Protocol Characteristics

- Fixed format packets
  - (more or less)
- Fairly simple
  - One packet looks like any other packet
    - As far as protocol is concerned
    - Data carried (payload) may vary
- Not all protocols are like this
- Next:
  - Look at protocols that have more complexity
    - Not necessarily more complicated

E-MAIL (RFC2822)

- Much more complex protocol than ones examined so far
- Many more options
  - much more flexible
- Much more to do.

```
Return-Path: <s4612099@maliwan.psu.ac.th>
Received: from maliwan.psu.ac.th (maliwan.psu.ac.th [202.12.73.4])
        by munnari.OZ.AU (8.12.8/8.11.6) with ESMTP id h6PEbr2O026452
        for <kre@munnari.OZ.AU>; Sat, 26 Jul 2003 00:37:57 +1000 (EST)
Received: from Santichai (santichai.std.coe.psu.ac.th [172.30.20.134])
        by maliwan.psu.ac.th (8.11.6/8.11.6) with SMTP id h6PEbo104443
        for <kre@munnari.OZ.AU>; Fri, 25 Jul 2003 21:37:50 +0700 (ICT)
Message-ID: <000e01c352ba$54e9a9a0$86141eac@Santichai>
From: "Santichai Chuaywong" <s4612099@maliwan.psu.ac.th>
To: <kre@munnari.OZ.AU>
Subject: Some mistake about lecture slide
Date: Fri, 25 Jul 2003 21:37:52 +0700
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_NextPart_000_000B_01C352F5.009425C0"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.50.4807.1700
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4910.0300
```

Defining complex protocols

- Much to specify
- One option is to use a grammar
  - similar to that used for programming languages
  - based upon grammar for natural language
- Specifies what is legal and what is not
  - with text to specify meaning
- RFC822 (and predecessors, RFC733, ...)
  - Old e-mail standard
    - recently updated/replaced
  - Contained specification of grammar
  - Used that grammar to define e-mail

ABNF (RFC2234)

- Augmented Backus-Naur Form (BNF)
  - BNF grammar defined for programming languages
  - augmented
    - usually means "with stuff added"
    - here means, completely revised...
- Used to specify exactly what is permitted
  - Consists of a a series of rules
- Each rule specifies one particular syntax element
  - from the whole specification
  - to the smallest element
    - what characters can be used
- Rules use other rules

ABNF defined in ABNF

```
rulelist       =  1*( rule / (*c-wsp c-nl) )
rule           =  rulename defined-as elements c-nl
                        ; continues if next line starts
                        ;  with white space
rulename       =  ALPHA *(ALPHA / DIGIT / "-")
defined-as     =  *c-wsp ("=" / "=/") *c-wsp
                        ; basic rules definition and
                        ;  incremental alternatives
elements       =  alternation *c-wsp
c-wsp          =  WSP / (c-nl WSP)
c-nl           =  comment / CRLF
                        ; comment or newline
comment        =  ";" *(WSP / VCHAR) CRLF
```

- First rule defines the overall specification
  - rulelist
    - List of rules gives a grammar
- Upper case (convention) are tokens
  - Syntax elements
  - defined in appendix to spec
    - represent the obvious values

ABNF Appendix A

- Core Rules

```
ALPHA          =  %x41-5A / %x61-7A   ; A-Z / a-z
BIT            =  "0" / "1"
CHAR           =  %x01-7F
                        ; any 7-bit US-ASCII character,
                        ;  excluding NUL
CR             =  %x0D ; carriage return
CRLF           =  CR LF
                        ; Internet standard newline
CTL            =  %x00-1F / %x7F
                        ; controls
DIGIT          =  %x30-39
                        ; 0-9


LF             =  %x0A ; linefeed
LWSP           =  *(WSP / CRLF WSP)
                        ; linear white space (past newline)


WSP            =  SP / HTAB
                        ; white space
```

- And several more

ABNF (cont)

```
alternation    =  concatenation
                  *(*c-wsp "/" *c-wsp concatenation)
concatenation  =  repetition *(1*c-wsp repetition)
repetition     =  [repeat] element
repeat         =  1*DIGIT / (*DIGIT "*" *DIGIT)
element        =  rulename / group / option /
                  char-val / num-val / prose-val
```

- Alternatives
  - A / B
    - Either A or B (not both)
- Sequence
  - A B C
    - A followed by B followed by C
- Repetition
  - m*nA or pA
  - a sequence of A's
    - at least m no more than n
      - defaults for m and n: 0 and infinity
    - or exactly p A's