# Exp 2: MAC 地址查询

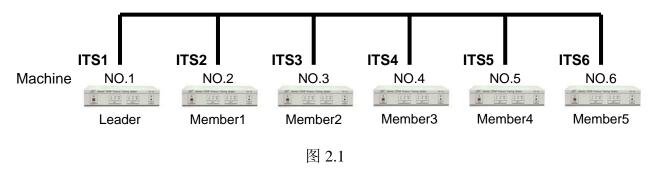
目的: 学习通过 MDDL 编辑简单的控制程序,并将实验结果与手动操作的结果

做比较。

摘要: 使用局域网基本通讯机制,完成数据报文的请求与回复的实验。

时间: 3 hrs。

# 一、网络拓扑



# 二、技术背景

以下我们自定义一个MAC address 的询问及响应格式 (如:图 2.1)。我们利用自订以太网数据包的类型和架构来完成MAC 地址的询问及响应,如:(0xA001)表示询问MAC 地址,而(0xA002 or 0xA003)则表示MAC地址的响应其步骤如下:

首先由学生发出MAC 地址的询问数据包,注意在数据包内需要注明询问ITS的机器号码及网卡的编号。接着当其他学生接收到这个询问数据包,并且询问对象为本身时,就选择以Format 1 或Format 2 回应询问端。

# MAC Address Request

Type	Data		
A0:01	Quer	ied Machine Number	Queried Interface Number
12 <sup>th</sup> 13	th 14 <sup>th</sup>	15 <sup>th</sup>	byte
		(a)	

# MAC Address Responses

# Format 1

A0:02	Queried Machine Number	Queried Interface	Queried MAC
		Number	Address

# Format 2

A0:03	Queried Machine Number	Queried Interface		
		number		
	(b)			

表 2.1

# 三、实验步骤

18

# 1、网络拓扑连线

在 Hubox 上网络拓扑连线如图 2.2。

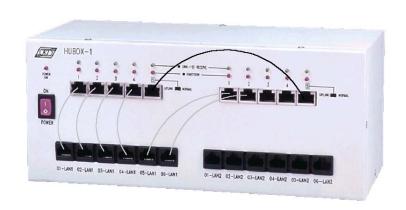




图 2.2

## 2、手动询问与回应帧

#### Leader and Members

- 1) 执行 **XCLIENT.BAT**, 打开 KCodes Network Explorer。
- 2) 打开网络信息浏览器(Network Message Browser). 勾选 Listening On。

#### Leader

- 3) 打开 Network Message Sender 对话框界面. 在 Destination MAC Address 文本框中输入 "FF:FF:FF:FF:FF:FF",在类型文本框中输入 "A0:01"。
- 4) 首先决定哪一位 Member 以及该 Member 的哪个网卡要成为询问的对象。再参照表 2.1(a)将数值输入到 Data 部分。举个例子,假设我们要询问 Member1 的第一张网卡(Interface 1)的 MAC 地址,就将 "02:01"(拓扑图中, Member1 的 ITS 机器号为 2) 输入到 Data 部分,如图 2.3 所示,最后单击 Send 按钮。

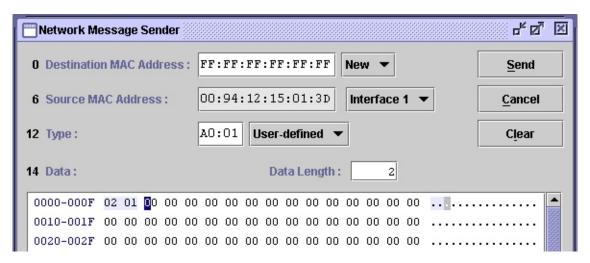


图 2.3

### <u>Members</u>

- 5)当 ITS 的网络信息浏览器收到帧时,观察该帧的详细内容。如果发现该帧是来自于 Leader 而且要询问的正式你的号码,请打开帧发送界面"Network Message Sender"。
- 6) 将 Leader 的网卡 MAC 输入到 **Destination MAC Address**,并自行定义一种以太网帧类型 "**A0:02**" or "**A0:03**",输入到 TYPE 文本框中。例如,当 Member1 收到来自 Leader 的询问帧时,Member1 就要将 Leader 的网卡 MAC "**00:94:12:15:01:3D**" 输入到 **Destination MAC Address** 文本框中,

并将 "**A0:02**" 输入到类型文本框中。如 图 2.4 所示,再参照表 2.1(b)的 Format 1 的格式,将 "**02,01,00,94,12,15,01,3E**" 输入到 Data 文本框中,最后单

### 击 Send 按钮。

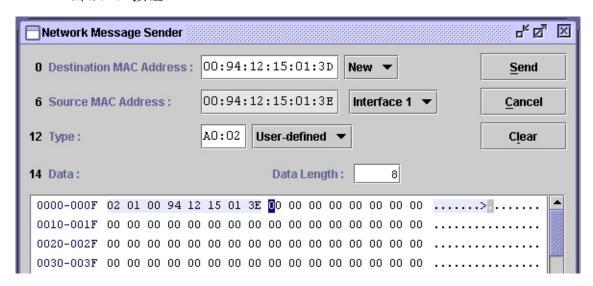


图 2.4

## 3、利用 MDDL 发送询问和回应帧

#### Leader and Members

7) 重新打开一个新的网络信息浏览器"Network Message Browser"界面,勾选 Listening On。

## Leader

- 8) 从 Reactor menu 中找到并运行 MDDL Reactor Panel ,会自动打开一个名为 MDDL Editor 的 MDDL 程序编辑平台。
- 9)在 MDDL Editor 的 MDDL 程序编辑平台中,单击 Load 按钮,调用 BCastRequest.mddl 程序(路径 C: \XClient \Data \Mddl \Tutorial \Ex02 \BCastRequest.mddl ),最后单击 Upld 按钮将程序载入到 ITS 中。您的 ITS 将会持续不断的发送一个询问用的广播帧,以每 5 秒一次的频率询问网络上的所有成员。

## Members

- 10) 从 Reactor menu 中找到并运行 MDDL Reactor Panel ,会自动打开一个名 为 MDDL Editor 的 MDDL 程序编辑平台。
- 11) 在 **MDDL Editor** 的 MDDL 程序编辑平台中,单击 **Load** 按钮,调用 AutoResponserFormat1.mddl 或者 AutoResponserFormat2.mddl 其中一个程序,可自由选择(路径为 C: \XClient \Data \Mddl \Tutorial \Ex02)。
- 13)将自己本身的 ITS 编号输入到 AutoResponserFormat1.mddl (如 图 2.5) 或者

AutoResponserFormat2.mddl (如图 2.6)程序中,最后单击 Upld 按钮。

```
ETHER_IN_HANDLER ANY
                                           输入您的 ITS 机器编号
   IF( S[12] == \{0XA0\} \&\& S[13] == \{0x01\} \&\& S[14] == 2 )
   {
       SEND_OUT_ETHER_FROM_INTERFACE INTERFACE() WITH_DATA
          T
                             = S,
          T.ETHER_MACADDRDST = S.ETHER_MACADDRSRC,
          T.ETHER\_MACADDRSRC = MYMAC(INTERFACE()),
//ETHER_MACADDRDST,
          T.[12]
                            = \{0xA0\},\
          T.[13]
                            = \{0x02\},\
          T.[14]
                            = S.[14],
          T.[15]
                            = S.[15],
          T.[16,21]
                        =MYMAC(S.[15])
       }
   }
```

图 2.5

```
ETHER IN HANDLER ANY
                                              输入您的 ITS 机器编号
   IF( S[12] == \{0XA0\} \&\& S[13] == \{0x01\} \&\& S[14] == \{0XA0\} \&\& S[15] == 1 || S[15] == 2 ))
    {
       SEND_OUT_ETHER_FROM_INTERFACE INTERFACE() WITH_DATA
           T
                               = S,
           T.ETHER_MACADDRDST = S.ETHER_MACADDRSRC,
           T.ETHER_MACADDRSRC =
MYMAC(INTERFACE()),//ETHER_MACADDRDST,
           T.[12]
                              = \{0xA0\},\
           T.[13]
                              = \{0x03\},\
           T.[14]
                              = S.[14],
           T.[15]
                              = S.[15]
        }
```

图 2.6

14)程序载入后,如果当 Leader 发的帧是询问帧,ITS 就将会自动以 Format 1 或 Format 2 的格式回复给 Leader。

### **DISCUSSIONS**

- 1、讨论 Format 1 和 Format 2的不同之处(提示: 试着询问任意一台机器的 interface2)。
- 2、以 Format 1 和 Format 2 为基础, 讨论是否可以设计出一个更好更完善的协议?

### **REACTOR PROGRAMS**

1. AutoResponserFormat1.mddl

```
ETHER_IN_HANDLER ANY
```

```
IF( S[12] == \{0XA0\} \&\& S[13] == \{0x01\} \&\& S[14] == 01 \&\& (S[15] == 1 || S[15] == 2))
           {
               SEND_OUT_ETHER_FROM_INTERFACE INTERFACE() WITH_DATA
               {
                   T
                                        = S,
                   T.ETHER\_MACADDRDST = S.ETHER\_MACADDRSRC,
                   T.ETHER_MACADDRSRC
       MYMAC(INTERFACE()),//ETHER_MACADDRDST,
                   T.[12]
                                      = \{0xA0\},\
                   T.[13]
                                      = \{0x02\},\
                   T.[14]
                                      = S.[14],
                   T.[15]
                                      = S.[15],
                   T.[16,21]
                                   =MYMAC(S.[15])
               }
           }
       }
2, AutoResponserFormat2.mddl
       ETHER_IN_HANDLER ANY
       {
           IF( S[12] == \{0XA0\} \&\& S[13] == \{0x01\} \&\& S[14] == 01 \&\& (S[15] == 1 || S[15] == 2))
           {
               SEND_OUT_ETHER_FROM_INTERFACE INTERFACE() WITH_DATA
               {
                   T
                                        = S,
                   T.ETHER\_MACADDRDST = S.ETHER\_MACADDRSRC,
                   ETHER_MACADDRSRC=MYMAC(INTERFACE()),//ETHER_MACADDRDST,
                   T.[12]
                                      = \{0xA0\},\
                   T.[13]
                                      = \{0x03\},\
```

= S.[14],

T.[14]

```
T.[15] = S.[15] } }
```

# 3, BCastRequest.mddl

```
TIMER_WITH_PERIOD 5000
   SEND_OUT_ETHER_FROM_INTERFACE 1 WITH_DATA
   {
      T.ETHER\_MACADDRDST = CNST\_MACADDR\_BROADCAST,
      T.ETHER\_MACADDRSRC = MYMAC(1)
      T.ETHER_TYPE = \{0xA0,0x01\}
      T.[14, ] = \{0x02,0x01\}
   }
TIMER_WITH_PERIOD 5000
{
   SEND_OUT_ETHER_FROM_INTERFACE 1 WITH_DATA
   {
      T.ETHER_MACADDRDST = CNST_MACADDR_BROADCAST,
      T.ETHER\_MACADDRSRC = MYMAC(1)
      T.ETHER_TYPE
                        = \{0xA0,0x01\}
      T.[14, ] = \{0x03,0x01\}
   }
}
TIMER_WITH_PERIOD 5000
```

```
SEND_OUT_ETHER_FROM_INTERFACE 1 WITH_DATA
   {
      T.ETHER_MACADDRDST = CNST_MACADDR_BROADCAST,
      T.ETHER\_MACADDRSRC = MYMAC(1)
      T.ETHER_TYPE = \{0xA0,0x01\}
      T.[14, ] = \{0x04,0x01\}
   }
TIMER_WITH_PERIOD 5000
{
   SEND_OUT_ETHER_FROM_INTERFACE 1 WITH_DATA
   {
      T.ETHER_MACADDRDST = CNST_MACADDR_BROADCAST,
      T.ETHER\_MACADDRSRC = MYMAC(1)
      T.ETHER_TYPE
                        = \{0xA0,0x01\}
      T.[14,] = \{0x05,0x01\}
   }
}
TIMER_WITH_PERIOD 5000
{
   SEND OUT ETHER FROM INTERFACE 1 WITH DATA
   {
      T.ETHER_MACADDRDST = CNST_MACADDR_BROADCAST,
      T.ETHER\_MACADDRSRC = MYMAC(1)
      T.ETHER_TYPE = \{0xA0,0x01\}
      T.[14, ] = \{0x06,0x01\}
   }
}
```