

Common protocol components: data and interface representations



L. Bukys

Dept. of Computer Science, University of Rochester



How not to re-invent the wheel

- Many systems have need of certain oft-used components:
 - Network data representation
 - Rendezvous mechanism
 - Conventions for communication and control



Common Data Representations

- Examples:
 - XDR (eXternal Data Representation)
 - e.g. Sun NFS (Network File System)
 - ASN.1 & BER
 - e.g. SNMP (Simple Network Management Protocol)
 - XML (eXtensible Meta-Language)
- Features:
 - description language
 - \Rightarrow Code-generation of packing/unpacking (marshalling) functions



Example Data Description: XDR

```
const MAXUSERNAME = 32;
/* max length of a user name */

const MAXFILELEN = 65535;
/* max length of a file */

const MAXNAMELEN = 255;
/* max length of a file name */

/*
 * Types of files:
 */
enum filekind {
    TEXT = 0,    /* ascii data */
    DATA = 1,   /* raw data */
    EXEC = 2     /* executable */
};
```

```
/** File information, per kind of file: */
union filetype switch (filekind kind) {
    case TEXT:
        void;          /* no extra information */
    case DATA:
        string creator<MAXNAMELEN>; /* data creator */
    case EXEC:
        string interprtr<MAXNAMELEN>; /* program interpreter */
};

/** A complete file */
struct file {
    string filename<MAXNAMELEN>; /* name of file */
    filetype type;          /* info about file */
    string owner<MAXUSERNAME>; /* owner of file */
    opaque data<MAXFILELEN>; /* file data */
};
```

And there is a corresponding binary representation.



Example Data Description: ASN.1

```
struct GetRequest {
int headerOnly; // flag: do we only want headers?
int lock; // flag: should we checkout the object?
string url; // the url to fetch
AcceptTypes* acceptTypes; // Optional list of accept
    types that only apply
// to this request
};

struct AcceptTypes {
List<bitset>* standardTypes; // list of bitmaps
    indicating which
// preference order for standard types
List<string>* otherTypes; // nonstandard types in
    preference order
};
```

```
GetRequest ::= [APPLICATION 0] IMPLICIT
    SEQUENCE {
headerOnly BOOLEAN,
lock BOOLEAN,
acceptTypes AcceptTypes OPTIONAL
url OCTET STRING,

}

AcceptTypes ::= [APPLICATION 1] IMPLICIT
    SEQUENCE {
standardTypes [0] IMPLICIT SEQUENCE OF BIT
    STRING
{html(0),plain-text(1),gif(2),jpeg(3)} OPTIONAL,

otherTypes [1] IMPLICIT SEQUENCE OF OCTET
    STRING OPTIONAL
}
```

And there is a corresponding binary representation using BER (Basic Encoding Rules).



Data Description: XML

- XML Schema languages
 - DTD (Document Type Definition)
 - RELAX NG (REgular LAnguage for XML Next Generation)
 - XML Schema (W3C)
- A Validator tests conformance of an XML object to a particular schema.

And there is a corresponding binary representation using BER (Basic Encoding Rules).

Example Data Representation: XML

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
</CATALOG>
```

And there is no corresponding binary representation.



Common Interface Mechanisms

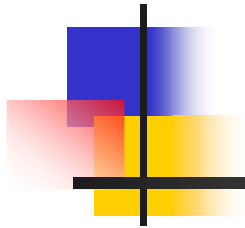
- Features:
 - Data description
 - Interface description language
 - System-provided rendezvous and language glue
 - Data marshalling
 - Control
 - exceptions



Common Interface Mechanisms

- REST (Representational State Transfer) over HTTP
- Sun (ONC) RPC [uses XDR]
- CORBA (Common Object Request Broker Architecture) [uses own IDL and runtime]
- RMI (Remote Method Invocation) [Java]
- DCOM (Distributed Component Object Model) [Microsoft]
- XML-RPC [uses XML]
- SOAP (Simple Object Access Protocol) [uses XML]

Final Review



Dept. of Computer Science, University of Rochester



TCP

- **reliable data transfer:**
 - guaranteed arrival, no error, in order
 - a sliding window protocol (combination of selective repeat and go-back-N)
- **flow controlled:**
 - sender does not overwhelm receiver
- **congestion controlled:**
 - sender does not overwhelm the network
- **connection management**
 - 3-way handshake
- **efficiency**



Network Applications

- Web and HTTP
- Electronic Mail: SMTP, POP3, IMAP
- DNS
- Internet load balancing
- Web caching
- Content-distribution networks
- Peer-to-peer networks
- Overlay networks

Multimedia Networking and Quality of Service Support



- multimedia applications and their QoS requirements
- applying application-level end-to-end techniques to enhance the "best effort" service of the Internet
- QoS support within the Internet: policing and scheduling mechanisms
- next generation Internet QoS architecture: Intserv, Diffserv



Network Security

Basic techniques.....

- cryptography (symmetric and public)
- authentication
- message integrity
- key distribution

.... network security in practice

- firewall
- attacks and countermeasures
- secure email (PGP)
- secure transport (SSL)
- secure network (IPsec)



Wireless Networks and Mobility

- Wireless networking in the data link layer
 - IEEE 802.11 wireless LAN (Media access protocol):
CSMA, ACK, RTS/CTS
- Mobility in the network layer
 - Base station model: mobile IP
 - Ad hoc model



Some Trivia Questions

- What does it mean when a socket call "send()" returns?
- In a TCP socket connection, can a message sent out by a single "send()" be received in pieces? How about UDP?
- How many open sockets can a program have?