University of York

The Department of Computer Science

Embedded Systems Design & Implementation Development Report

Prepared by

ID: Y6332150

November 2013

1. Key functionality of the relay node

The goal of the relay node is to synchronise with the sink and forward as many frames as possible from the sources. The strategy chosen was based around a concept of priority. Combined with the need to switch channels of the radio, led to a simple stack based system (Relay, pushSession/popSession, 427-437/400-421), where each element (Session, Session, 39-43) on the stack corresponds to a period allocated for a specific channel at a specific time. This stack also enforces the priority of the channels, with the sink being highest priority and the sources being prioritised in decreasing order of their period (Relay, onChannelTimer, 444-457).

The stack based relay operates in two distinct modes, the first being the discovery period[1], during which the exact timing of each channel is determined. During the discovery phase, each channel is visited in sequence, starting with the sink. The channel is only switched when either the periodic event is received (Relay, onSinkReceive/onSourceReceive, 212/301), allowing the timing to be determined, or when a channel, whose timing has already been determined, undergoes its' periodic event cycle (Relay, onChannelTimer, 444-457). The second phase makes use of the timings determined in the first phase, being awakened by timers (Relay, onChannelTimer/onScheduleTransmit, 444/357) in order to receive or send frames.

All received frames are stored in a circular frame buffer (Relay, onSourceReceive, 322-323), which has a predefined capacity of 5 (Relay, frameBuffer, 79), which offers some redundancy, even if one sending phase is missed. During the sending phase (Relay, onScheduleTransmit, 362) as many frames as possible are sent, either until the buffer is emptied (Relay, transmitFromBuffer, 368) or the window for transmission has closed (Relay, onTransmit, 339).

In order to allow for some laxity in the timings of the system, all calculations use a buffer value (Relay, TIMING_BUFFER, 42) to either schedule timers early (Relay, onSinkReceive/onSourceReceive, 259-269/310) or make sure the process is ended early (Relay, onScheduleTransmit, 359). In addition, calculating periods and time differences is based on the timestamp given by the radio and not on the current time of the callback, allowing for lateness in the callbacks (Relay, onSinkReceive/ onSourceReceive, 238/310).

---

[1] Appendix A includes a UML diagram describing an example of how the relay node behaves during the discovery period

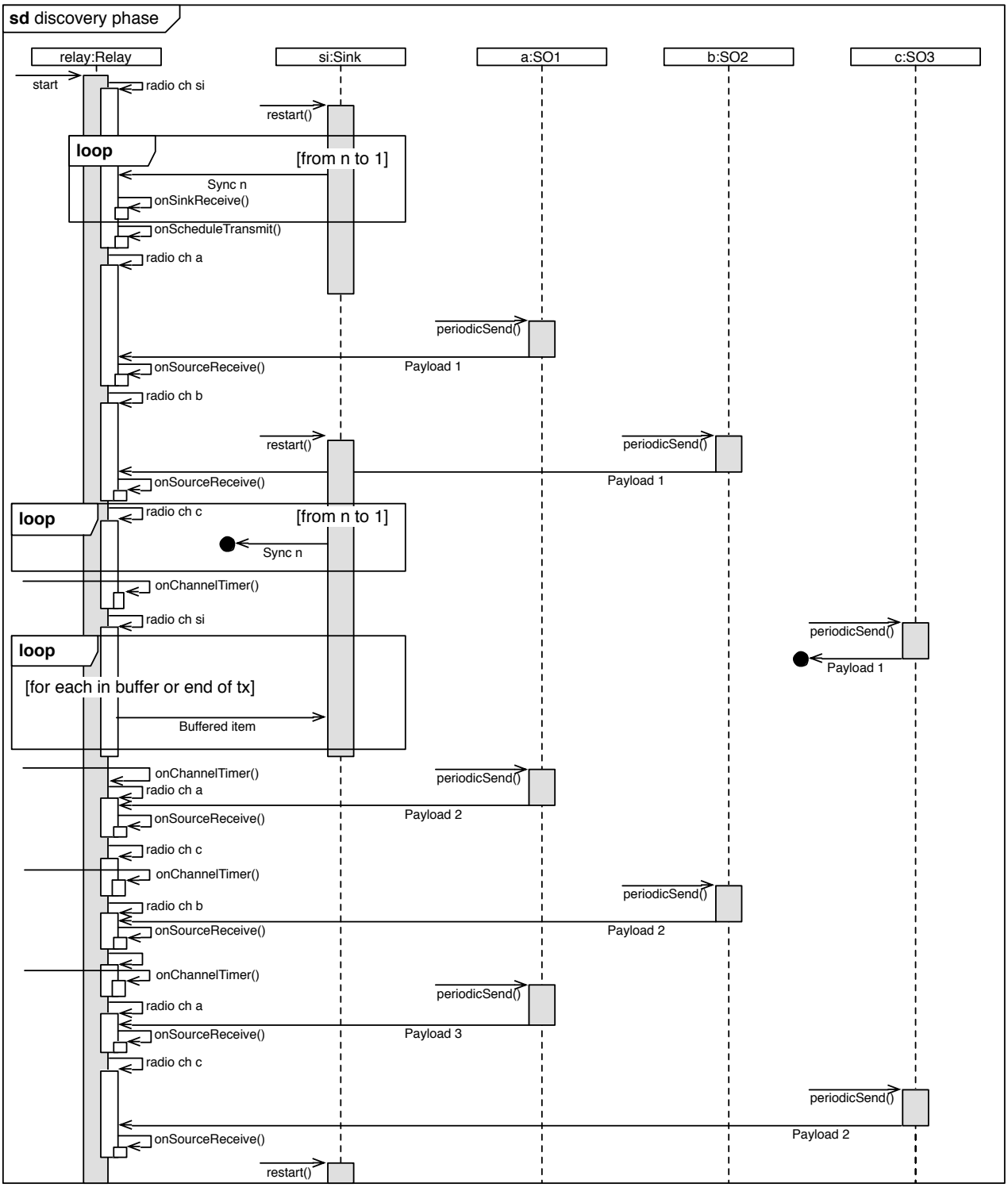
2. Energy efficiency

After the relay has determined the timings of all channels, the system stack will only contain one indefinite session for CHANNEL_OFF (Relay, static, 166). This special constant is processed by the system in a way that simply stops the radio reception and thus allows the radio to fall asleep (Relay, setChannel, 481).

In the second phase, the radio is periodically re-enabled for a short period of time during the periodic source and sink node events on the network. Waking the system is handled by timers (Relay, onChannelTimer, 444), allowing the system to sleep in between the events.

Finally, the signal strength for transmission of frames to the sink is adjusted to reduce energy consumption. The strength is calculated during the sync phase, based on the RSSI from the sink (Relay, onSinkReceive, 282-284). This saves energy, but reduces flexibility, as the sink is not expected to move during the second phase without risking losing contact with the relay.

3. Appendix A


sd discovery phase
relay:Relay
si:Sink
a:SO1
b:SO2
c:SO3
start
radio ch si
restart()
loop
[from n to 1]
Sync n
onSinkReceive()
onScheduleTransmit()
radio ch a
periodicSend()
onSourceReceive()
Payload 1
radio ch b
restart()
periodicSend()
onSourceReceive()
Payload 1
loop
radio ch c
[from n to 1]
Sync n
onChannelTimer()
radio ch si
periodicSend()
Payload 1
loop
[for each in buffer or end of tx]
Buffered item
onChannelTimer()
periodicSend()
radio ch a
Payload 2
onSourceReceive()
radio ch c
onChannelTimer()
periodicSend()
radio ch b
Payload 2
onSourceReceive()
onChannelTimer()
periodicSend()
radio ch a
Payload 3
onSourceReceive()
radio ch c
periodicSend()
Payload 2
onSourceReceive()
restart()