Introduction

µTasker is an operating system designed especially for embedded applications where a tight control over resources is desired along with a high level of user comfort to produce efficient and highly deterministic code.

The operating system is integrated with TCP/IP stack and important embedded Internet services along side device drivers and system specific project resources.

µTasker and its environment are essentially not hardware specific and can thus be moved between processor platforms with great ease and efficiency.

However the µTasker project setups are very hardware specific since they offer an optimal pre-defined (or a choice of pre-defined) configurations, taking it out of the league of "board support packages (BSP)" to a complete "project support package (PSP)", a feature enabling projects to be greatly accelerated.

This document describes the DHCP protocol and its implementation.

DHCP

DHCP (Dynamic Host Configuration Protocol – RFC 2131) is used for automatic configuration of network clients in which case one or more DHCP servers are configured to supply the correct parameters for a given network.

The clients first locate the server they want to use and then request the necessary parameters, either suggesting preferred ones or letting the DHCP server offer them.

It is to be noted that the use of DHCP imposes a constraint on the minimum Ethernet buffer size in order to be able to receive all possible DHCP messages, which can have a length of up to 576 bytes (longer message size can be negotiated): 44 bytes header, 64 bytes for server host name, 128 bytes boot file name and 312 bytes vendor specific information. In small foot print systems the length of the Ethernet buffers can often be programmed to be as small as 128 bytes and up to 1,5k, adequate to receive the maximum Ethernet frame of 1514 bytes. TCP for example supports capabilities of operating with small buffer sizes without restrictions – apart from throughput issues – allowing suitable compromises in such systems to enable tradeoffs between minimum RAM resources and performance. DHCP doesn't support a suitable mechanism and so the buffer size must be set adequately.

The DHCP client operation is started by an application task by calling:

```
extern int fnStartDHCP(signed char cTask)
```

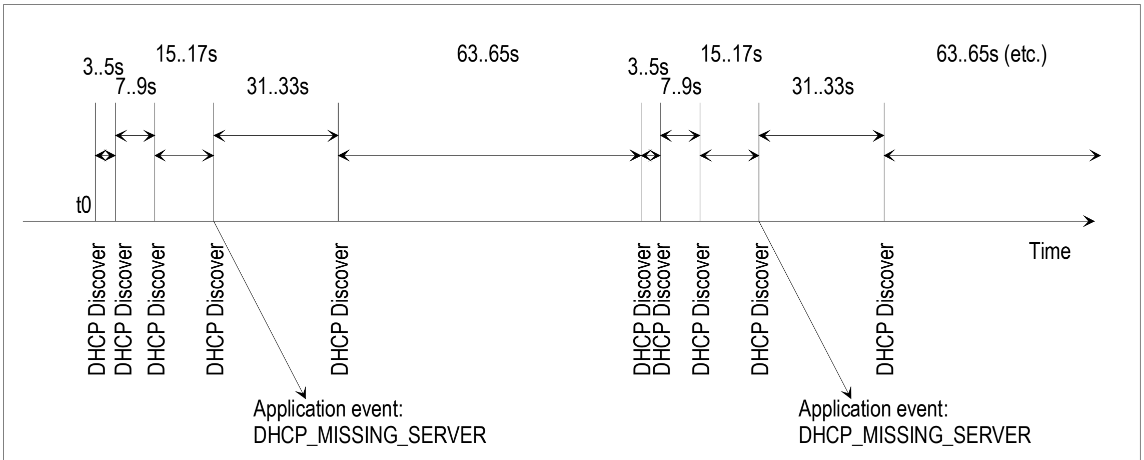
The application task name is passed so that DHCP events can be passed to this task. Examples of such events are DHCP_SUCCESSFUL [DHCP negotiation has been successfully terminated and network operation can begin/continue] and DHCP_LEASE_TERMINATED [DHCP lease has expired without being able to renegotiate lease extension – all network operation should terminate].

DHCP message timeouts

DHCP messages are initially broadcast in nature and after a DHCP server has leased a set of parameters they become unicast with the specific DHCP server. Requests and responses are matched using a transaction ID (xid) which is a random 32bit number generated by the client and transmission are retried after defined delays should a response not be received.

As specified in RFC 2131, the DHCP client is responsible for all retransmissions and uses an initial 4s repetition time with +/-1s randomisation before switching to an exponential back-off up to a maximum of 64s.

The µTasker DHCP client will send its initial discovery message with the following periodicity when there is no DHCP server responding to it:



The task which started the DHCP operation receives an event each time that the repetition time increases to 32s (+/- 1s) and can decide to disable DHCP and switch to a default set of parameters since it is probable that no DHCP server exists in the network to perform negotiation with. In this case the DHCP operation can be stopped by calling

```
extern void fnStopDHCP(void)
```

and restarted at any time using the start call again. Note that the local IP address is set on calling the stop command to the value contained in the preferred DHCP location. This allows the system to continue using its preferred address as default if no DHCP server was found. During DHCP negotiation the local IP address is set to 0.0.0.0 to indicate that network operation is not yet permitted.

DHCP Implementation

The DHCP protocol is realised as a state event machine by the µTasker, as described in RFC 2131.

The BOOTP broadcast flag is always set and so all messages from the DHCP server during negotiation will be broadcast types (and not unicast).

As soon as a DHCP server makes an offer in response to a discovery message from the DHCP client, the client will request these lease parameters.

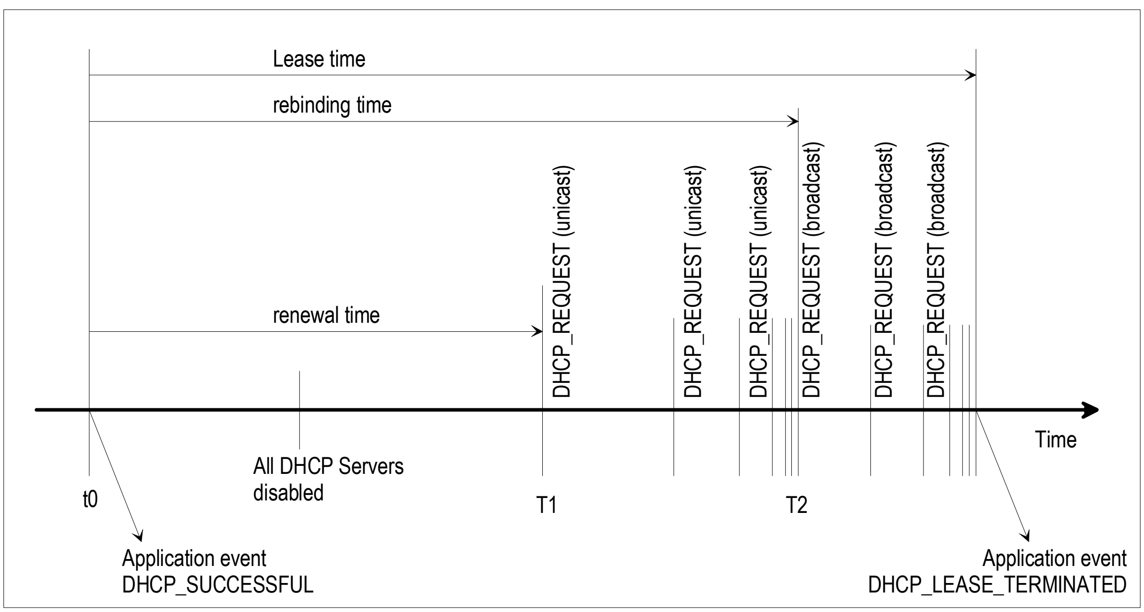
Lease timeouts are controlled by three timeout values returned by the DHCP server during negotiation. These are:

Lease time – the total lease time which has been given (0xffffffff is infinite)

Renewal time – the time after which the renewal process should be started (generally half of the least time). The DHCP client should request an extension of the lease from the DHCP server which has given the lease. This is referred to time T1.

Rebinding time – the time after which new parameters should be requested from any DHCP server (generally 0,875 times the lease duration. This is referred to as time T2 and is only used when the lease could not be extended after T1.

If a lease times out because it could neither be extended nor a new lease be negotiated, the DHCP client no longer has the right to use the leased parameters and is no longer network capable.



This diagram illustrates the operation of the renewal/rebinding process when the leasing server and all other DHCP servers on the network are deactivated after the original lease parameters have been obtained.

Up to the time T1 there is no activity and the DHCP client may use the leased parameters. At T1 the DHCP client tries to renew the lease. In the normal case the DHCP server would still exist and know the DHCP client and renew the release which would effectively restart at t0 (although the application is not informed after a renewal). However since the DHCP server which leased the parameters is no longer present there is no answer. The DHCP client retries again after the remaining time to T2 has halved and thereafter with increasing frequency (halving remaining time) until the remaining time has decreased to less than 60s, in which case no more repeats are made.

At T2 the rebinding period starts where the DHCP request is no longer sent unicast to the original leasing server but is sent as a broadcast to allow any DHCP server in the network to respond. The retransmission period is half the remaining time until the end of the lease and is halved until the retransmission timeout decreases to less than 60s, after which no further retransmissions are performed.

Once the lease time runs out without being able to rebind the DHCP device is no longer allowed to use the leased parameters. The application is informed via a DHCP_LEASE_TERMINATED event and can either switch to a default set of parameters or stop all network activity.