This not summarizes the changes to the 1394b C code which have been made to create the 1394c C code.

1 Revised data_structures.h

1.1 Introduces the constants:-

#define TMODE_JITTER 9 // number of 10 bit "no symbol" symbol times resulting from insertion of
                     // 8 bit idles between data bursts on 802.3ab encoding
#define TMODE_RESUME_CHECKS ((GMII_LINK_PULSE_INTERVAL/10)+1) // number of pulse times per
                  // interval

1.2 Add to port_wait_times

enum port_wait_times {...

, GMII_PORT_ENABLE_TIME,
GMII_AUTONEGOTIATE_TIME, GMII_LINK_PULSE_DURATION, GMII_LINK_PULSE_INTERVAL,
GMII_RECEIVER_INIT_TIME};

1.3 Add internal decode position state:-

// Tport symbol position state
typedef enum {A, B, C, D } symbol_position;

1.4 Add a port state

typedef enum {... , P13} port_state_type;

1.5 Add two more ArbState internal symbols for use in the T port.

...
1.6 Add a structure to represent the GMII interface

typedef struct {
    int TXD;
    int TX_EN;
    int TX_ER;
    int GTX_CLK;

    int COL;

    int RXD;
    int RX_ER;
    int RX_CLK;
    int RX_DV;

    int CRS;
} GMII;

2 Revised shared.h

2.1 Change the receive FIFO size

#define FIFO_DEPTH (10+TMODE_JITTER+2) // depth of receive FIFO - 10 normally, add extra for tports
    // add two extra symbols for quantization effects

2.2 Add shared variables and constants:

    // shared between tport_tx.c and tport_rx.c

#ifdef unsubscripted
    GMII gmii;                   // TPORT data transfer interface
#else
    GMII gmii[NPORT];
#endif

const Tmode_port;       // TRUE if the port is a T_mode (and therefore T_mode only) port

// shared between port.c, tport_rx.c and tport_tx.c

boolean tport_on;       // Set to true to instruct the T port to commence operating
// Set to false to instruct the T port to cease operating

boolean tport_on[NPORT];

boolean TPORT_802_mode; // Set if the port is operating in TPORT_802_mode (and we can't get at it!)

boolean T_mode;         // Set if the port is operating in T mode,
// unset otherwise (i.e. when operating in Beta_mode or DS-Mode,
// unset otherwise (i.e. when operating in DS-Mode,
// or when in P0:disconnected).
// This is maintained as a read-only bit in the port register map.
3 Revised receive_functions()

Add code to center the FIFO allowing for the jitter on a T_mode connection:-

In start_rx_packet():-

```c
if (T_mode[receive_port]) {
    arb_timer = 0;
    while ((fifo_backlog < (FIFO_DEPTH/2)) && (arb_timer < TMODE_JITTER)) {
        // Allow time for the FIFO to load, but start repeat immediately if critical
        fifo_backlog = (FIFO_DEPTH + fifo_wr_ptr[receive_port] - fifo_rd_ptr[receive_port]) %
        FIFO_DEPTH;
    }
}
```

4 Revised Phy services

4.1 New control request types

```c
// 1394c specific

PMD_LINKPULSE_ON,       // instructs the PMD to generate an Ethernet link pulse
PMD_LINKPULSE_OFF,      // instructs the PMD to cease generating an Ethernet link
PMD_AUTONEGOTIATE,      // instructs the PMD to perform 1394c/Ethernet
PMD_SELECT_TPORT,       // PMD uses bport for t mode I/O - only set while
                        // tport_on is TRUE, unset during suspend etc
PMD_DISCONNECT_TPORT,   // Stops 802.3ab PHY operating as a Tport PHY.
PMD_TPORT_OFF          // Clears operating mode status bits
}
PMD_control_request_type;
```

4.2 New status values

```c
// 1394c
#define PMD_TPORT_SIGNAL_DETECT 0x00000040
#define PMD_AUTONEGOTIATION_ACTIVE 0x00000080
#define PMD_TPORT_FW 0x00000100  // autonegotiation has established FW mode
#define PMD_TPORT_802 0x00000200  // autonegotiation has established GE mode
#define PMD_TPORT_OK 0x00000400  // 802.3ab connection is synchronized (FW mode only)
```

5 Revised port.c

Numerous updates to add in tport control – see code
6 New tport_rx.c and tport_tx.c
Loosely based in bport_rx.c and bport_tx.c.