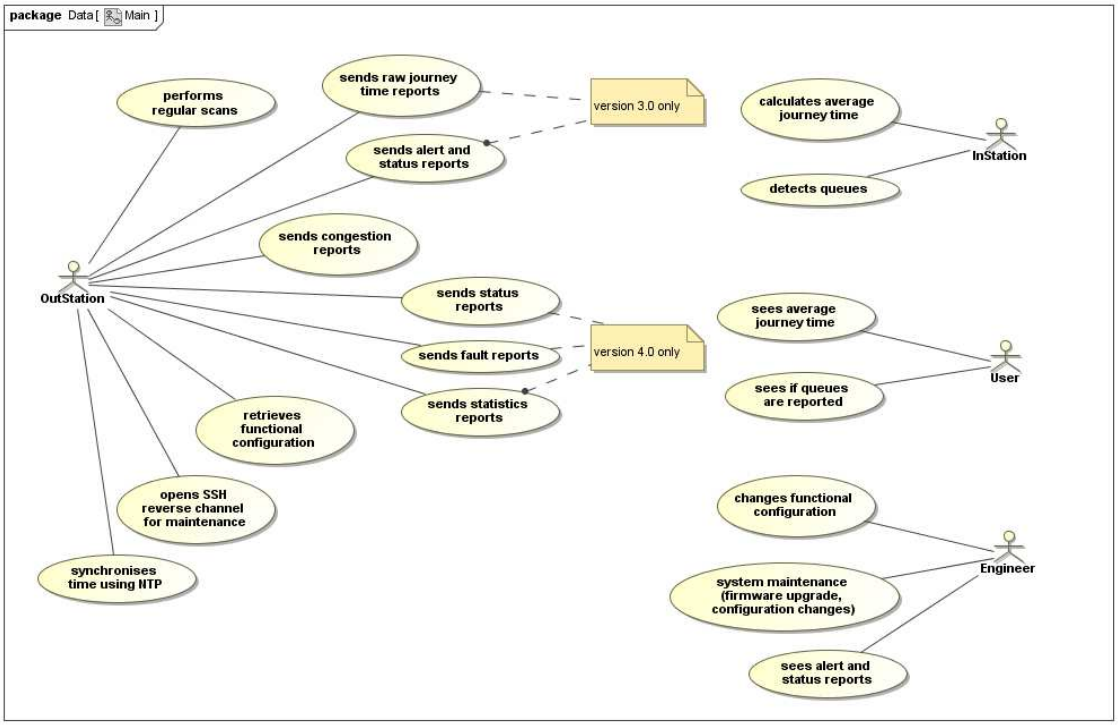

Figure 2: Main Functionality of the BlueTruth system

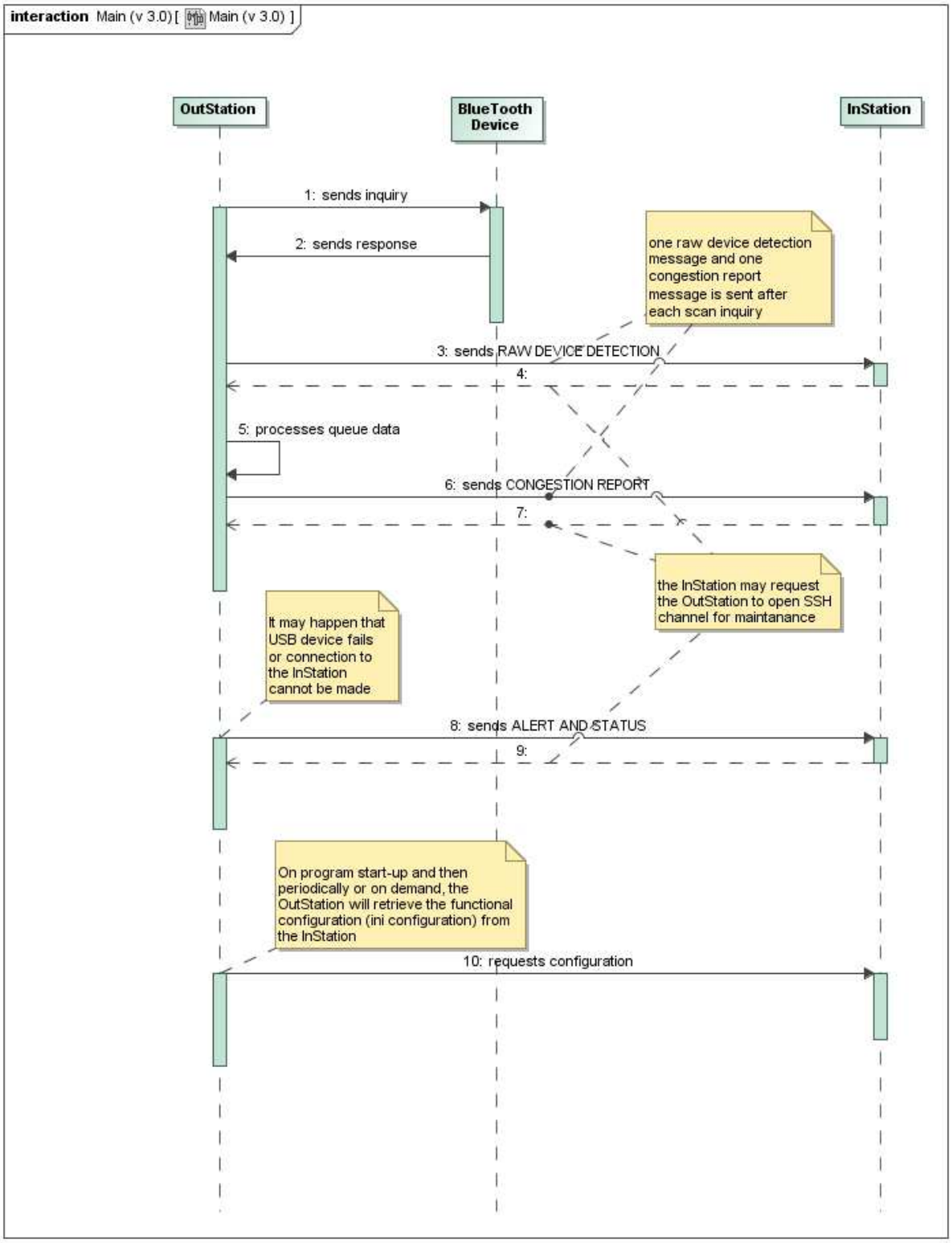

Figure 3: Overview of the OutStation functionality (version 3.0)

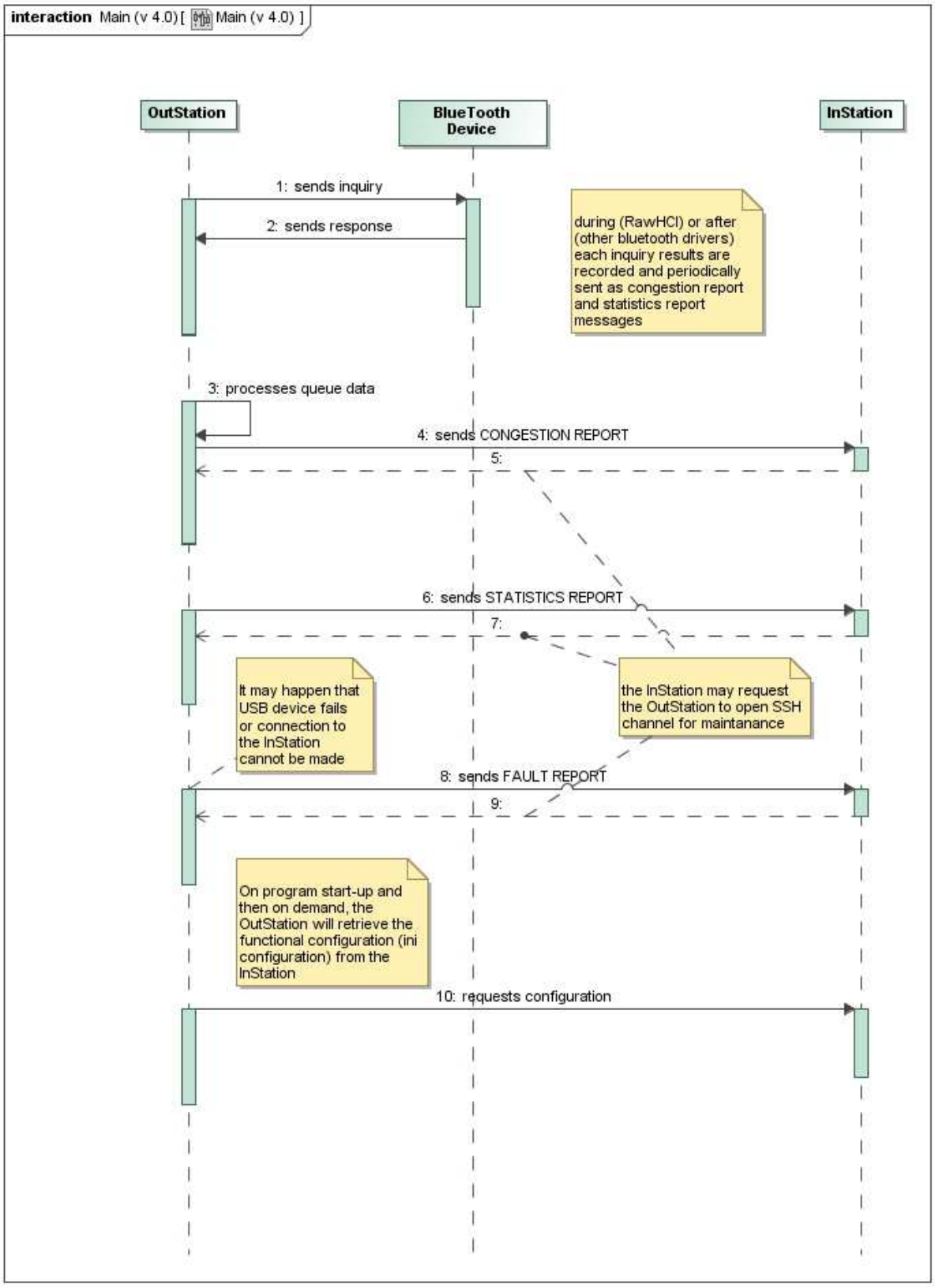

Figure 4: Overview of the OutStation functionality (version 4.0)

3 OutStation/InStation Communication Protocol

3.1 Connection Overview

The InStation and OutStation systems will be connected via a GPRS link. The two systems will use a standard TCP/IP stream socket connection to transfer data with HTTP protocol at the application layer. The OutStation system will be the client and InStation will be the server on the link. The InStation server will use a dedicated TCP/IP port.

The IP addresses of the server will be specified by configuration.

The OutStation system will initiate the stream socket connection and the InStation will listen for a connection request from the OutStation system.

The OutStation will send data with HTTP protocol minimising the amount of data in the message header. Both GET and POST commands will be used as described in the following sections. The *Content-Type* of messages will be either *application/x-www-form-urlencoded* (version 3.0) or *text/plain* (version 4.0). All accepted messages should be responded with Standard HTTP response of status class code 200. In the case when the InStation is to pass information to the OutStation it should put this information into the message body of the response.

From version 4.0, if the *functional_configuration / signReports* is set, each message sent out from an OutStation must be ended with a hex integer value from range <1,0xFFFF> called RND. This feature will provide a simple digital signature for the contents of the message. The value will be calculated from a seed using a pseudo-random number generator proposed by Park and Miller (see. http://c-faq.com/lib/rand.html) but only the last 2 bytes of the generated value will be sent in the message (the result will be ANDed with 0xFFFF value). The generator is defined as:

$X(n+1) = (aX(n) + c) \bmod m$ for a=48271, m=2147483647

The initial value for the generator will be provided by the InStation in a file *seed.xml* described in section 5.3 and uploaded over the secure SSH connection as described in section 7.1. The seed value and the contents of the file *seed.xml* will be changed by the OutStation at regular intervals (e.g. every 4000 messages). If the InStation decides to change the seed it will request opening of the reversed SSH connection (section 3.3.2), securely copy the file over the SSH tunnel and signal this fact to the OutStation in the response (sections 3.3.2, 3.3.6 and 5.3).

The link will be reset if a corrupt message is received or if idle conditions occur.

In the event of a disconnection the OutStation and InStation will try to deliver unsent messages for a specified (configurable) time and in the case of failure will discard any unsent or unanswered messages.

On disconnection the client (OutStation) will implement a back-off timer to prevent accumulating resources on both systems.

The OutStation can be in one of four modes; specified by the *functional_configuration / OutStationMode* parameter:

- Mode 0 – Idle. The outstation does not send RAW DEVICE DETECTION / STATISTICS REPORT or CONGESTION REPORT messages.
- Mode 1 – Journey Time. The outstation sends RAW DEVICE DETECTION / STATISTICS REPORT messages but does not send CONGESTION REPORT messages.
- Mode 2 – Occupancy. The outstation sends CONGESTION REPORT messages but does not send RAW DEVICE DETECTION / STATISTICS REPORT messages.
- Mode 3 – Journey Time & Occupancy. The outstation sends RAW DEVICE DETECTION / STATISTICS REPORT and CONGESTION REPORT messages.

In the following sections references to core and functional configuration parameters will be used; The core parameters will be marked with italics and have a form *core_configuration/parameterName*.

The functional parameters will be marked with italics and have a form *functional_configuration/parameterName.*

3.2 OutStation to InStation Message Description

The following messages will be sent between the InStation and the OutStation:

Message name	Version	Direction	Response
RAW DEVICE DETECTION	1-3	Out → In	None
CONGESTION REPORT	1-4	Out → In	None
ALERT AND STATUS	1-3	Out → In	None
STATUS REPORT	4	Out → In	None
FAULT REPORT	4	Out → In	None
REQUEST CONFIGURATION	1-4	Out → In	None
STATISTICS REPORT	4	Out → In	None

3.2.1 RAW DEVICE DETECTION

This message will be used to provide raw device detections used for calculation of expected journey time.

URL: *functional_configuration/urlJourneyTimesReporting*

HTTP method: POST

Message body format:

outstationID=<id>&startTime=<st>&devCount=<N>&d1=<d1_id>…&d<N>=<dN_id>

<id> – outstation site (detector) identifier,

<st> - inquiry start time (UTC) in format YY-mm-dd HH:MM:ss,

<N> - device count

<d1_id>…<dN_id> - Bluetooth device address (BD_ADDR).

Example:

POST /BlueTruthReceiver1_50/DeviceDetection HTTP/1.1
Host: 37.152.43.178:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 86

outstationID=&startTime=1970-01-01
00:00:00&devCount=2&d1=c8df7cc81be7&d2=000000001235

3.2.2 CONGESTION REPORT

This message will be used to provide information about queues and congestions (for details see Queue Detection Algorithm):

Version 1-3: URL: *functional_configuration/urlCongestionReporting*

Version 4: URL: *functional_configuration/urlCongestionReports*

HTTP method: POST.

Message body format:

Version 1-3: id=<id>&t=<t>&f=<f>&m=<m>&s=<s>&vs=<vs>&st=<st>&qs=<qs>|qe=<qe>

Version 4: <id>,<u>,<f>:<m>:<s>:<vs>:<st>,<qp>[,RND]

<id> - outstation site (detector) identifier,

<t> - time of report (UTC) in format YY-mm-dd HH:MM:ss,

<u> - time of report (UTC) in number of seconds since 1/1/1970 0:00 in hex format,

<f> - number of devices in the free flow bin,

<m> - number of devices in the moderate flow bin,

<s> - number of devices in the slow flow bin,

<vs> - number of devices in the very slow flow bin,

<st> - number of devices in the static flow bin,

<qs>/<qe> - queue start/end (UTC) in format YY-mm-dd HH:MM:ss. Only one value <qs> or <qe> can be submitted in the report. If a queue starts each subsequent CONGESTION REPORT message will contain <qs> field until the queue finishes, in which case <qe> will be sent once.

<qp> - queue present flag, 0 – no queue, 9 – queue present, FE - not ready, FF - fault. Other values are left for the future e.g. for soft queue detection.

RND – optional random positive integer 16-bit number as specified in section 3.1.

The OutStation will start sending CONGESTION REPORT messages with queue information with a delay of *functional_configuration / queueDetectStartupIntervalInSeconds* after program startup or USB bluetooth device recovery. Before that (but when USB bluetooth device is healthy) FE value will be reported to the InStation. In the case of USB bluetooth device failure FF value will be reported.

Example:

Version 3.0:

```
POST /BlueTruthReceiver1_50/Occupancy HTTP/1.1
Host: 37.152.43.178:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 52

id=1235&t=2013-10-07 17:00:21&f=2&m=0&s=1&vs=1&st=23&qs=2013-01-01 00:00:00
```

Version 4.0

```
POST /BlueTruthReceiver2_00/Occupancy HTTP/1.1
Host: 37.152.43.178:80
Content-Type: text/plain
Content-Length: 45

1235,2592c00,2:0:1:1:23,9,1234
```

3.2.3 ALERT AND STATUS

This message will be used to provide status and fault information about the outstation.

URL: *functional_configuration/urlAlertAndStatusReports*

HTTP method: POST

Message body format:

id=<id>&dt=<dt>&m=3&s=<code1>:<count1>…&<codeN>=<countN>

<id> – outstation site (detector) identifier,

<dt> - inquiry start time (UTC) in format YY-mm-dd HH:MM:ss,

<m> - message identifier,

<code1>:<count1>…<codeN>:<countN> - string comprised of a set of “code:count” message entries separated by commas. Different codes were defined in [4].

The following error codes will be used in version 3 legacy implementation:

- 054 - functional configuration file syntax error,
- 055 - functional configuration invalid parameter value,
- 058 - response from the InStation was not "STATUS OK 200" (server did not accept report),
- 059 - unable to retrieve functional configuration file,
- 066 - unable to send RAW DEVICE DETECTIONS
- 082 - Bluetooth device discovery error.

Example:
POST /BlueTruthReceiver1_50/Message HTTP/1.1
Host: 37.152.43.178:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 48

id=1234&dt=1970-01-01 00:00:00&m=3&s=100:5,101:4

3.2.4 STATUS REPORT

This message will replace ALERT AND STATUS message and be used to provide status information about the outstation. The message will be used as a response for OPEN SSH CONNECTION, CLOSE SSH CONNECTION, RELOAD_CONFIGURATION and GET_STATUS_REPORT.

URL: *functional_configuration/urlStatusReports*

HTTP method: POST

Message body format:

<id>,<u> <status1>,...,<statusN>[,RND]

<id> – outstation site (detector) identifier,,

<u> - inquiry start time (UTC) in number of seconds since 1/1/1970 0:00 in hex format,

<status1>,…,<statusN> - a list of N status records separated by the & character. Each status record will be of the format: <sn>=<value>

where:

<sn> - status name,

<value> - a string containing a meaningful value of status name variable,

RND – optional, random positive integer 16-bit number as specified in section 3.1.

The following status variables have been defined:

Status Name	Type	Description

boot		This status is provided to confirm request to reboot. 0 – no intention to reboot, 1 – about to reboot.
fv	String	Firmware version
sn	String	Serial number read from core configuration file
cv	String	Configuration version - functional in-use configuration MD5 hash value
sl	Integer	Signal level, -255..255 - average GSM modem signal level in dBm over the last period (configurable in functional configuration, default value 5 min). Value of -255 will mean value not available.
of	Integer	Obfuscation function used to encode Bluetooth device addresses. Values: 0 – no obfuscation 1 – rand1 function used (see 3.2.7)
key	Integer	0 – request for a new private/public key (see section 7.1)
seed	Integer	0 – request for a new value or value cannot be read, seed id – random number provided by the InStation used to confirm use of the latest update of the seed (see section 5.3).
ssh	String	Status of the ssh connection. Values: “closed” (when closed), “open xx.xx.xx.xx:yy zz” (when open, xx.xx.xx.xx is the InStation address, yy is the InStation local port number, zz is the remote port number), “not supported”.
up	Integer	Up-time – the number of seconds the BlueTruth application have been running since its start.

Other values e.g. average cpu load, memory used, temperature can be added in the future.

Example:

POST /x/z.html HTTP/1.1
Host: 37.152.43.178:80
Content-Type: text/plain
Content-Length: 38

1234,2592c00,fv=0.99,ssh=closed,3245

3.2.5 FAULT REPORT

This message will replace ALERT AND STATUS message and be used to provide fault information about the outstation.

URL: *functional_configuration/urlFaultReports*

HTTP method: POST

Message body format:

<id>,<u>, <fault1>,...,<faultN>[,RND]

<id> – outstation site (detector) identifier,,

<u> - inquiry start time (UTC) in number of seconds since 1/1/1970 0:00 in hex format,

<fault1>,…,<faultN> - a list of N fault records separated by the character ‘,’ (comma). Each fault record will be of the format: <fn>:<u>:<status>

where:

<fn> - fault number,

<u> - time when fault was set or cleared, as number of seconds since 1/1/1970 0:00 in hex format,

<status> - fault status: 0 – cleared, 1 – set.

RND – optional, random positive integer 16-bit number as specified in section 3.1.

The following fault numbers have been defined:

Fault Name	Fault Number	Description
Configuration	50-99	Various failures related to configuration files 51 - functional configuration syntax error, 52 - functional configuration invalid parameter/-s value. 60 - missing or invalid seed file
Bluetooth device failure	100-199	Various failures related to Bluetooth device 100 - General device failure, device has been removed, device stopped communicating
InStation communication failure	200-299	InStation Communication failures groups: 2x1 - Retrieve Configuration Client, 2x2 - Congestion Report Client, 2x3 - Raw Device Detection Report Client, 2x4 - Alert and Status Report Client, 2x5 - Status Reporting Client, 2x6 - Fault Reporting Client, 2x7 - Statistics Reporting Client. Specific failures: 20x - Failure to establish communication to the InStation or InStation does not respond, 21x - InStation does not accept reports responding not with Status OK, 22x - InStation response message body contains error and the OutStation is not able tot parse it.
SSH connection failure	300-310	The InStation has requested to open ssh connection but the OutStation did not managed to. 300 – server does not respond or does not allow to connect.

GSM Modem connection failure	400-410	400 – unable to connect to GSM modem over ssh connection.

Example:

POST /x/z.html HTTP/1.1
Host: 37.152.43.178:80
Content-Type: text/plain
Content-Length: 52

1234,2592c00,101:386d4380:1,102:386d438a:0,ABCD

3.2.6 REQUEST CONFIGURATION

This message will be used to retrieve the functional configuration off the InStation

URL: core_configuration / <Ini_Configuration_URL_Path><Ini_Configuration_URL_FilePrefix><SerialNumber>< Ini_Configuration_URL_FileSuffix>

HTTP method: GET

Message body format: None

Example:

GET /DetectorConfigurationDownload/1_50/12347_ini.txt HTTP/1.1
Host: 127.0.0.1:80
Accept: */*

3.2.7 STATISTICS REPORT

This message will be introduced to optimise the amount of information exchanged between the InStation and the OutStations. The message will be of the form:

URL: *functional_configuration/urlStatisticsReports*

HTTP method: POST.

Message body format:

<id>,<st>,<Δet>,<d1>,…,<dN>,RND

<i> - outstation site (detector) identifier,

<st> - report start time (UTC) in number of seconds since 1/1/1970 0:00 in hex format,

<Δet> - report end time in number of seconds from <st> in hex format,

<d1>,…,<dN> - a list of N device records separated by the & character. Each device record will be of the format: <m>:<cod>:<tf>:<t0>:<tl> . In the case of no devices detected this field will be empty.

<m> - obfuscated Bluetooth device address (BD_ADDR). The following functions may be used:

- 0 - No obfuscation – 12:34:56:78:90:AB address will passed as 1234567890AB,
- 1 - rand1 function – random 16 bit number multiplied by 2^48 will be added to the number resulting from “no obfuscation” case and then bytes will be remapped 1->4, 2->7, 3->1, 4->8, 5->2, 6->5, 7->6, 8->3. Example: address “1234567890AB” will be converted to “5690XX12ABXX3478” where X will denote a random 4bit value.

<cod> - class of device (in hex),

<tf> - first observation time (UTC) in number of seconds since 1/1/1970 0:00 in hex format,

<t0> - estimate of reference point observation time in number of seconds since <tf> in hex format. If a device has not left the scanned area it will be reported as 0.

<tl> - last observation time in number of seconds since <tf> in hex format. If a device has not left the scanned area it will be reported as 0. If a device has been moved to the Background Bin and is perceived as background it will be reported as FFFFFFFE. In the case of USB Bluetooth device fault all devices will be reported as FFFFFFFF.

RND – random positive integer 16-bit number as specified in section 3.1.

The last observation will be reported if the device was absent in the monitoring area in the last T_{ABS} seconds (*functional_configuration / btMACBinDropOutScanCycle* or *absenceThresholdInSeconds*).

The STATISTICS REPORT message will be sent at least every T_{STAT} time (*functional_configuration / statisticsReportPeriodInSeconds*). Depending on the configuration parameter *functional_configuration/statisticsReportContents* devices will be reported in the following way:

- *statisticsReportContents=full*: Each device will be reported if is has been observed regardless of wheather it left the scanned area, became background or fault,
- *statisticsReportContents=brief*: Each device will be reported if it has been observed and either left the scanned area (see *functional_configuration/absenceThresholdInSeconds*), has been categorised as background (see *functional_configuration / backgroundLatchTimeThresholdInSecond)* or faulty.

. Devices may be reported multiple times if they have entered (and left) the monitoring area multiple times between the subsequent STATISTICS REPORT messages.

A device may be categorised as a background device if it remains in the area for longer than $T_{BACK,S}$ (*functional_configuration / backgroundLatchTimeThresholdInSeconds*). A device may be removed from the background devices list if it remains in the area for longer than $T_{BACK,E}$. (*functional_configuration / backgroundClearanceTimeThresholdInSeconds*). The background devices will not be reported to the InStation. The OutStation will create a persistent list of background devices.

Some other conditions may be added in the future for a device to become background.

If no devices are to be reported an empty STATISTICS REPORT message will be sent, i.e. containing no device records.

Example:

POST /x/z.html HTTP/1.1
Host: 37.152.43.178:80
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
Content-Length: 74

1234,0,3c,abcdef012345:123:10:20:30,abcdef543210:123:10:20:40,0234

3.3 OutStation to InStation Message Description

The InStation may pass information about changes related to functional configuration, requests to open secure communication channel, etc. This information should be put into the message body of the response to the first message received from the OutStation. The HTTP *Content Type* for such a message must be defined as *text/plain*. Each such request must be confirmed by the OutStation with STATUS REPORT message containing the relevant data.

3.3.1 RELOAD CONFIGURATION

This response will be used to inform the OutStation about changes in the functional configuration. The OutStation should regularly download its functional configuration but this message response will be used to instruct reloading of the configuration.
Response message body format:

reloadConfiguration

Other commands may be added to this request and should be separated by the ‘,’ character.

3.3.2 OPEN SSH CONNECTION

This response message will be used whenever the InStation would like to have access to the OutStation via reversed SSH connection. The SSH access is required for any kind of maintenance tasks including firmware upgrades or core configuration changes.

Response message body format:

openSSHConnection:<rep>

where:

<rep> - is the remote entry port.

Expected response: STATUS REPORT containing information about SSH status.

Other commands may be added to this request and should be separated by the ‘,’ character.

Example:

HTTP/1.1 200 OK
Date: Sat, 05 Oct 2013 21:21:12 GMT
Content-Type: text/plain
Content-Length: 23
Connection: close

openSSHConnection:50000

3.3.3 CLOSE SSH CONNECTION

This response message will be used whenever the InStation would like to close the existing reversed SSH connection from the OutStation established after the OPEN SSH CONNECTION message.

Response message body format:

closeSSHConnection

Expected response: STATUS REPORT containing information about SSH status.

Other commands may be added to this request and should be separated by the “,” character.

3.3.4 GET STATUS REPORT

This response message will be used whenever the InStation would like to receive report of the current status of the OutStation STATUS REPORT.

Response message body format:

getStatusReport

Expected response: full STATUS REPORT.

Other commands may be added to this request and should be separated by the “,” character.

3.3.5 REBOOT

This response message will be used whenever the InStation would like the OutStation to be rebooted.
Response message body format:

reboot

Expected response: STATUS REPORT confirming intention to reboot.

3.3.6 CHANGE SEED

This response message will be used whenever the InStation would like the OutStation to reread the contents of the seed file *seed.xml*.
Response message body format:

changeSeed

Expected response: STATUS REPORT containing information about seed id to be used.

Other commands may be added to this request and should be separated by the "," character.

3.3.7 FLUSH BACKGROUND

This response message will be used whenever the InStation would like the OutStation to clear the status of all devices perceived as background to normal
Response message body format:

flushBackground

Expected response: full STATUS REPORT.

Other commands may be added to this request and should be separated by the "," character.

3.3.8 LATCH BACKGROUND

This response message will be used whenever the InStation would like the OutStation to set the status of devices that are not perceived as normal to background. Only these devices will be latched that stay in the area for longer than *durationInSeconds*.
Response message body format:

latchBackground:durationInSeconds

Expected response: full STATUS REPORT.

Other commands may be added to this request and should be separated by the "," character.

3.4 Start-up Sequence

Hardware:
It is possible that using certain combinations of Vortex SBC board and Bolutek Bluetooth USB module that on power up the USB device will not be visible by the Vortex SBC and the operating system. To overcome this problem the Bolutek Bluetooth USB port will be electrically switched off for a short period during operating system bootup using Vortex SBC GPIO port controlling a power MOSFET, (SSL product number 709, Vortex USB Power Control Module). Once the operating system is running the Bolutek USB port will be continuously powered on.

Version 3:
None

Version 4:
On every start-up the OutStation must send full STATUS REPORT message containing current status of the OutStation including firmware version, configuration MD5 hash.

3.5 Periodic Activities

The following activities related to InStation interface will be performed periodically:
Version 3:

- Download of the functional configuration file (see 3.2.6). In modes 1, 2 and 3 *functional_configuration / settingsCollectionInterval1* parameter will define download

interval, In mode 0 functional_configuration / settingsCollectionInterval2 parameter will define download interval.

Version 4:

- Download of the functional configuration file (see 3.2.6). In modes 1, 2 and 3 *functional_configuration / settingsCollectionInterval1* parameter will define download interval, In mode 0 *functional_configuration / settingsCollectionInterval2* parameter will define download interval.
- Upload STATISTICS REPORT. The interval to sent this message will be specified by *functional_configuration /statisticsReportPeriodInSeconds* parameter
- Upload CONGESTION REPORT. The interval to send this message will be specified by *functional_configuration / congestionReportPeriodInSeconds* parameter.
- Upload STATUS REPORT. The interval to send this message will be specified by *functional_configuration / statusReportPeriodInSeconds* parameter.

3.6 Error Handling

In the case where one party does not respond to a request message, the requesting party will wait for T_{ROUT} time (*functional_configuration / httpTimeOut* or *httpResponseTimeOutInSeconds*). After this timeout the requesting party will retry to send a request until timeout T_{COUT} happens (*functional_configuration / httpConnectionTimeOutInSeconds*). If it still does not receive expected response it will assume that the connection is broken, will delete all accumulated messages and report an alarm for the first occurrence.

If the TCP connection is broken, the OutStation will continue to collect statistic.

In the case of broken connection the client will regularly try to establish communication.

After the connection is re-established the OutStation will send all previously acquired statistics to the InStation but only if timeout T_{TOUT} has not occurred.

3.7 Communication Parameters

The communication parameters are described in the following table:

Parameter	Default Value	Description
T_{ROUT}	15 s	Response timeout (*functional_configuration / httpTimeOut* (v3) *or httpConnectionTimeOutInSeconds* (v4))
T_{COUT}	300 s	Communication timeout (*functional_configuration / httpConnectionTimeOutInSeconds*

4 BlueTooth

The OutStation will perform periodic inquiries (scans) for remote devices using local Bluetooth radio using native HCI commands or operating system API calls. After each scan the following messages may be sent to the InStation (see OutStation modes) containing data related to the last scan:

- Version 3.0: RAW DEVICE DETECTION (3.2.1), CONGESTION REPORT (3.2.2).
- Version 4.0+: CONGESTION REPORT (3.2.2), STATUS REPORT (3.2.4), FAULT REPORT (3.2.5), STATISTICS REPORT (3.2.7),

In the case of no detected devices no RAW DEVICE DETECTION message will be sent.

The frequency of device inquiries is defined by the *functional_configuration/inquiryCyclePeriod* (v3) or *inquiryCycleDurationInSeconds* (v4) parameter.

4.1 Queue Detection

The BlueTruth will categorise discovered Bluetooth devices into the following categories (bins):

- Free Flow.
- Moderate Flow.
- Slow Flow.
- Very Slow Flow.
- Static Flow.
- Background.

Each bin will be configurable in terms of number of scan cycles.
The BlueTruth OutStation will track each device by its unique address (BD_ADDR) to measure the amount of time the device is seen and store them in the appropriate bin. This will mean that for slow down in traffic (queue) devices will move from the Free Flow bin towards the Static bin.
The Background bin will store all devices which exceed the Static bin category (e.g. a person working in the office).
Each device will be dropped out of the relevant bin if it is not seen for a configurable number of cycles *functional_configuration / btMACBinDropOutScanCycle* (v3) or seconds *functional_configuration / absenceThresholdInSeconds* (v4).
A queue will be reported if the number of devices in the bin specified in parameter *functional_configuration / queueAlertThresholdBin* or above (i.e. higher rank bins) exceed the value of parameter *functional_configuration / queueDetectThreshold.* A previously reported queue will be cleared if the number of devices in the bin specified in parameter *functional_configuration / queueAlertThresholdBin* or above (i.e. higher rank bins) drop below the value of parameter *functional_configuration / queueClearanceThresholdDetectionNumber* (v3) or *functional_configuration / queueClearanceThreshold* (v4).
The following configuration parameters will be used for queue detection:
Version 3.00:

- *functional_configuration/freeFlowSpeedCyclesThreshold* – maximum number of scan cycles a device has to be visible to be included in “Free Flow” bin.
- *functional_configuration/moderateSpeedCyclesThreshold* – maximum number of scan cycles above *freeFlowSpeedCyclesThreshold* a device has to be visible to be included in “Moderate Flow” bin.
- *functional_configuration/slowSpeedCyclesThreshold* – maximum number of scan cycles above *freeFlowSpeedCyclesThreshold* + *moderateSpeedCyclesThreshold* a device has to be visible to be included in “Slow Flow” bin.
- *functional_configuration/verySlowSpeedCyclesThreshold* – maximum number of scan cycles above *freeFlowSpeedCyclesThreshold* + *moderateSpeedCyclesThreshold* + *slowSpeedCyclesThreshold* a device has to be visible to be included in “Very Slow Flow”

bin. All devices that have been visible for more than *freeFlowSpeedCyclesThreshold* + *moderateSpeedCyclesThreshold* + *slowSpeedCyclesThreshold* + *verySlowSpeedCyclesThreshold* will be included in the "Static" bin.

Version 4.00+

- *functional_configuration/freeFlowBinThresholdInSeconds* – maximum number of seconds a device has to be visible to be included in "Free Flow" bin.
- *functional_configuration/moderateFlowBinThresholdInSeconds* – maximum number of seconds a device has to be visible to be included in "Moderate Flow" bin.
- *functional_configuration/slowFlowBinThresholdInSeconds* – maximum number of seconds a device has to be visible to be included in "Slow Flow" bin.
- *functional_configuration/verySlowFlowBinThreshold* – maximum number of seconds a device has to be visible to be included in "Very Slow Flow" bin.
- All devices that have been visible for more than *backgroundLatchTimeThresholdInSeconds* will be included in the "Background" bin.
- All devices that have been present in "Background" and than absent for more than *backgroundEndTimeThresholdInSeconds* will be removed from the "Background" bin.

Remark for all bins: If a device is intermittently visible with gaps less than *functional_configuration / btMACBinDropOutScanCycle* (v3) or *absenceThresholdInSeconds* it will be still counted as visible. The scan cycle count will include scan cycles when it was not observed,

Version 3.0 (e.g. inquiryCyclePeriod = 10):

Number of observations:
A=3: "Free Flow" bin,
B=8 (4+4): "Moderate Flow" bin,
C=13 (4+6+3): "Slow" bin,
D=20 (4+6+7+3): "Very Slow" bin,
E=30 (4+6+7+9+4): "Static" bin

Ini configuration values:
freeFlowSpeedCyclesThreshold=4
moderateSpeedCyclesThreshold=6
slowSpeedCyclesThreshold=7
verySlowSpeedCyclesThreshold=9

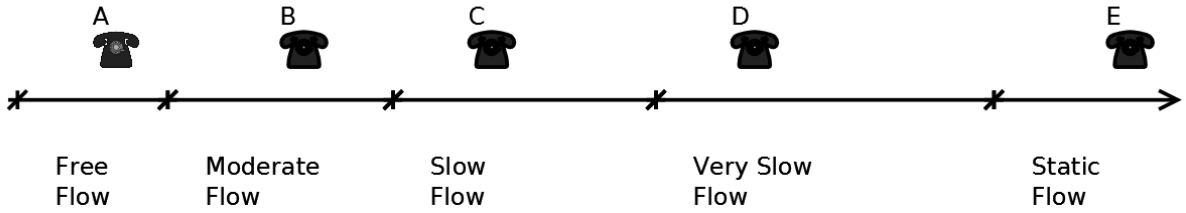

Figure 5 Queue Detection. Counting of Devices in Various Bins

For version 4.0 the functional configuration values would have the following values:
inquiryCycleDurationInSeconds = 10,
freeFlowSpeedCyclesThresholdInSeconds = 40
moderateSpeedCyclesThresholdInSeconds = 100
slowSpeedCyclesThresholdInSeconds = 170
verySlowSpeedCyclesThresholdInSeconds = 260

5 Configuration

The BlueTruth will have a few configuration files:

- core configuration - containing basic communication parameters.
- functional configuration - containing other parameters affecting algorithms.
- seed file – containing seed id and value used to sign messages sent from OutStations to the InStation to protect against spoofing.

5.1 Core Configuration

This file will be installed in */opt/bt/etc* directory as *core_configuration.xml*.
The following fields will be defined in the core configuration file:

- Version (Version) - BlueTruth OutStation protocol to be used when communicating to InStations.
- OutStation serial number (Identity/SerialNumber).
- SSH connection to the InStation parameters:
    - address (InStationSSHConnection/Address),
    - port (InStationSSHConnection/Port),
    - login (InStationSSHConnection/Login),
    - password (InStationSSHConnection/Password)
- functional configuration URL. The functional configuration URL will be build from the following items:
    - path (IniConfigurationURL/Path),
    - prefix (IniConfigurationURL/Prefix),
    - outstation serial number (/Identity/SerialNumber),
    - suffix (IniConfigurationURL/Suffix),

  URL = Path + Prefix + SerialNumber + Suffix

  e.g. for Path = "http://37.152.43.178/DetectorConfigurationDownload/1_50/", Prefix = "", SerialNumber="178534", Suffix="_txt.ini"

  URL="http://37.152.43.178/DetectorConfigurationDownload/1_50/178534_ini.txt"

  If this URL is empty the BlueTruth will use the local file functional_configuration.conf. If this URL contains a value the last saved version (last_functional_configuration.conf) will be loaded on program startup, and any newer downloaded file will be used after startup.
- Preferred Bluetooth device address (BD_ADDR) to be used (BlueToothDevice / MAC_AddressOfDeviceToBeUsed). This feature will be used in the case of multiple Bluetooth local devices (radios) available to indicate preference. Leave blank to use any device.
- Preferred Bluetooth device driver (BlueToothDevice / Driver). For Linux the following drivers are implemented:
    - raw HCI (RawHCI), using raw HCI commands,
    - native bluez library API (NativeBluez),
    - using special protocol using AT Commands for Parani SD1000U device (Parani).
- Logging level (file - Logging/FileLogLevel /console - Logging/ConsoleLogLevel), maximum number of entries per file (Logging / MaxNumberOfEntriesPerFile) and

maximum log file age (Logging / MaxLogFileAgeInSeconds). The log level will indicate the threshold above which log entries should be recorded to the console or the file. The files exceeding maximum log file age will be regularly deleted by the OS utilities.

The core file will be stored in xml format as *core_configuration.xml* with the following schema .

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.simulation-systems.co.uk/BlueTruth/coreConfigurationSchema"
    xmlns:coreConfiguration="http://www.simulation-
systems.co.uk/BlueTruth/coreConfigurationSchema">
    <xs:simpleType name="BlueTruthVersionType">
        <xs:restriction base="xs:integer">
            <xs:pattern value="[0-9]\.[0-9]\.[0-9]"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="IdentityType">
        <xs:sequence>
            <xs:element name="SiteIdentifier" type="xs:string"/>
            <xs:element name="SerialNumber" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="InStationSSHConnectionType">
        <xs:sequence>
            <xs:element name="Address" type="xs:string"/>
            <xs:element name="Port" type="xs:unsignedShort" default="22" minOccurs="0"
maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="IniConfigurationURLType">
        <xs:sequence>
            <xs:element name="Ini_Configuration_URL_Path" type="xs:string"/>
            <xs:element name="Ini_Configuration_URL_File_Prefix" type="xs:string"/>
            <xs:element name="Ini_Configuration_URL_File_Suffix" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DriverType">
        <xs:choice>
            <xs:element name="WindowsBluetooth"/>
            <xs:element name="WindowsWSA"/>
            <xs:element name="Parani" type="coreConfiguration:ParaniType"/>
            <xs:element name="NativeBluez"/>
            <xs:element name="RawHCI"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="ParaniType">
        <xs:attribute name="PortName" type="xs:string" default="/dev/ttyUSB0"/>
        <xs:attribute name="BitRate" type="xs:unsignedInt" default="115200"/>
    </xs:complexType>
    <xs:complexType name="BlueToothDeviceType">
        <xs:sequence>
            <xs:element name="MAC_AddressOfDeviceToBeUsed" type="xs:string"/>
            <xs:element name="Driver" type="coreConfiguration:DriverType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="GSMModemConnectionType">
        <xs:sequence>
            <xs:element name="Type" type="xs:unsignedShort"/>
            <xs:element name="Address" type="xs:string"/>
            <xs:element name="Port" type="xs:unsignedShort" default="22" minOccurs="0"
maxOccurs="1"/>
            <xs:element name="Login" type="xs:string"/>
            <xs:element name="Password" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="LoggerType">
        <xs:sequence>
            <xs:element name="FileLogLevel" type="xs:unsignedInt"/>
            <xs:element name="ConsoleLogLevel" type="xs:unsignedInt"/>
            <xs:element name="MaxNumberOfEntriesPerFile" type="xs:unsignedInt"/>
            <xs:element name="MaxNumberOfCharactersPerFile" type="xs:unsignedInt"/>
            <xs:element name="MaxLogFileAgeInSeconds" type="xs:unsignedLong"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="CoreConfigurationType">
        <xs:sequence>
            <xs:element name="Version" type="coreConfiguration:BlueTruthVersionType"/>
            <xs:element name="Identity" type="coreConfiguration:IdentityType"/>
```

```
                    <xs:element name="InStationSSHConnection"
type="coreConfiguration:InStationSSHConnectionType"/>
                    <xs:element name="IniConfigurationURL"
type="coreConfiguration:IniConfigurationURLType"/>
                    <xs:element name="BlueToothDevice"
type="coreConfiguration:BlueToothDeviceType"/>
                    <xs:element name="GSMModemConnection"
type="coreConfiguration:GSMModemConnectionType"/>
                    <xs:element name="AlgorithmParameters"
type="coreConfiguration:AlgorithmParametersType"/>
                    <xs:element name="Logging" type="coreConfiguration:LoggerType"/>
            </xs:sequence>
    </xs:complexType>
    <xs:element name="CoreConfiguration" type="coreConfiguration:CoreConfigurationType"/>
</xs:schema>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<coreConfiguration:CoreConfiguration xmlns:coreConfiguration="http://www.simulation-
systems.co.uk/BlueTruth/coreConfigurationSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.simulation-systems.co.uk/BlueTruth/coreConfigurationSchema
misc/core_configurationSchema.xsd">
    <Version>4.0.3</Version>
    <Identity>
            <SiteIdentifier>3421</SiteIdentifier>
            <SerialNumber>1234</SerialNumber>
    </Identity>
    <InStationSSHConnection>
            <Address>service.bluetruth.co.uk</Address>
            <Login>bt</Login>
            <Password></Password>
    </InStationSSHConnection>
    <IniConfigurationURL>
            <Path>http://service.bluetruth.co.uk/DetectorConfigurationDownload/2_00/</Path>
            <File_Prefix></File_Prefix>
            <File_Suffix>_ini.txt</File_Suffix>
    </IniConfigurationURL>
    <BlueToothDevice>
            <Driver><RawHCI/></Driver>
    </BlueToothDevice>
    <GSMModemConnection>
            <Type>1</Type>
            <Address>192.168.2.1</Address>
            <Login>adm</Login>
            <Password>123456</Password>
    </GSMModemConnection>
    <Logging>
            <FileLogLevel>0</FileLogLevel>
            <ConsoleLogLevel>1</ConsoleLogLevel>
            <MaxNumberOfEntriesPerFile>5000</MaxNumberOfEntriesPerFile>
            <MaxNumberOfCharactersPerFile>2000000</MaxNumberOfCharactersPerFile>
            <MaxLogFileAgeInSeconds>345600</MaxLogFileAgeInSeconds>
    </Logging>
</coreConfiguration:CoreConfiguration>
```

For security reasons this file can be modified or exchanged only by means of the communication over SSH channel.

5.2 Functional Configuration

This configuration will be either deployed locally in */opt/bt/etc* directory as *functional_configuration.conf* or exchanged in the contents of REQUEST CONFIGURATION message and stored locally in */var/cache/bt* directory as *last_functional_configuration.conf.*

The contents of the configuration for version 3 have been defined in [4]. The parameters that are used have been presented in the following table:

Name	Version	Default Value	Description

OutStationMode	3	3	Mode 0 – Idle. The outstation does not send RAW DEVICE DETECTION / STATISTICS REPORT or CONGESTION REPORT messages. Mode 1 – Journey Time. The outstation sends RAW DEVICE DETECTION / STATISTICS REPORT messages but does not send CONGESTION REPORT messages. Mode 2 – Occupancy. The outstation sends CONGESTION REPORT messages but does not send RAW DEVICE DETECTION / STATISTICS REPORT messages. Mode 3 – Journey Time & Occupancy. The outstation sends RAW DEVICE DETECTION / STATISTICS REPORT and CONGESTION REPORT messages
outStationMode	4		
settingsCollectionInterval1	3& 4	15	Period, in minutes, between retrieving the functional configuration in Mode 1, 2 and 3, range 0..1440 (24h)
settingsCollectionInterval2	3& 4	5	Period, in minutes, between retrieving the functional configuration in Mode 0, range 0..1400 (24h)
urlCongestionReporting	3		URL to send CONGESTION REPORT
urlCongestionReports	4		
urlJourneyTimesReporting	3		URL to send RAW DEVICE DETECTION
urlAlertAndStatusReports	3		URL to send ALERT AND STATUS
urlStatusReports	4		URL to send STATUS REPORT
urlFaultReports	4		URL to send FAULT REPORT
urlStatisticsReports	4		URL to send STATISTICS REPORT
inquiryCyclePeriod	3	10	Duration of the inquiry cycle in seconds, range 0..60 (0 - no scanning)
inquiryCycleDurationInSeconds	4	10	
inquiryPower	4	20	Inquiry transmit power level used to transmit the inquiry data packets in dBm. Range -70..20 (see [5] section 7.3.62)
obfuscatingFunction	4	0	BD_ADDR Obfuscation function. 0 - no obfuscation, 1 - rand1 function will be used (see 3.2.7).
statisticsReportPeriodInSeconds	4	60	Period, in seconds, of sending STATISTIC REPORT, range 0..600
congestionReportPeriodInSeconds	4	60	Period, in seconds, of sending CONGESTION REPORT, range 0..600
statusReportPeriodInSeconds	4	600	Period, in seconds, of sending STATUS REPORT, range 0..3600

backgroundLatchTimeThresholdInSeconds	4		see 4.1, the value of this parameter is site dependant, range 0..86400 (24h)
backgroundClearanceTimeThresholdInSeconds	4		see 4.1, the value of this parameter is site dependant, range 0..2678400 (31 days)
freeFlowSpeedCyclesThreshold	3		see 4.1, the value of this parameter is site dependant. Range 0..10000 (v3) or 0..86400 (24h, v4)
freeFlowBinThresholdInSeconds	4		
moderateSpeedCyclesThreshold	3		see 4.1, the value of this parameter is site dependant. Range 0..10000 (v3) or 0..86400 (24h, v4)
moderateFlowBinThresholdInSeconds	4		
slowSpeedCyclesThreshold	3		see 4.1, the value of this parameter is site dependant. Range 0..10000 (v3) or 0..86400 (24h, v4)
slowFlowBinThresholdInSeconds	4		
verySlowSpeedCyclesThreshold	3		see 4.1, the value of this parameter is site dependant. Range 0..10000 (v3) or 0..86400 (24h, v4)
verySlowFlowBinThresholdInSeconds	4		
btMACBinDropOutScanCycle	3		see 4.1, the value of this parameter is site dependant. Range 0..10000 (v3) or 0..86400 (24h, v4)
absenceThresholdInSeconds	4	10	
queueAlertThresholdBin	3&4		Name of the bin to be used to for detection of a queue. see 4.1 for details. Allowed values: freeFlow, moderateFlow, slowFlow, verySlowFlow and staticFlow.
queueDetectThreshold	3&4		Threshold that determines how many detected devices must be present in the relevant bin (*queueAlertThresholdBin*) to start reporting of the queue. The value of this parameter is site dependant. Range 0..65535
queueClearanceThresholdDetectionNumber	3		If the number of devices in the queueAlertThresholdBin drops below this threshold the queue will be reported as cleared. The value of this parameter is site dependant. Range 0..65535.
queueClearanceThreshold	4		
queueDetectionStartupIntervalInSeconds	4		Time interval after which the OutStation may make judgement about queue presence. This value is used to protect the OutStation from stating about queue while not having sufficient amount of data. This value should be above the threshold value of the bin specified in queueAlertThresholdBin parameter.
signReports	4	0	Enable (=1) or disable (=0) appending of a random number to each STATISTICS REPORT or CONGESTION REPORT to sign report. See 3.1 for details.
httpTimeOut	3	15	Period, in seconds, of time the OutStation will

httpResponseTimeOutInSeconds	4	15	wait for the target URL to respond before quitting this attempt and trying again. Range 0..600.
httpConnectionTimeOutInSeconds	4	300	Period, in seconds, of time the OutStation will try to deliver accumulated messages to the InStation, after which (if failed) it will discard all messages and raise the relevant fault. Range 0..600.
gsmModemSignalLevelSamplingPeriodInSeconds	4	30	Sampling period, in seconds, to record GSM modem signal level. Range 0..600.
gsmModemSignalLevelStatisticsWindowInSeconds	4	300	Window in seconds, over which the average value of GSM signal level will be calculated. Range 0..3600. The quotient of gsmModemSignalLevelStatisticsWindowInSeconds and gsmModemSignalLevelStatisticsWindowInSeconds must not exceed 100.
statisticsReportContents	4	brief	Values *full* and *brief*. *Full* – devices will be reported after being spotted and may still be in the scanning area, *brief* – devices will be reported after being spotted and only if left the area (*absenceThresholdInSeconds* has been exceeded)
reportStorageCapacity	4	1000	Maximum number of messages (e.g. STATUS REPORT) to be stored in the internal queue. If the number of messages in the internal queue is *reportStorageCapacity* and a new message is to be appended the oldest message will be removed from the queue and discarded. Range 100…100000.
initialStartupDelayInSeconds	4	30	Delay after program startup before the configuration will start to be retrieved or any reports will be sent. This delay is required for the GSM modem to establish communication with the base station. Range 0..600.

Example:

Version 3.0 (only relevant entries have been presented):

```
OutStationName=A Detector
OutStationMode=3
settingsCollectionInterval1=120
settingsCollectionInterval2=30
urlCongestionReporting=http://37.152.43.178/BlueTruthReceiver1_50/Occupancy
urlJourneyTimesReporting=http://37.152.43.178:80/BlueTruthReceiver1_50/DeviceDetection
urlAlertAndStatusReports=http://37.152.43.178/BlueTruthReceiver1_50/Message
inquiryCyclePeriod=10
freeFlowSpeedCyclesThreshold=3
moderateSpeedCyclesThreshold=6
slowSpeedCyclesThreshold=9
verySlowSpeedCyclesThreshold=20
staticSpeedCyclesThreshold=50
btMACBinDropOutScanCycle = 100;
queueAlertThresholdBin=Slow
```

```
queueDetectThreshold=10
queueClearanceThresholdDetectionNumber=5
httpTimeOut=10
```

Version 4.0 (only relevant entries have been presented):

```
OutStationName=A Detector
OutStationMode=3
settingsCollectionInterval1=120
settingsCollectionInterval2=30
urlCongestionReports=http://37.152.43.178/BlueTruthReceiver2_00/Occupancy
urlStatusReports=http://37.152.43.178/BlueTruthReceiver2_00/Status
urlFaultReports=http://37.152.43.178/BlueTruthReceiver2_00/Faults
urlStatisticsReports=http://37.152.43.178/BlueTruthReceiver2_00/Statistics
inquiryCycleDurationInSeconds=10
statisticsReportPeriodInSeconds=60
congestionReportPeriodInSeconds=60
statusReportPeriodInSeconds=600
backgroundLatchTimeThresholdInSeconds=3000
backgroundClearanceTimeThresholdInSeconds=3000
freeFlowBinThresholdInSeconds=30
moderateFlowBinThresholdInSeconds=60
slowFlowBinThresholdInSeconds=90
verySlowFlowBinThresholdInSeconds=200
staticFlowBinThresholdInSeconds=500
absenceThresholdInSeconds  = 100;
queueAlertThresholdBin=Slow
queueDetectThreshold=10
queueClearanceThreshold=5
httpTimeOut=10
signReports=1
```

On system startup the last saved configuration will be used. The BlueTruth application will regularly try to retrieve configuration (see settingsCollectionInterval1, settingsCollectionInterval2 parameters) with the first attempt immediately after program start-up.

5.3 Seed File

This file will be stored in */var/cache/bt* directory as *seed.xml*.

The file will be created by the InStation and uploaded to each OutStation over SSH connection. The contents of this file will be used to protect InStations from spoofing attack and has been described in section 3.1. The file will be of xml format with the following schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Seed">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ID" type="xs:unsignedInt"/>
                <xs:element name="Value" type="xs:unsignedInt"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Seed>
    <ID>9876</ID>
    <Value>1234</Value>
</Seed>
```

6 Implementation

6.1 Operating System and Main Application

The BlueTruth will be installed on a Linux PC with Windriver based distribution.

The main program *outstation* name will be installed in */opt/bt/bin* directory. The program will be run as service on system start-up.

The following libraries will be required for the program:

- boost version 1.42+.
- openssl version 1.0.1e.
- libssh2 version 1.4.3.

6.2 Configuration

The core configuration file will be stored as *core_configuration.xml* in */opt/bt/etc* directory.

The schema for the core configuration file will be stored as *core_configuration.xsd* in */opt/bt/etc* directory.

The default functional file will be stored as *functional_configuration.conf* in */opt/bt/etc* directory.

The last downloaded functional file will be stored as *last_functional_configuration.conf* in */var/cache/bt* directory.

The seed file will be stored as *seed.xml* in */var/cache/bt* directory.

6.3 Certificates

The certificate files *to_instation_key* (private), *to_outstation_key.pub* (public) to be used for ssh reversed connection will be stored in two files in */var/cache/bt* directory. The certificate *generic_outstation_to_instation_key* (public) will be stored in */opt/bt/etc* directory. For connection to the InStation *to_instation_key* will be used if found, otherwise *generic_outstation_to_instation_key* will be used.

6.4 Program Exit Status Codes

The BlueTruth program may need to exit in some circumstances. The following exit status codes have been defined:
0 - normal program exit,
1 - fatal error (e.g. lack of configuration),
16 - program restart required (after a significant configuration change),
17 - system restart required (after REBOOT command).

7 Other Aspects

7.1 SSH Communication

To allow maintenance the communication over SSH channel will be enabled.
To bypass firewall restrictions the functionality of a reverse connection will be provided. Upon receipt of OPEN SSH CONNECTION response message (3.3.2) the OutStation will open a reverse SSH connection to the InStation. Than the engineer may connect from the server to the OutStation and perform maintenance activities or files (seed.xml, to_instation_key, to_outstation_key.pub) can be uploaded. The parameters of the connection from the OutStation to the InStation will be contained in the core configuration file (see 5.1).
The command used to open the reversed SSH connection will be:

- If the password specified in the core configuration file is blank
  ssh -t -t -i private_key_file_name –R<rep>:localhost:22 <login>@<address>
- If the pasword specified in the core configuration is not blank
  sshpass -p <password> ssh -t -t –R<rep>:localhost:22 <login>@<address>

Where <rep> is the remote entry point reported in the OPEN_SSH_CONNECTION, <login>, <address> and <password> are values read from the core configuration file.

The SSH authentication between an InStation and OutStations will be based on private/public key sharing. The OutStation will be initially deployed with a generic private OutStation/InStation keys which will be replaced by the new ones generated and managed by the InStation. The generic keys will be used only during the very first connection to the InStation and will be replaced by a newly generated key afterward by the InStation through the reverse tunnel by means of *scp*. The sequence to replace the generic keys with the new ones will be:

- The OutStation connects to the InStation and sends status command with field *key=0*,
- The InStation responds with *openConnection:NNN* where *NNN* is a remote port number (see 3.3.2),
- The OutStation opens SSH tunnel and responds with STATUS REPORT containing field *ssh=open xx.xx.xx.xx:yy NNN* where *xx.xx.xx.xx* is IP address of the InStation, *yy* is the port (default value 22), *NNN* is the remote port number,
- The InStation copies the new certificate files to the OutStation by means of scp and sends a reboot request in the response to STATUS REPORT,
- The OutStation responds with STATUS REPORT with field *boot=1* and reboots then.

7.2 Logging

The system will log all significant events into the log directory */var/log/bt*. The space in the log directory will be managed by a dedicated script provided with the operating system image.

7.3 GSM Modem

A GSM modem will be used to connect with the InStation. To provide metrics about quality of connection the OutStation application (*outstation*) will maintain connection to it (via telnet or SSH protocol). The GSM signal level will be periodically retrieved (see *functional_configuration / gsmModemSignalLevelSamplingPeriodInSeconds*), averaged (see *functional_configuration / gsmModemSignalLevelStatisticsWindowInSeconds*) and reported to the InStation in the form of STATUS REPORT messages.

7.4 Time Synchronisation (NTP)

The time accuracy will be of crucial importance when calculating the Average Travel Time between other OutStations. A NTP client will be run by the operating system at regular intervals (T_{NTP}) to increase time accuracy of the OutStation.

Document Control

Title: Functional Specification for BlueTruth OutStation

Document No: SSL/4012/B/10.02.2015

SSL Project Number:	87/1/0
Product Number:	564, 570

Status:	Definitive
File:	SSL4012B.doc

Synopsis:
This document provides detailed description of the functional capabilities of BlueTruth OutStation

Author(s) of this Issue of the Document:	Radoslaw Golebiewski

	Printed Name	Signature
Authorisation:	Louis Thompson	

	Printed Name	Signature
Doc. Review:	William Clifford	

Circulation List:

1. Internal SSL

All other copies are Uncontrolled

Change Control: Document must be reviewed by someone other than the Author(s).

Retention Period: This issue should be held on file until 02/2025.

Changes History:

Issue:	Date:	Changes:
A	21/01/2014	Original Issue
B	10/02/2015	Changes to increase consistency with Low-Power BlueTruth OutStation: - new parameters for functional configuration have been added, - NTP support by the OutStation application removed (provided by the Operating System). - Statistics Report can be sent as blank – with no device data.