

Aculab Prosody™ 2 (TiNG) T.38 Gateway logging.

Revision 1.4



PROPRIETARY INFORMATION

The information contained in this document is the property of Aculab plc and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission. It should not be used for commercial purposes without prior agreement in writing.

All trademarks recognised and acknowledged.

Aculab plc endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission.

The development of Aculab's products and services is continuous and published information may not be up to date. It is important to check the current position with Aculab plc.

Copyright © Aculab plc. 2009-2013 all rights reserved.

Document Revision

Rev	Date	By	Detail
0.1	7-Dec-09	D.C	Draft for review.
1.0	18-Apr-11	D.C	Update.
1.1	24-May-11	D.C	Corrected 't38gwtypes' ref. Added T.38 Gateway introduction document ref.
1.2	08-July-11	D.C	Added fax spoofing for g57.
1.3	19-Aug-11	D.C	Added 32/64 for g58.
1.4	16 May-13	EBJ	Updated to corporate fonts and reformatted.

CONTENTS

1	Introduction.....	4
2	T.38 Gateway logging.....	5
2.1	T.38 Gateway logging function.....	5
2.2	T.38 Gateway logging function buffering requirement.....	6
2.3	T.38 Gateway log timestamp.....	6
2.4	T.38 Gateway logging levels.....	6
2.5	Additional T.38 Gateway logging options.....	7
3	T.38 Gateway Sessions.....	8
4	T.38 Gateway Paths.....	9
4.1	T.38 Gateway Job numbers.....	9
4.2	T.38 Gateway Events.....	9
5	T.38 Job start and configuration.....	10
5.1	T.38 Gateway Type 'A' (image to network).....	10
5.2	T.38 Gateway Type 'B' (image from network).....	11
5.3	T.38 Gateway VMP/FMP.....	13
6	Displaying fax sequence.....	14
7	T.38 Gateway FCF modification.....	17
7.1	Digital Information Signal modification.....	17
7.2	Non Standard Facilities modification.....	17
8	Data display.....	18
8.1	Message and Image carriers.....	18
8.2	V21.....	18
9	The Stop Sequence.....	20
10	Callback function.....	22
11	Fax spoofing.....	23
11.1	T.38 Gateway 'A' spoofing.....	23
11.2	T.38 Gateway 'B' spoofing.....	24
12	References.....	27

1 Introduction.

The T.38 Gateway logging functionality is not only a Aculab development tool but it is also intended to be used by customers during initial development and on actual deployment. This document provides a brief introduction to T.38 Gateway logging starting with implementation and then focusing upon T.38 Gateway configuration and diagnosis of the T.30 fax and T.38 Gateway stop sequence.

It should be read in conjunction with the Aculab T.38 Gateway API guide [2] and the Aculab T.38 Gateway introduction [3].

2 T.38 Gateway logging.

The Aculab T.38 Gateway includes a comprehensive logging facility that provides information on the status and progress of individual T.38 gateway jobs. The logging is configurable to one of three levels or can be turned off completely. Additional debug logging levels can also be enabled to allow diagnosis of individual site problems or conditions.

2.1 T.38 Gateway logging function.

The Aculab T.38 Gateway log data is outputted by a customer defined function accessed by the T.38 Gateway using the exported pointer to function variable 'T38GW_showtrace'.

The prototype for the customer defined logging function is :-

```
int fn(const char *fmt, va_list ap);
```

For example, to output a timestamp and then the standard trace:

```
int timedtrace(const char *fmt, va_list ap)
{
    DWORD tm = GetTickCount();
    printf("%u: ", tm);
    vprintf(fmt, ap);
    return 0;
}
```

The exported variable then needs to be loaded *once during the application initialisation* before any T.38 Gateway activity occurs.

```
T38GW_showtrace = timedtrace;
```

Care should be taken when implementing the logging function as it is called within the context of the Aculab T.38 Gateway and therefore should be implemented as efficiently as possible. Looping or other complex routines should be avoided.

Most systems involve simultaneous T.38 Gateway transfers resulting in multiple T.38 jobs writing to a common output. Each T.38 job is however, individually identified and although inter-spacing of T.38 job output may occur occasionally this does not create a problem even on systems running a large number of simultaneous T.38 Gateway transfers.

The use of a mutex or other thread locking mechanism in the logging function should be avoided as it may well have an effect upon operation of the T.38 Gateway at high usage levels.

Output can be stdio or file output, however if output is to a file system it should not be accessed across a network when large numbers of T.38 jobs are running simultaneously as at higher logging levels the amount of data written can be considerable.

Circular logging to a fixed number of output files can be used to limit the amount of storage required for extended test runs or actual deployments. This type of logging scheme can be implemented without mutex locks by ensuring:-

1. that the logging function uses a local copy of the file handle loaded with an atomic function.
2. code used to move the file handle on to the next file is atomic.
3. A file is not closed whilst a T.38 Gateway log is in progress just after a transfer of the file handle.

2.2 T.38 Gateway logging function buffering requirement.

Temporary buffers allocated to hold data from the logging function require to be at least 6180 bytes long to allow for the longest possible data that can be outputted.

2.3 T.38 Gateway log timestamp.

Some form of timestamp output in the logging function is essential for analysis of T.38 Gateway log traces. The most useful form of this has found to be Seconds/ mS as T.30 fax timeouts are in the range 3 – 4 Seconds.

2.4 T.38 Gateway logging levels.

The Aculab T.38 Gateway logging level is set using the exported variable **T38Gwtrace** by 'oring' in a selection of debugging options. All logging options and levels operate on both T.38 Gateway data paths.

Normally the T.38 Gateway logging level should be set to one of the standard levels:-

T38GW_TRACE_LVL_LOW

T38GW_TRACE_LVL_MED

T38GW_TRACE_LVL_HIGH

For most applications setting the log level to high will provide adequate logging output.

```
T38Gwtrace = (T38GW_TRACE_LVL_HIGH);
```

Aculab technical support may request that additional logging options are included to provide more information for support issues.

2.5 Additional T.38 Gateway logging options.

The Aculab T.38 Gateway transfers both v21 and Fast (Image) data that can optionally be displayed in the T.38 Gateway log. This however, results in an increase in log size and increased T.38 Gateway application program loading required for the additional logging.

There are two additional data logging options:-

T38GW_TRACELVL_INDATA *log all input data on both t38 gateway paths.*

T38GW_TRACELVL_OUTDATA *log all output data on both t38 gateway paths.*

Normally the input data will be the same as the output data with the only exceptions to this being the two v21 FCF messages **NSF** (*Non Standard Facilities*) where the payload is cleared by the T.38 Gateway before transfer and **DIS** (*Digital Signature*) where the T.38 Gateway *may* alter the message payload to conform to customer set options. Input data, output data or both can be set using:-

```
T38GWtrace = (T38GW_TRACELVL_HIGH | T38GW_TRACELVL_INDATA);
```

```
T38GWtrace = (T38GW_TRACELVL_HIGH | T38GW_TRACELVL_OUTDATA);
```

```
T38GWtrace = (T38GW_TRACELVL_HIGH | T38GW_TRACELVL_OUTDATA |
               T38GW_TRACELVL_INDATA);
```

Data displayed in the T.38 log is bit reversed and will need to be converted before use.

Other logging options include,

T38GW_TRACELVL_TX *Transmit sequence.*

T38GW_TRACELVL_INFOHDL *HDLC framing.*

T38GW_TRACELVL_INFOCARRIER *Carrier activity.*

All logging options are defined in the header file '**t38gwtypes.h**' however most of these are used for development and support purposes.

3 T.38 Gateway Sessions.

Applications must create at least one T.38 Gateway session [3] that must exist for the duration of the T.38 Gateway activity. Each session can have 0 –6 T.38 Gateway jobs associated with it but applications may however choose to run only one T.38 job per session.

```
26:14:56:06:665 CREATE SESSION
```

```
26:14:57:40:774 (0x386bc0*) STOP SESSION
```

```
26:14:57:40:790 (0x386bc0*) DESTROY SESSION
```

The stop and destroy session log entries display a unique session identifier.

4 T.38 Gateway Paths.

The Aculab T.38 Gateway comprises two data paths path 'A' and path 'B'.

Path 'A' in both configurations of the Gateway, is *always* the data path therefore the CNG tone and the image data is always received from the path A input channel and transmitted using the path 'A' Tx channel.

Path 'B' in both configurations of the Gateway, is *always* the return path therefore the CED tone is always received from the path 'B' input channel and transmitted using the path 'B' Tx channel.

Both paths transfer T.30 v21 control messages.

The T38 Gateway logging displays path A as 0 and path B as 1 for example:-

```
15:42:32.829 (0x56afc0*) 1 DIS // DIS rcvd on path B.
15:42:35.470 (0x56afc0*) 0 DCS // DCS rcvd on path A.
```

4.1 T.38 Gateway Job numbers.

All T.38 Gateway jobs are allocated a unique job number that can be used to filter the output for individual jobs from log files that contain output from multiple concurrent T.38 jobs.

This job number is unique only during the life of a T.38 job and may then be re-used on subsequent jobs. So a filter on a specific job number on a large log file containing numerous T.38 jobs may produce a number of T.38 jobs that have been allocated this job number.

4.2 T.38 Gateway Events.

The T.38 Gateway is an event driven system with all processing instigated by events set by TiNG or by events internally set in the T.38 Gateway. The T.38 Gateway log displays tone, v21, fast data and stop events for both paths.

```
20:305 (0x386bf8) STATE A TONE EV
20:305 (0x386bf8*) 0 CNG

21:009 (0x386bf8) STATE B V21 EV
21:009 (0x386bf8) READ ff ff ff ff 00 00 00 00 05 00 00 00
21:009 (0x386bf8) Flags (1)

33:915 (0x386bf8) STATE A FAST EV
33:915 (0x386bf8) READ 40 07 40 07 c0 96 08 b0 b1 44 00
```

Generally the direction through the T.38 Gateway can be deduced by the event and or the path number in the prompt (0 is path A, 1 is path B). All image (fast) data is transferred on path A only.

5 T.38 Job start and configuration.

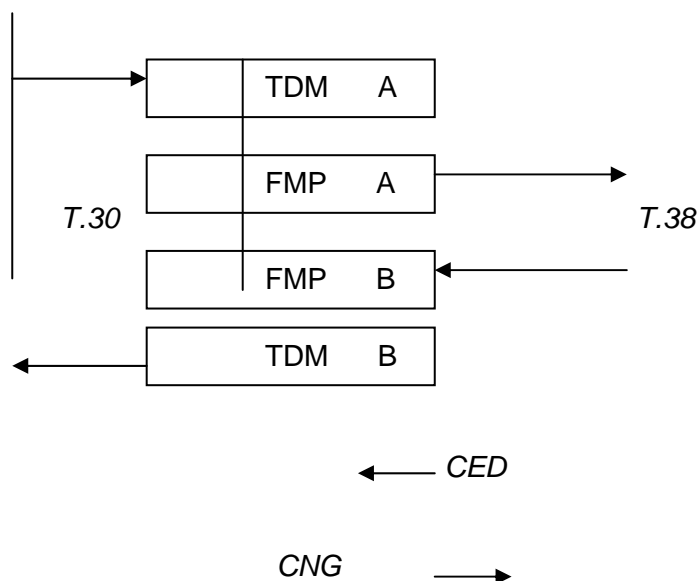
At present the T.38 Gateway has two legal configurations, Configuration 'A' (*Image to Network*) and Configuration 'B' (*Image from Network*) these are displayed in the T.38 Gateway log when a job is created along with certain other configuration information.

5.1 T.38 Gateway Type 'A' (image to network).

The example shown uses TDM/FMP.

```
(0x8077058*) CREATE JOB (modems= 0x7)
(0x8077058*) (g56+m1382.) ECM allow   ASN1=0
(0x8077058*) PATH A RX EP = T30TDM
(0x8077058*) PATH A TX EP = T38FMP
(0x8077058*) PATH B RX EP = T38FMP
(0x8077058*) PATH B TX EP = T30TDM
(0x8077058) Tx channel      path A (0x80774b0)
(0x8077058) Msg channel     path A (0x8077d00)
(0x8077058) Img channel     path A (0x80778b0)
(0x8077058) Tx channel      path B (0x8078150) -18
(0x8077058) t38 input channel path B (0x80785e0)
(0x8077058) rxstate = 0: newstate = 2
(0x8077058) chan - conrx (0) (c=8077d00).
(0x8077058) carriers set off 1, path 0
(0x8077058) rxstate = 1: newstate = 9
(0x8077058) chan - conrx (1) (c=80785e0).
```

TDM.



Other information included in the T.38 Gateway start trace are:-

modems

t30 modems allowed for this T.38 Gateway job (sm_t38gw_create_job [2]).

(100 = v17, 010 = v29, 001=v27)

(g56+m1382.)

T.38 Gateway software version number and the TiNG version the T.38 Gateway library was built against. The TiNG version may be the same or earlier than the one in use. 64 bit versions add (64).

ECM allow

ECM disable option for this T.38 Gateway job (sm_t38gw_create_job [2]).

ANS1

T.38 ANS1 version for this T.38 Gateway job (sm_t38gw_create_job[2]).

< 2 is Pre Corrigendum.

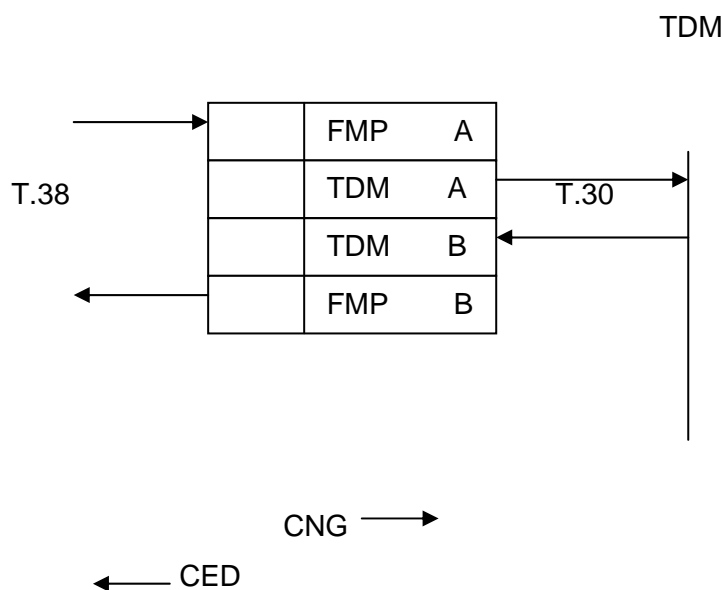
The channel identifiers are provided for use with *TiNGTRACE*, path 'A' the T30/T38 image path has one v21 (*Msg*) input channel and one Image (*Fast*) input channel plus one output (*Tx*) channel.

Path 'B' (*T38/T30*) handles v21 only and has one t38 input channel and one output (*Tx*) channel. Where appropriate the TDM fsk_tx_power setting is also displayed, in this case -18dBm0.

5.2 T.38 Gateway Type 'B' (image from network).

The example shown uses TDM/FMP.

```
(0x8077070*) CREATE JOB (modems= 0x7)
(0x8077070*) (g56+m1382.) ECM allow   ASN1=0
(0x8077070*) PATH A RX EP = T38FMP
(0x8077070*) PATH A TX EP = T30TDM
(0x8077070*) PATH B RX EP = T30TDM
(0x8077070*) PATH B TX EP = T38FMP
(0x8077070) Tx channel          path A (0x80774c8) -18
(0x8077070) t38 input channel path A (0x80778b8)
(0x8077070) Tx channel          path B (0x8077d08)
(0x8077070) Msg channel         path B (0x8078158)
(0x8077070) rxstate = 0: newstate = 2
```



Other information included in the T.38 Gateway start trace are:-

Modems

t30 modems allowed for this T.38 Gateway job (sm_t38gw_create_job [2]).
 (100 = v17, 010 = v29, 001=v27)

(g56+m1382.)

T.38 Gateway software version number and the TiNG version the T.38 Gateway library was built against. The TiNG version may be the same or earlier than the one in use.

ECM allow

ECM disable option for this T.38 Gateway job (sm_t38gw_create_job [2]).

ANS1

T.38 ANS1 version for this T.38 Gateway job (sm_t38gw_create_job[2]).
 < 2 is Pre Corrigendum.

The channel identifiers are provided for use with TiNGTRACE, path 'A' the T38/T30 image path has one t38 input channel that handles both v21 and Fast data plus one output (Tx) channel.

Path 'B' (T30/T38) handles v21 only and has one t30 input channel and one output (Tx) channel. Where appropriate the TDM fsk_tx_power setting is also displayed, in this case -18dBm0.

5.3 T.38 Gateway VMP/FMP.

Logging produced for both TDM and VMP options are almost identical. The only major difference is in the T.38 Gateway job start prompt. As for the TDM option the logging output can be used to check the configuration.

GWA Vmp/Fmp.

```
:07:051 (0x80773a8*) CREATE JOB (modems= 0x7)
:07:051 (0x80773a8*) (g56+m1382.) ECM allow ASN1=0
:07:051 (0x80773a8*) PATH A RX EP = T30VMP
:07:051 (0x80773a8*) PATH A TX EP = T38FMP
:07:052 (0x80773a8*) PATH B RX EP = T38FMP
:07:052 (0x80773a8*) PATH B TX EP = T30VMP
:07:052 (0x80773a8) Tx channel path A (0x8077800)
:07:052 (0x80773a8) Msg channel path A (0x8078090)
:07:052 (0x80773a8) Img channel path A (0x8077c00)
:07:052 (0x80773a8) Tx channel path B (0x80784c0)
:07:052 (0x80773a8) t38 input channel path B (0x8078910)
:07:052 (0x80773a8) rxstate = 0: newstate = 2
```

GWB Vmp/Fmp.

```
:14:528 (0x80773a8*) CREATE JOB (modems= 0x7)
:14:528 (0x80773a8*) (g56+m1382.) ECM allow ASN1=0
:14:528 (0x80773a8*) PATH A RX EP = T38FMP
:14:528 (0x80773a8*) PATH A TX EP = T30VMP
:14:528 (0x80773a8*) PATH B RX EP = T30VMP
:14:528 (0x80773a8*) PATH B TX EP = T38FMP
:14:528 (0x80773a8) Tx channel path A (0x8077800)
:14:528 (0x80773a8) t38 input channel path A (0x8077c00)
:14:528 (0x80773a8) Tx channel path B (0x8078090)
:14:528 (0x80773a8) Msg channel path B (0x80784c0)
:14:528 (0x80773a8) rxstate = 0: newstate = 2
```

6 Displaying fax sequence.

The T.38 Gateway logging inserts a “*” character at certain points in the output such as the start-up section and also when fax control (FCF) messages are received on either path. Filtering on the job number with an asterix (for example 0x56afc0*) will display the start up log and sequence of T.30 messages processed by the T.38 Gateway allowing the fax sequence to be displayed.

Example 1.

```
13:34:25.427 (0x80c6268*) CREATE JOB (modems= 0x7)
13:34:25.427 (0x80c6268*) (g56+m1382.) ECM allow ASN1=0
13:34:25.427 (0x80c6268*) PATH A RX EP = T30TDM
13:34:25.427 (0x80c6268*) PATH A TX EP = T38FMP
13:34:25.427 (0x80c6268*) PATH B RX EP = T38FMP
13:34:25.427 (0x80c6268*) PATH B TX EP = T30TDM )
13:34:25.526 (0x80c6268*) 0 CNG
13:34:26.236 (0x80c6268*) 1 CED
13:34:30.828 (0x80c6268*) 1 CSI
13:34:30.987 (0x80c6268*) 1 DIS
13:34:33.998 (0x80c6268*) 0 TSI
13:34:34.428 (0x80c6268*) 0 DCS
13:34:34.428 (0x80c6268*) V27 - 4800
13:34:37.120 (0x80c6268*) 0 TCF
13:34:39.701 (0x80c6268*) 1 CFR
13:36:36.500 (0x80c6268*) 0 PPS_NULL
13:36:39.410 (0x80c6268*) 1 MCF
13:38:36.50 (0x80c6268*) 0 PPS_NULL
13:38:38.932 (0x80c6268*) 1 MCF
13:39:45.377 (0x80c6268*) 0 PPS-EOP
13:39:48.278 (0x80c6268*) 1 MCF
13:39:49.897 (0x80c6268*) 0 DCN
13:39:50.63 (0x80c6268*) DESTROY JOB
```

T.38 Gateway type A (Image to network), ECM allowed (and used in this fax), v17,v29,v27 modems enabled (transfer used v27 4800), Pre-corrigendum ASN1, Fax transfer proceeded without a problem.

The fax transmitter is connected to the T.38 Gateway over the TDM bus, the fax receiver is connected via the network.

When the T.38 job has been terminated by a DCN or ABORT, it is deleted from the session job list. The job is destroyed when the application program calls the ‘sm_t38gw_destroy_job [2]’ api function.

Example 2.

```
13:34:25.517 (0x80cd320*) CREATE JOB (modems= 0x1)
13:34:25.517 (0x80cd320*) (g56+m1382.) ECM allow ASN1=0
13:34:25.517 (0x80cd320*) PATH A RX EP = T38FMP
13:34:25.517 (0x80cd320*) PATH A TX EP = T30TDM

13:34:25.517 (0x80cd320*) PATH B RX EP = T30TDM
13:34:25.517 (0x80cd320*) PATH B TX EP = T38FMP
13:34:25.557 (0x80cd320*) 0 CNG
13:34:26.207 (0x80cd320*) 1 CED
13:34:30.338 (0x80cd320*) 1 CSI
13:34:30.758 (0x80cd320*) 1 DIS
```

```

13:34:34.488 (0x80cd320*) 0 TSI
13:34:34.648 (0x80cd320*) 0 DCS
13:34:34.648 (0x80cd320*) V27 - 4800
13:34:39.609 (0x80cd320*) 1 CFR
13:36:36.729 (0x80cd320*) 0 PPS_NULL
13:36:39.311 (0x80cd320*) 1 MCF
13:38:36.271 (0x80cd320*) 0 PPS_NULL
13:38:38.841 (0x80cd320*) 1 MCF
13:39:45.597 (0x80cd320*) 0 PPS-EOP
13:39:48.187 (0x80cd320*) 1 MCF
13:39:50.37 (0x80cd320*) 0 DCN
13:39:50.568 (0x80cd320*) DESTROY JOB

```

T.38 Gateway type B (Image from network), ECM allowed (and used in this fax), v27 only enabled, Pre-corrigendum ASN1, Fax transfer of three partial pages proceeded without a problem.

The fax transmitter is connected to the T.38 Gateway over the network, the fax receiver is connected via the TDM bus.

Example 3.

```

15: 6: 6.114 (0x56b7d8*) CREATE JOB (modems= 0x7)
15: 6: 6.114 (0x56b7d8*) (g56+m1382.) ECM allow ASN1=0
15: 6: 6.114 (0x56b7d8*) PATH A RX EP = T38FMP
15: 6: 6.114 (0x56b7d8*) PATH A TX EP = T30TDM
15: 6: 6.114 (0x56b7d8*) PATH B RX EP = T30TDM
15: 6: 6.114 (0x56b7d8*) PATH B TX EP = T38FMP
15: 6: 6.207 (0x56b7d8*) 1 CED
15: 6:10.942 (0x56b7d8*) 1 CSI
15: 6:11.379 (0x56b7d8*) 1 DIS
15: 6:13.785 (0x56b7d8*) 0 DCS
15: 6:13.785 (0x56b7d8*) V17 - 14400
15: 6:18.754 (0x56b7d8*) 1 CFR
15: 6:28.67 (0x56b7d8*) 0 MPS
15: 6:30.98 (0x56b7d8*) 1 RTN
15: 6:32.4 (0x56b7d8*) 0 DCS
15: 6:32.4 (0x56b7d8*) V17 - 14400
15: 6:36.926 (0x56b7d8*) 1 CFR
15: 6:42.67 (0x56b7d8*) 0 MPS
15: 6:44.98 (0x56b7d8*) 1 RTN
15: 6:46.20 (0x56b7d8*) 0 DCS
15: 6:46.20 (0x56b7d8*) V17 - 14400
15: 6:50.926 (0x56b7d8*) 1 CFR
15: 7:12.4 (0x56b7d8*) 0 EOP
15: 7:14.66 (0x56b7d8*) 1 MCF
15: 7:15.801 (0x56b7d8*) 0 DCN
15: 7:16.426 (0x56b7d8*) DESTROY JOB

```

T.38 Gateway type B (Image from network), ECM allowed (although not used in this fax), v17,v29,v27 modems enabled (transfer used v17 14400), Pre-corrigendum ASN1.

Fax receiver trained returning a CFR to the fax transmitter, first page is transferred but the fax receiver returned a RTN to the transmitter. The fax transmitter re-trained at the same speed, re-sent the page but again a RTN was returned by the fax receiver. The page is finally transferred at the third attempt, then a DCN is sent by the fax transmitter.

Example 4.

```
09:14:36:31:399 CREATE SESSION
09:14:36:31:415 (0x388c20*) CREATE JOB (modems= 0x3)
09:14:36:31:415 (0x388c20*) (g56+m1382.) ECM allow ASN1=3
09:14:36:31:415 (0x388c20*) PATH A RX EP = T30TDM
09:14:36:31:415 (0x388c20*) PATH A TX EP = T38FMP
09:14:36:31:415 (0x388c20*) PATH B RX EP = T38FMP
09:14:36:31:415 (0x388c20*) PATH B TX EP = T30TDM
09:14:36:44:961 (0x388c20*) 1 CED
09:14:36:50:133 (0x388c20*) 1 CSI
09:14:36:50:321 (0x388c20*) 1 DIS
09:14:36:52:336 (0x388c20*) 0 CNG
09:14:36:56:446 (0x388c20*) 1 CSI
09:14:36:56:617 (0x388c20*) 1 DIS
09:14:36:59:367 (0x388c20*) 0 CNG
09:14:37:02:742 (0x388c20*) 1 CSI
09:14:37:02:914 (0x388c20*) 1 DIS
09:14:37:06:352 (0x388c20*) 0 CNG
09:14:37:09:039 (0x388c20*) 1 CSI
09:14:37:09:227 (0x388c20*) 1 DIS
09:14:37:15:336 (0x388c20*) 1 CSI
09:14:37:15:523 (0x388c20*) 1 DIS
09:14:37:16:852 (0x388c20*) 0 CNG
09:14:37:21:148 (0x388c20*) 1 DCN
09:14:37:22:273 (0x388c20*) DESTROY JOB
09:14:37:22:289 (0x388be8*) STOP SESSION
09:14:37:22:289 (0x388be8*) DESTROY SESSION
```

T.38 Gateway type A (Image to network), ECM allowed, v29,v27 modems enabled, NOT Pre-corrigendum ASN1. Fax transmitter is sending CNG tones, fax receiver sends CED tone, then repeated CSI/DIS sequences before finally terminating the transfer with a DCN.

The fax transmitter is not receiving the CSI/DIS so continually sends CNG tones, the fax receiver never receives a DCS so repeats the CSI/DIS. The sequence is terminated by a DCN from the fax receiver.

This log also includes the session create, stop and destroy.

7 T.38 Gateway FCF modification.

The T.38 Gateway is connected between the fax transmitter and receiver and performs protocol conversion transparently without having any effect upon the fax endpoints or the transfer. There are however, two cases where the T.38 Gateway modifies a FCF received from a fax endpoint before transferring it.

7.1 Digital Information Signal modification.

When a T.38 job is created a parameter specifies the modems that can be used for the fax transfer. The T.38 Gateway enforces this by masking out the appropriate bits in the DIS FCF payload. In a similar way if ECM is disabled the FCF payload is modified and a prompt displayed in the T.38 Gateway log.

```
(0x81140d8) COMPLETE FRAME
(0x81140d8*) 1 DIS
(0x81140d8) dsr (0xee) -> (0xce) // (v17,v29,v27) =>(v29,v27)
(0x81140d8) ECM forced off
```

The T.38 Gateway cannot impose the use of ECM for a fax transfer.

7.2 Non Standard Facilities modification.

The T.38 Gateway clears out any payload contained in a NSF message received from a fax receiver endpoint. The cleared fields are displayed before and after modification.

```
15:41: 2.735 (0x564a48*) 1 NSF
15:41: 2.735 (0x564a48) STATE_B_TXIDLE
15:41: 2.735 (0x564a48) STATE_B_WAITTONEV21
15:41: 2.735 (0x564a48) NSF cleared ( 0 => 0),( 0 => 0)
```

The NSF payload fields may have already been cleared.

8 Data display.

When the T.38 Gateway logging options *T38GW_TRACELVL_INDATA* and/or *T38GW_TRACELVL_OUTDATA* are enabled all received and transmitted bytes are displayed. Otherwise only the number of received or transmitted bytes is displayed. Three types of data are transferred by the T.38 Gateway v21, Fast Sync data, Fast ECM data.

8.1 Message and Image carriers.

The start and end of the v21 (*Message*) data and the Fast (*Image*) data is displayed as carrier on/off entries. Since v21 is received on both paths, the Msg carrier indication also displays the path. The path A Msg carrier off indication may not be displayed for path A on the switch from v21 to Fast data after the DCS.

```
41:24.110 (0x564a48) STATE A V21 EV
41:24.110 (0x564a48) Msg carrier (0) on.
.....
41:25.501 (0x564a48) STATE A V21 EV
41:25.501 (0x564a48) Msg carrier (0) off.
.....
41:18.954 (0x564a48) STATE A FAST EV
41:18.954 (0x564a48) Img carrier on
.....
41:24. 48 (0x564a48) STATE A FAST EV
41:24. 48 (0x564a48) Img carrier off
```

8.2 V21.

V21 uses the HDLC protocol described in [1].

```
1.954 (0x564a48) STATE B V21 EV
1.970 (0x564a48) READ  ff ff ff ff 00 00 00 00 05 00 00 00
1.970 (0x564a48) Flags (1)
2.548 (0x564a48) STATE B V21 EV
2.548 (0x564a48) READ  ff 03 20 00
2.735 (0x564a48) STATE B V21 EV
2.735 (0x564a48) READ  00 00 55 55 00 8c 90 80 4b 0d 00 00 ff
ff ff ff 3e 00 00 00 00 00 00
2.735 (0x564a48*) 1 NSF
2.735 (0x564a48) STATE_B_TXIDLE
2.735 (0x564a48) STATE_B_WAITTONEV21
2.735 (0x564a48) NSF cleared ( 0 => 0),( 0 => 0)
```

The bytes are displayed in the T.38 Gateway log bit reversed and the path can be identified by the event proceeding the read. In this case it is a read on path B from a fax receiver. The v21 'Flags' and the NSF indicators display (1) that is path B. V21 data received on path A is displayed in the same format with '0' indicating path A.

```
(0x4a48) STATE A V21 EV
(0x4a48) READ  ff 03 43 30
(0x4a48) STATE A V21 EV
(0x4a48) STATE A V21 EV
(0x4a48) READ  31 20 37 36
(0x4a48) STATE A V21 EV
(0x4a48) READ  20 20 20 20
(0x4a48) STATE A V21 EV
(0x4a48) READ  20 20 20 20 20 20 20 20
(0x4a48) STATE A V21 EV
(0x4a48) READ 20 20 2b a8 ff ff ff ff bf 3e 00 00 00 00 00 00
(0x4a48*) 0 TSI
```

Sync data.

The TCF (Training Check Flag) consists of 1.5 seconds of zero data that is always transferred as sync on path A.

```
41:14.376 (0x564a48) STATE A FAST EV
41:14.376 (0x564a48) READ  00 00 00 00 00 00 00 00 00 00 ...
41:14.376 (0x564a48) STATE A TXFAST
41:14.376 (0x564a48) SEND  00 00 00 00 00 00 00 00 00 00 00 ...
41:14.376 (0x564a48) STATE A TXFAST
41:14.626 (0x564a48) STATE A FAST EV
41:14.626 (0x564a48) READ  00 00 00 00 00 00 00 00 00 ..
41:14.626 (0x564a48) STATE A TXFAST
41:14.626 (0x564a48) SEND  00 00 00 00 00 00 00 00 00 00 00..
```

The T.38 Gateway *always* uses 'transferred TCF' and does not modify or correct the TCF in any way. The TCF can be checked by extracting the SEND or READ data entries and then inspecting for length and content.

Image data sent as sync is displayed in the same format.

```
42:49.860 (0x56afc0) STATE A FAST EV
42:49.860 (0x56afc0) READ  0e 05 00 e8 ff 00 00 08 86 21..
42:49.860 (0x56afc0) STATE A TXFAST
42:49.860 (0x56afc0) SEND  0e 05 00 e8 ff 00 00 08 86 21..
```

ECM.

ECM mode uses HDLC protocol [1] for image transfer with the T.38 Gateway log displaying PPS-NUL, PPS-MPS and PPS-EOP image page transfer FCF messages.

9 The Stop Sequence.

A T.38 Gateway job can be stopped by a DCN received on either path (that is from the fax transmitter or fax receiver) or by an ABORT command issued by the application. Early job termination can also be initiated by an error condition detected by the T.38 Gateway.

In all cases the main sequence is exactly the same being initiated by the DCS, application ABORT call or an error condition. It is assumed that a shutdown can occur at any point in a fax sequence.

This shutdown sequence consists of an ordered stop of all the channels currently enabled in the T.38 Gateway job. When all of these are terminated the customer supplied callback function is called and the job is deleted from the session. At this point control of the job and the job resources are returned to the application.

```
(0x386bf8) STATE A V21 EV
(0x386bf8) READ  ff 13 fb 9a ff ff ff ff f6 3e 00 00 00 00 00
(0x386bf8) COMPLETE FRAME
(0x386bf8*) 0 DCN
(0x386bf8) STATE_A_TXV21
(0x386bf8) STATE_A_WAITV21_END
(0x386bf8) STOP ev (AMcfg 1, AICfg 1, Bmcfg 1, ATxcfg 1, BTxcfg 0)
(0x386bf8) stop_job(): in_progress = 57
(0x386bf8) Stopping V21 A
(0x386bf8) Stopping Fast A
(0x386bf8) Stopping V21 B
(0x386bf8) stop_job() - exit: in_progress = 56
(0x386bf8) Waiting for 56 jobs to finish
(0x386bf8) STATE A TXV21
(0x386bf8) SEND  18 00 00 00 ff 13 fb 00
(0x386bf8) STATE A TXV21
(0x386bf8) SEND  00 00 00 00
(0x386bf8) STATE A TXV21
(0x386bf8) TX V21 in stopping state
(0x386bf8) SEND  00 00 00 00
(0x386bf8) STATE A TXV21
(0x386bf8) STATE A FAST EV
(0x386bf8) STATE_A_WAITV21_END: got finish from stopped msg
(0x386bf8) rxstate = 8: newstate = 14
(0x386bf8) STOP ev (AMcfg 1, AICfg 0, Bmcfg 1, ATxcfg 1, BTxcfg 0)
(0x386bf8) Waiting for 52 jobs to finish
(0x386bf8) STATE A V21 EV
(0x386bf8) rxstate = 8: newstate = 14
(0x386bf8) STOP ev (AMcfg 0, AICfg 0, Bmcfg 1, ATxcfg 1, BTxcfg 0)
(0x386bf8) Waiting for 50 jobs to finish
(0x386bf8) STATE B V21 EV
(0x386bf8) t38 input channel path (1) stopped.
(0x386bf8) STOP ev (AMcfg 0, AICfg 0, Bmcfg 0, ATxcfg 1, BTxcfg 0)
(0x386bf8) Waiting for 40 jobs to finish
(0x386bf8) STATE A TXV21
(0x386bf8) txstate = 2: txnewstate = 0
(0x386bf8) in_progress = 0
(0x386bf8) STOP ev (AMcfg 0, AICfg 0, Bmcfg 0, ATxcfg 0, BTxcfg 0)
(0x386bf8) Callback_1(in)
(0x386bf8) Callback_1(out) (1 - 0 - 1)
```

(0x386bf8) STATUS **// Application called status**

(0x386bf8*) DESTROY JOB **// Application destroyed job.**

A log entry displays the status of the shutdown sequence where **cfg z* indicates the state of a channel when *z=0* the channel is stopped.

(0x386bf8) STOP ev (AMcfg 1, AIfcg 1, Bmcfg 1, ATxcfg 1, BTxcfg 0)

Amcfg *path A message (v21) channel.*

AIfcg *path A image (fast) channel.*

Bmcfg *path B message (v21) channel.*

ATxcfg *path A transmit channel.*

BTxcfg *path B transmit channel.*

The progress of the stop sequence is displayed on every stop event with.

(0x386bf8) Waiting for 56 jobs to finish

The number in this entry is a hex 'in progress' value with each bit allocated to a particular job channel (in this case 01010110). When this value goes to zero the job is then stopped.

On an error condition, the stop sequence is proceeded by an error prompt including the message *'kill_job'*. Note that when a job is aborted or killed the job is terminated immediately regardless of the state of the T.30 fax sequence.

10 Callback function.

The T.38 Gateway calls the customer supplied callback function on reception of the message conformation from the receiver for each page sent and at the end of the stop sequence to inform the application that the job has terminated and has been removed from the session.

In ECM mode the maximum size of an image page is limited so a fax set using ECM may consist of more pages than the same fax sent in non ECM mode.

```
(0x564a48) STATE B V21 EV
(0x564a48) READ ff 13 8c 00 ff ff ff ff 00 3e 00 00 00 00 00
(0x564a48*) 1 MCF
(0x564a48) STATE_B_TXV21
(0x564a48) STATE_B_WAITV21
(0x564a48) callback_2(in)
(0x564a48) callback_2(out)
```

Callback at the end of the stop sequence.

```
(0x386bf8) STOP ev (AMcfg 0, AICfg 0, Bmcfg 0, ATxcfg 0, BTxcfg 0)
(0x386bf8) Callback_1(in)
(0x386bf8) Callback_1(out) (1 - 0 - 1)
```

In both of these cases it is the same callback function invoked. The *Callback_1 (out)* entry also displays the job status results, termination reason, and page count.

```
enum tSMT38GWJobStatus {
    tSMT38GWJobStatusRunning,
    tSMT38GWJobStatusTerminated,
} status

enum tSMT38GWJobTerminationReason {
    tSMT38GWJobTerminationDCN, // job terminated due to a DCN.
    tSMT38GWJobTerminationUser, // job terminated by ABORT.
    tSMT38GWJobTerminationError, // job terminated early due to an error
} termination_reason.
```

The termination reason relates to the T.38 Gateway job only and therefore cannot be used as an indication of the status of the transferred fax image.

Timestamps provide the time spent within the customer supplied callback function.

Care should be taken when implementing the callback function as it is called within the context of the Aculab T.38 Gateway and therefore should be implemented as efficiently as possible. Looping or other complex routines should be avoided.

No resources used by the T.38 Gateway job should be deleted within the callback function. The most efficient callback function implementation simply sets an event and then returns.

11 Fax spoofing.

The methods used for T.30 fax spoofing are described in [3]. By default spoofing is enabled when a T.38 Gateway job is created with `sm_t38gw_create_job` [2] and only produces output to the T.38 log on a spoof timeout.

11.1 T.38 Gateway 'A' spoofing.

This example shows spoofing of path B with v21 flags, the T.38 Gateway has transferred the TCF and is awaiting the CFR from the fax receiver. On the spoof timeout a second stage timeout is set to limit the maximum v21 length. This is dependent upon the length of the expected FCF.

A reduced number of v21 flags is sent as preamble to the CFR.

```

1587537 (0x100a24b8) Tx Path A Paused.
1587588 (0x100a24b8) A TX WAIT TOUT
1590127 (0x100a24b8*) Path A Spoof timeout. 2nd= 106
1590128 (0x100a24b8) txstate = 4: txnewstate = 6
1590128 (0x100a24b8) chan - disrx (0) (c=100b4cb8)
1590128 (0x100a24b8) chan - disrx (0) (c=1006e528)
1590136 (0x100a24b8) STATE B Tx EV
1590137 (0x100a24b8) STATE_B_TXV21
1590137 (0x100a24b8) Tx Generic - carrier on, path - 1 [st0] 2
1590137 (0x100a24b8) STATE B Tx EV
1590137 (0x100a24b8) STATE_B_TXV21
1590137 (0x100a24b8) SEND (4).
1590137 (0x100a24b8) STATE B Tx EV
1590137 (0x100a24b8) STATE_B_TXV21

1590247 (0x100a24b8) STATE B V21 EV
1590247 (0x100a24b8) Carrier (1) ON t38 chan - 10078ca8
1590247 (0x100a24b8) Msg carrier (1) on.
1590247 (0x100a24b8) STATE B V21 EV
1590247 (0x100a24b8) READ (12).
1590247 (0x100a24b8) Flags (1) [10]
1590247 (0x100a24b8) Path B Spoof flags reduced (-28).
1590247 (0x100a24b8) Send flags(2-Msg) (path 1) -28.
1590248 (0x100a24b8) STATE B Tx EV
1590248 (0x100a24b8) STATE_B_TXV21
1590248 (0x100a24b8) STATE B Tx EV
1590248 (0x100a24b8) STATE_B_TXV21
1590248 (0x100a24b8) STATE B Tx EV
1590248 (0x100a24b8) STATE_B_TXV21
1590248 (0x100a24b8) SEND (4).
1590248 (0x100a24b8) STATE B Tx EV
1590248 (0x100a24b8) STATE_B_TXV21
1590316 (0x100a24b8) STATE B V21 EV
1590316 (0x100a24b8) READ (12).
1590316 (0x100a24b8) Flags (1) [20]
1591446 (0x100a24b8) STATE B V21 EV
1591446 (0x100a24b8) READ (16).
1591446 (0x100a24b8*) 1 CFR

```

11.2 T.38 Gateway 'B' spoofing.

v21 HDLC flags.

Spoofing of v21 data to path 'A' follows the same format as (11.1).

Image HDLC flags.

This example shows spoof of the image path ('A') to the fax receiver on a delayed message conformation (MCF).

```
2133837 (0x10095638*) 1 MCF
2133837 (0x10095638) STATE_B_TXV21
2133837 (0x10095638) STATE_B_WAITV21
2133837 (0x10095638) callback_2(in)
2133837 (0x10095638) callback_2(out)
2133837 (0x10095638) rxstate = 8: newstate = 6
2133837 (0x10095638) carriers set off 10, path 0
2133837 (0x10095638) STATE B Tx EV
2133837 (0x10095638) STATE_B_TXV21
2133837 (0x10095638) SEND (8).
2133838 (0x10095638) STATE B Tx EV
2133838 (0x10095638) STATE_B_TXV21
2133847 (0x10095638) STATE A V21 EV(t38 rx)
2133887 (0x10095638) STATE B V21 EV
2133887 (0x10095638) READ (12).
2133907 (0x10095638) STATE B V21 EV
2133907 (0x10095638) Carrier (1) OFF t30 chan - 1007e9a8
2133907 (0x10095638) Msg carrier (1) off.
2133907 (0x10095638) STATE B Tx EV
2133907 (0x10095638) STATE_B_TXV21
2133907 (0x10095638) SEND (4).
2133907 (0x10095638) STATE B Tx EV
2133907 (0x10095638) STATE_B_TXV21
2133947 (0x10095638) STATE B Tx EV
2133947 (0x10095638) STATE_B_TXV21
2133978 (0x10095638) STATE B Tx EV
2133978 (0x10095638) STATE_B_TXV21
2133978 (0x10095638) txstate = 6: txnewstate = 4
2133978 (0x10095638) in_progress = 1f
2136807 (0x10095638) Path B Spoof timeout.
2136807 (0x10095638*) Path B spoof send Flags (-10)
```



```
2136807 (0x10095638) txstate = 0: txnewstate = 3
2136807 (0x10095638) chan - disrx (1) (c=1007e9a8)
2136817 (0x10095638) STATE A Tx EV
2136817 (0x10095638) STATE A TXFAST
2136817 (0x10095638) Tx Generic - carrier on, path - 0 [st1] 2
HDLC flags will be sent to path A image channel until image data arrives or a spoof
sequence timeout in the T.38 Gateway.
```

Image Sync spoof data.

Here the image path (Path A) is spoofed with Sync data to prevent a timeout of the fax receiver. Spoofing attempts to maintain a fixed amount of buffered data (in this case 40 mS). When the image data is received the spoofing sequence is terminated.

The T.38 Gateway buffers the first image packet in a page to improve inter-working with other devices.

```
2015687 (0x100b3418) A TX WAIT START [300]
2015987 (0x100b3418) A TX WAIT TOUT
2016329 (0x100b3418) Path B Spoof timeout
2016329 (0x100b3418*) Path B spoof Sync. - start
2016329 (0x100b3418) Path B spoof Sync.  1 sp(4800) sz( 60)
                tout( 60) tr(  0) dl[  0]
2016329 (0x100b3418) STATE A Tx EV
2016329 (0x100b3418) STATE A TXFAST
2016329 (0x100b3418) SEND  (60)
2016329 (0x100b3418) STATE A Tx EV
2016329 (0x100b3418) STATE A TXFAST
2016389 (0x100b3418) Path B Spoof timeout
2016389 (0x100b3418) Path B spoof Sync.  2 sp(4800) sz( 60)
                tout(200) tr(  60) dl[140]
2016389 (0x100b3418) STATE A Tx EV
2016389 (0x100b3418) STATE A TXFAST
2016389 (0x100b3418) SEND  (60)
2016390 (0x100b3418) STATE A Tx EV
2016390 (0x100b3418) STATE A TXFAST
2016428 (0x100b3418) STATE A FAST EV
2016428 (0x100b3418) READ  (168).
2016589 (0x100b3418) Path B Spoof timeout
2016589 (0x100b3418) Path B spoof Sync.  3 sp(4800) sz( 60)
                tout(100) tr( 260) dl[ 40]
2016589 (0x100b3418) STATE A Tx EV
2016589 (0x100b3418) STATE A TXFAST
```

```
2016589 (0x100b3418) SEND (60).
2016589 (0x100b3418) STATE A Tx EV
2016589 (0x100b3418) STATE A TXFAST
2016689 (0x100b3418) Path B Spoof timeout.
2016689 (0x100b3418) Path B spoof Sync. 4 sp(4800) sz( 60)
                tout(100) tr( 360) dl[ 40]
2016689 (0x100b3418) STATE A Tx EV
2016689 (0x100b3418) STATE A TXFAST
2016689 (0x100b3418) SEND (60)
2016689 (0x100b3418) STATE A Tx EV
2016689 (0x100b3418) STATE A TXFAST
2016698 (0x100b3418) STATE A FAST EV
2016698 (0x100b3418) READ (168)
2016698 (0x100b3418) STATE A Tx EV
2016698 (0x100b3418) STATE A TXFAST
2016698 (0x100b3418) SEND (168)
```

This log sequence shows a spoof timeout, sending of spoof data and then the transition to image data.

12 References.

- [1] Prosody data communications Encodings: HDLC.
- [2] T.38_Gateway_API_guide.
- [3] T.38 Gateway introduction.