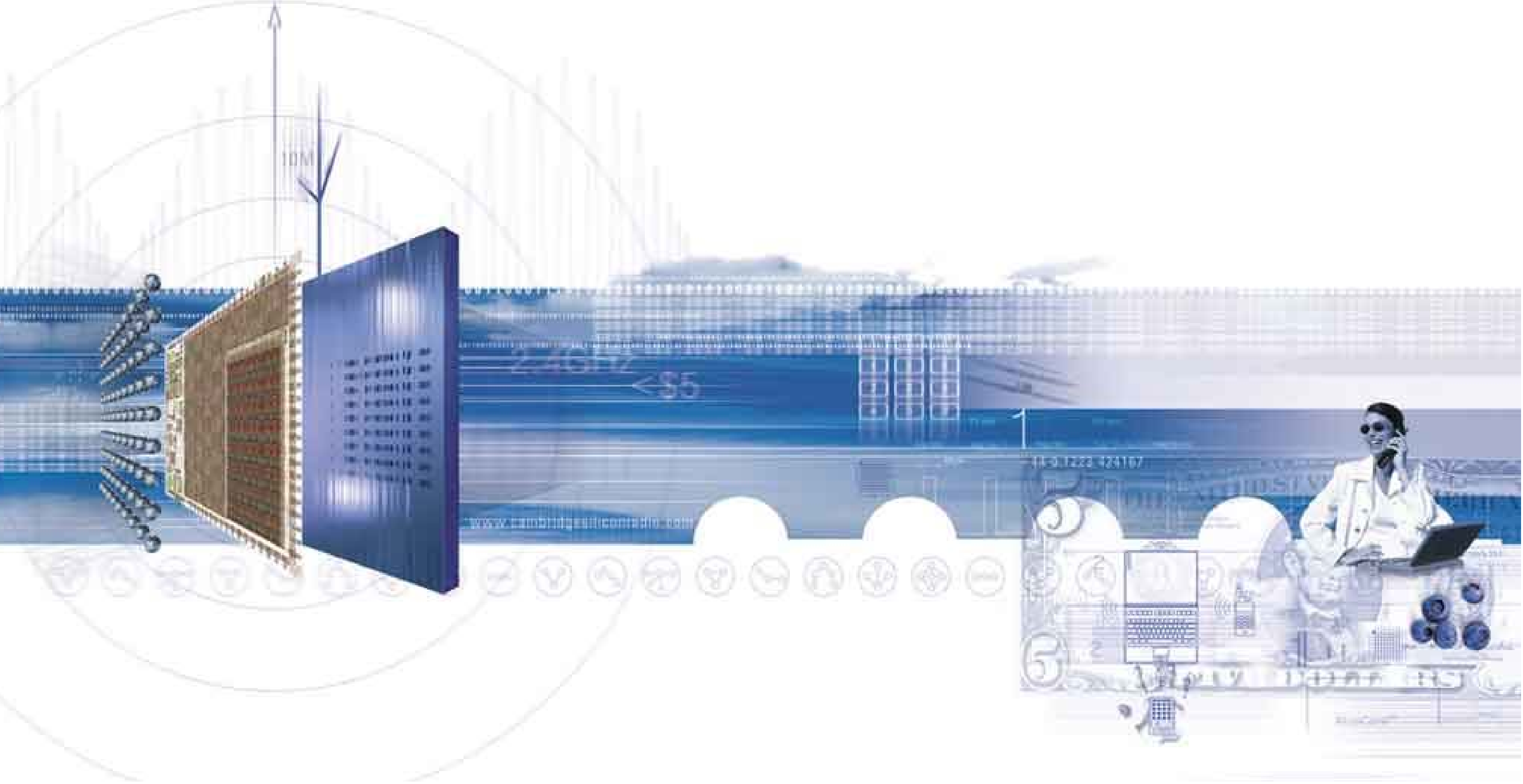
BlueCore™

BCSP Link Establishment Protocol

July 2004



CSR

Cambridge Science Park
Milton Road
Cambridge
CB4 0WH
United Kingdom

Registered in England 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Contents

List of Figures

List of Tables

1 Introduction

Document [BCSP] defines the BlueCore Serial Protocol (BCSP). That specification omits any description of how two BCSP stacks can initially synchronise with each other. This document describes a mechanism for performing initial link establishment.

When a system containing a pair of BCSP stacks is started, most traffic types are prevented from flowing to and from the peer; only one unreliable datagram channel is left open.

This document describes an entity that is layered above BCSP. When the system starts, the entity contacts its peer via this unreliable datagram channel through BCSP. When it is satisfied that the peer is operating correctly it removes the block from the local BCSP stack and so allows all traffic types to flow.

This mechanism reduces the probability of transmitting and receiving corrupt packets as the pair of systems is initialising.

The mechanism allows the local device to detect that the peer has restarted the link establishment protocol, e.g., because the peer has (crashed and) restarted.

The mechanism also makes predictable the types of packets initially flowing on the UART link, a characteristic that may be of value.

2 Context and Overview

The BCSP stack defined in [BCSP] is used to control and format information that flows between a Bluetooth Host and a Bluetooth Host Controller (BlueCore chip). The stack carries a set of parallel information flows between the two devices, multiplexing them over a single UART link.

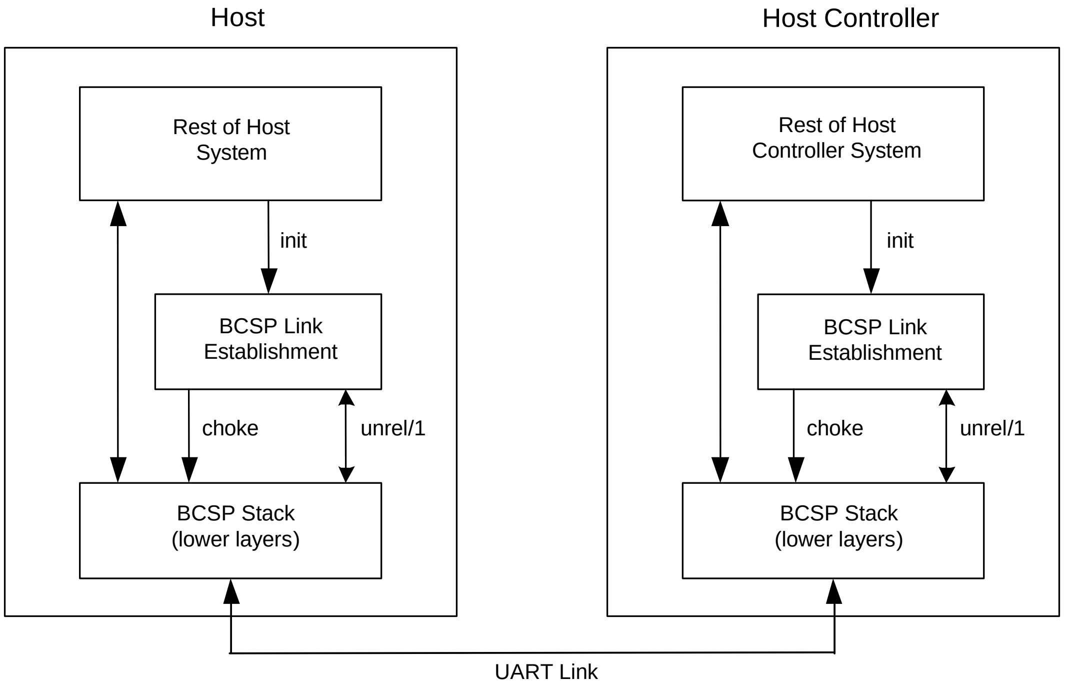


Figure 2.1: BCSP Link Establishment Context

The entity described in this document, "BCSP Link Establishment" (illustrated in Figure 2.1), receives an initialisation call from the rest of its local system. This init call will occur at the same time that the lower layers of the BCSP stack also receive initialisation calls.

After initialisation, the BCSP Link Establishment entity asserts the Boolean control signal "choke". This prevents the flow of all traffic types through the lower layers of the BCSP stack except unreliable packets with protocol identifier 1. The BCSP Link Establishment entity makes exclusive use of this channel ("unrel/1"). Protocol identifiers and use of channels are described in more detail in [BCSPCHAN]

The two BCSP Link Establishment entities send messages to each other, each expecting replies from its peer. When an entity receives the correct replies, it removes its local "choke" signal, allowing all BCSP stack traffic to flow.

3 BCSP Link Establishment

The BCSP Link Establishment entity is layered above the lower layers of the BCSP stack. The state machine depicted in Figure 3.1 describes the behaviour of the entity.

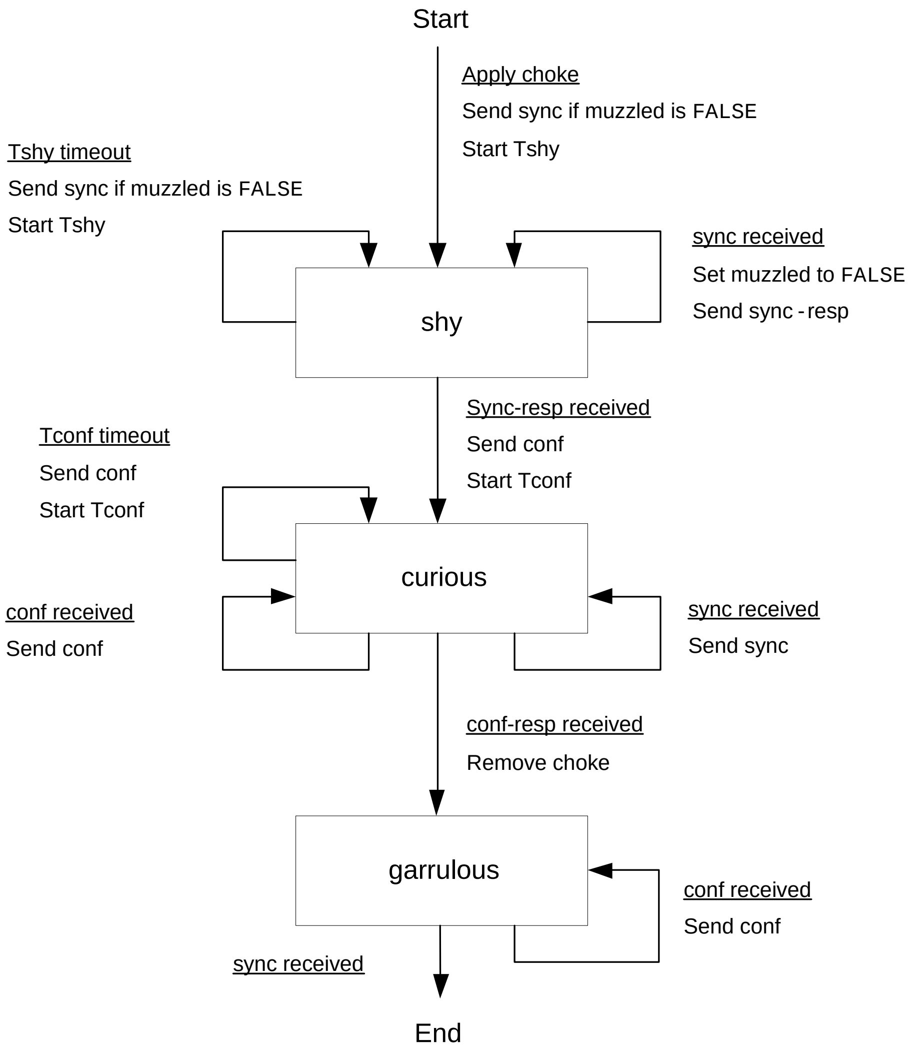


Figure 3.1: BCSP Link Establishment State Machine

The state machine uses the configuration control "muzzled". This prevents the local state machine from sending any messages to its peer until it has received at least one message from the peer.

Normal Behaviour

Assume initially that the Boolean variable "muzzled" is FALSE.

The two BCSP Link Establishment entities send "sync" messages to each other at regular intervals (Tshy).

When an entity receives a "sync" message it sends a "sync-resp" response message back to the peer.

When an entity receives a "sync-resp" message (a response to its own "sync" message) it takes this as confirmation that the peer is operating correctly, and it moves to the "curious" state.

In the "curious" state an entity sends "conf" messages to its peer at regular intervals (Tconf). The peer will only reply to these when it is in its "curious" (or "garrulous") state, so the peer will have assured itself that the local machine is operating correctly.

When an entity receives a "conf-resp" message it is sure the remote device knows that the local device is operating correctly, and so it removes the blockage of BCSP traffic (choke).

Normal BCSP traffic will not flow through both stacks until both have removed their blockages.

When both blockages have been removed the entities' state machines become dormant.

Passive Start

If the “muzzled” configuration variable is TRUE when the state machine is started then it starts to emit “sync” messages only after receiving a “sync” message from the peer. This holds the local Link Establishment entity silent until it has seen a spark of life from the peer. Obviously, only one of the pair of Link Establishment entities can be configured in this manner.

Restart Detection

If the state machine is in state “garrulous”, it knows that it should not receive a “sync” message from the peer. In practice the only reason for receiving a “sync” message is if the peer has restarted, so this gives the local state machine a means of detecting a reboot of the peer.

Table 3.1 may clarify the peer restart detection mechanism.

State	Activity
shy	Regularly emits “sync” messages. Replies to all “sync” messages with “sync-resp” messages. Moves to state “curious” on receipt of a “sync-resp” message.
curious	Regularly emits “conf” messages. Replies to all “sync” messages with “sync-resp” messages. Replies to all “conf” messages with “conf-resp” messages. Moves to state “garrulous” on receipt of a “conf-resp” message.
garrulous	Replies to all “conf” messages with “conf-resp” messages. Terminates on receipt of a “sync” message.

Table 3.1: Peer Restart Detection

Once the local entity has received a “conf-resp” message it knows the peer cannot be in state “shy”, so it knows that it cannot send a “sync” message (unless the peer has rebooted).

Note:

> Use of the “muzzled” option prevents the local state machine emitting “sync” messages at boot, so this can prevent the peer from recognising that the local state machine has rebooted.

Configuration Option

Zero or one of the Link Establishment entities on a BCSP link can have the “muzzled” configuration initially set to TRUE.

4 Packet Types

The BCSP Link Establishment entity uses four packet types: sync, sync-resp, conf and conf-resp.

All of these packets are simply tokens, and so can be of arbitrary values. Each packet’s value is a sequence of four bytes.

The packet length helps combat the possibility of data corruption.

The length also helps distinguish the packets as the start of BCSP traffic.

Table 4.1 lists the packets’ (almost arbitrary) byte sequences (listed in transmission order).

Message	Message Byte Sequence
`sync`	`{0xda, 0xdc, 0xed, 0xed}`
`sync-resp`	`{0xac, 0xaf, 0xef, 0xee}`
`conf`	`{0xad, 0xef, 0xac, 0xed}`
`conf-resp`	`{0xde, 0xad, 0xd0, 0xd0}`

Table 4.1: Link Establishment Packet Types

Document References

Document ID	Document Title	CSR Reference
[BCSP]	BlueCore Serial Protocol (BCSP)	bcore-sp-012P
[BCSPCHAN]	BCSP Channel Allocation	bcore-sp-007P

Acronyms and Definitions

BCSP	BlueCore Serial Protocol
BlueCore™	Group term for CSR's range of Bluetooth chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
HCI	Host Controller Interface
UART	Universal Asynchronous Receiver Transmitter

Record of Changes

Date	Revision	Comment
8 Jan 03	a	Document originally published as CSR reference bc01-an-005 (revisions a through b, versions through HCIStack1.1v15.x builds). New revision control number allocated to align with HCIStack1.1v16.x builds.
15 Jul 04	b	Updated formatting, corrected typos.

BlueCore™ BCSP Link Establishment Protocol

BCSP Link Establishment Protocol

bcore-sp-008Pb

July 2004

Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG Inc, USA and are licensed to CSR.

BlueCore™ is a trademark of CSR.

All other product, service and company names are trademarks, registered trademarks or service marks of their respective owners.

CSR's products are not authorised for use in life-support or safety-critical applications.