

Modelo OSI - Camada de Apresentação

- ... a camada de apresentação é empregada para alterar a representação dos dados transmitidos via rede para fins de:
 - ❶ compatibilizar a comunicação - representação canônica de dados;
 - ❷ segurança e privacidade - criptografia;
 - ❸ compactação de dados - compressão de dados.

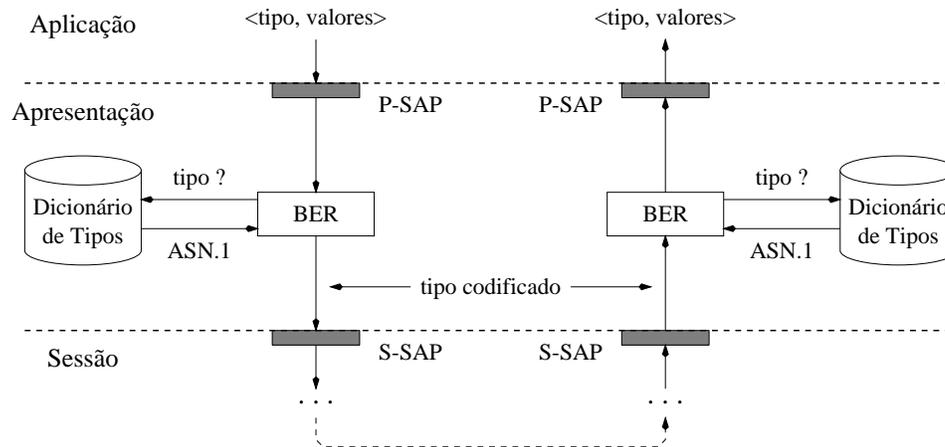
1. Representação Canônica dos Dados
2. Sintaxe ASN.1 - *Abstract Syntax Notation*
3. Sintaxe de Transferência
4. Primitivas OSI de Apresentação

1 - Representação Canônica dos Dados

- ... representar dados não é uma tarefa fácil, posto que diferentes computadores adotam representações diferentes para o mesmo dado
 - ✗ representação de texto usando EBCDIC ou ASCII;
 - ✗ sequência de *bits* em uma palavra (arquitetura de *hardware*);
- ... com o advento das redes heterogêneas, tornou-se necessário uma representação canônica de dados (ou sintaxe de transferência);
- **idéia** - dados que fluem na rede tem uma representação padronizada;
- No Modelo OSI a representação canônica de dados baseia-se numa sintaxe denominada ASN.1 (*Abstract Syntax Notation, version 1*).

2 - Sintaxe ASN.1

- ASN.1 (*Abstract Syntax Notation*) é uma linguagem formal de especificação de tipos de dados padronizada pela ISO (ISO 8824).
- Uma segunda padronização, BER (*Basic Encoding Rules*) especifica como dados na sintaxe ASN.1 são formatados numa mensagem (ISO 8825).



(cont.) 2 - Sintaxe ASN.1

- ... como toda linguagem formal, ASN.1 consiste de um conjunto básico de elementos (*tokens*) e regras de combinação desses elementos;
- ✦ **palavras** - sequências de letras maiúsculas ou minúsculas, dígitos e hífen (palavras definem tipos de dados, identificadores e palavras-chave);
- ✦ **números** - formações compostas apenas de dígitos;
- ✦ **strings** - cadeia de caracteres do tipo:
 - alfanumérico - qualquer conteúdo delimitado por "..."
 - hexadecimal - conteúdo entre 0-9 e A-F delimitado por '...'H
 - binário - 0 ou 1 delimitado por '...'B
- ✦ **pontuação** - caracteres do tipo . : = , { } < > - ' " |

2.1 - Tipos Primitivos de Dados

- ... ASN.1 define um conjunto de tipos primitivos de dados, a partir dos quais tipos mais complexos podem ser obtidos (agregação de tipos primitivos);

boolean	valores lógicos (TRUE, FALSE)
integer	números inteiros
real	números reais (mantissa, base, expoente)
bit string	sequência ordenada de 0s e 1s
octet string	sequência ordenada de bytes (octetos)
any	tipo a ser especificado
null	tipo sem valor
object descriptor	referencia algum objeto sem sintaxe formal (ex. título de comentário)
object identifier	define univocamente um objeto (protocolo de transferência de arquivo)
encrypted	tipo encriptado

(cont.) 2.1 - Tipos Primitivos de Dados

boolean	ChecksumInUse ::= BOOLEAN ChecksumInUse ::= FALSE
integer	TimeToLive ::= INTEGER TimeToLive ::= 60
real	Timeout ::= REAL Timeout ::= {250, 10, -3}
bitstring	Preamble ::= BITSTRING Preamble ::= '01010101'B
any	ForFutureUse ::= ANY
null	NoLongerInUse ::= NULL
object descriptor	Ftam ::= OBJECT DESCRIPTOR Ftam ::= "File Transfer, Access and Management"
object identifier	Ftam ::= OBJECT IDENTIFIER Ftam ::= { iso standard 8571 }

2.2 - Tipos Complexos de Dados (Construtores)

sequence	lista ordenada de tipos primitivos arbitrários
sequence of	lista ordenada de tipos primitivos homogêneos
set	lista não ordenada de tipos primitivos arbitrários
set of	lista não ordenada de tipos primitivos homogêneos
choice	conjunto de tipos dos quais apenas um deve ser considerado

```
PDUHeader ::= SEQUENCE {
    Length INTEGER,
    Flags BITSTRING,
    DestinationTSAP INTEGER,
    SourceTSAP INTEGER }
```

```
ProtocolClassOption ::= CHOICE {
    class-0 OCTETSTRING,
    class-1 OCTETSTRING,
    class-2 OCTETSTRING,
    class-3 OCTETSTRING,
    class-4 OCTETSTRING }
```

2.3 - Tipos Marcados (*Tagged*)

- ... carregam uma informação adicional para fins de eliminação de ambiguidades;

optional	identificar um campo como opcional
default	estabelece o valor que deve ser tomado caso o campo (opcional) seja omitido

```
PDUHeader ::= SEQUENCE {
    Length INTEGER,
    Flags BITSTRING OPTIONAL DEFAULT '00000000'B,
    DestinationTSAP INTEGER,
    SourceTSAP INTEGER }
```

- ... ao transmitir um PDU com o cabeçalho dado pela sequência acima, a solução é marcar explicitamente os componentes da sequência e transmitir;

```
PDUHeader ::= SEQUENCE {
    Length [0] INTEGER,
    Flags [1] BITSTRING OPTIONAL DEFAULT '00000000'B,
    DestinationTSAP [2] INTEGER,
    SourceTSAP [3] INTEGER }
```

(cont.) 2.3 - Tipos Marcados (*Tagged*)

- ... assim se o campo Flags for omitido, o receptor toma conhecimento pela ausência da marca [1] (com a transmissão da marca, o tipo fica redundante);
- ... esta redundância pode ser mantida como uma verificação adicional
- ... para eliminá-la deve-se utilizar a palavra-chave IMPLICIT;

```
PDUHeader ::= SEQUENCE {
  Length [0] IMPLICIT INTEGER,
  Flags [1] IMPLICIT BITSTRING OPTIONAL DEFAULT '00000000'B,
  DestinationTSAP [2] IMPLICIT INTEGER,
  SourceTSAP [3] IMPLICIT INTEGER }
```

- ... supondo que o protocolo possui vários tipos de PDUs (.. 9 para o TP-4) como o receptor (ao receber um PDU) descobre de que tipo se trata ?

(cont.) 2.3 - Tipos Marcados (*Tagged*)

- ... por exemplo, se o cabeçalho do exemplo anterior for do tipo 3, podemos marcar não apenas seus campos, mas também sua estrutura como um todo;

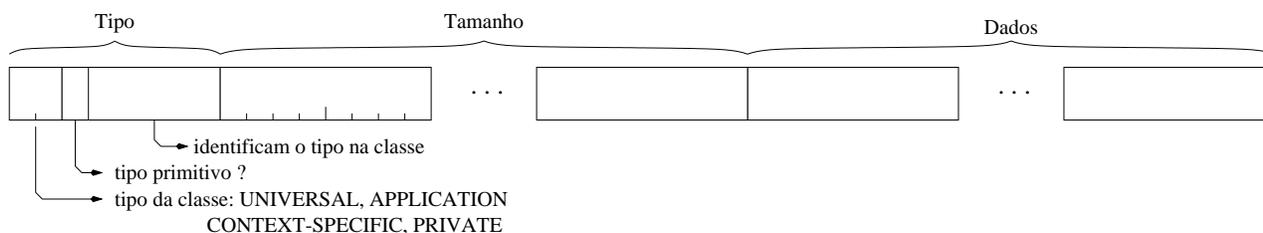
```
PDUHeader ::= [APPLICATION 3] SEQUENCE {
  Length [0] IMPLICIT INTEGER,
  Flags [1] IMPLICIT BITSTRING OPTIONAL DEFAULT '00000000'B,
  DestinationTSAP [2] IMPLICIT INTEGER,
  SourceTSAP [3] IMPLICIT INTEGER }
```

application	denota tipo de construtor no dicionário da dados da aplicação OSI
private	aplicação privada (pode substituir application)
context-specific	interpretação dependente do contexto
universal	tipos primitivos de ASN.1

3 - Sintaxe de Transferência

- ... especifica como o tipo ASN.1 é formulado numa mensagem, sendo que a norma BER (*Basic Encoding Rules*) fornece regras para esta formatação;
- BER adota inteiramente o código ASCII para caracteres
 - ✦ representações adotam o bit mais significativo a esquerda;
 - ✦ números são representados pelos seus complementos de dois.
- ... tipos (primitivos ou não) são formatados em 03 campos: a identificação do tipo, o tamanho do dado e o valor propriamente dito;
- ... caso o **tipo** não seja primitivo, o valor irá conter também outros 03 campos (tipo, tamanho, valor), ... até chegarmos unicamente a tipos primitivos.

3.1 - Identificação do Tipo, Tamanho e Valor



- ... para a classe UNIVERSAL, 1 denota BOOLEAN, 2 denota INTEGER, 3 denota BITSTRING, 4 denota OCTETSTRING, e assim por diante;
- ... caso uma classe tenha mais de 30 tipos, ativa-se todos os 5 bits do tipo seguindo-se tantos *bytes* quanto forem necessários para abrigar o índice máximo, todos começando com 1 no primeiro 1, exceto o último;
- ... se o dado é < 128 ocupa um *byte* e se > 128 , o primeiro bit é 1 e os 7 restantes estipulam quantos *bytes* a seguir armazenam o valor;

(cont.) 3.1 - Identificação do Tipo, Tamanho e Valor

- ... codificação de valor varia com o tipo - p.e. um OCTETSTRING de N *bytes* é codificado empregando-se um *byte* por caractere segundo do código ASCII;

```
número 100:    00000010  00000001  01100100
número 32768: 00000010  00000010  01000000  00000000
string 'A':   00000100  00000001  01100001
```

- ... tomemos o caso de uma SEQUÊNCIA - o *byte* que descreve o tipo indica tratar de uma sequência, o *byte* tamanho indica o número de elementos da sequência (N) e o *byte* valor dá lugar a N triplas (tipo, tamanho, valor);

```
Exemplo ::= [UNIVERSAL 7] SEQUENCE {
            elem-1 [1] IMPLICIT INTEGER,
            elem-2 [2] IMPLICIT OCTETSTRING }
```

Para valores <100,'A'>, a sequência acima é codificada como:

```
00001000  00000010  00000010  00000001  01100100  00000100  00000001  01100001
```

4 - Primitivas OSI de Apresentação

- ... as primitivas são as mesmas da camada de sessão;
- ... estas primitivas curto-circuitam a camada de apresentação, permitindo à camada de aplicação o acesso às funções da camada de sessão (tais primitivas existem para manter o conceito de camada);
- ... uma nova primitiva P-ALTER-CONTEXT - passar uma atividade de sessão para outra sem suspender a primeira;

P-CONNECT	estabelecimento de uma conexão de sessão	serv. confirmado
P-RELEASE	término de uma conexão de sessão	serv. confirmado
P-U-ABORT	termina abruptamente uma conexão	não confirmado
P-P-ABORT	informa o término abrupto de uma conexão	não confirmado

P-DATA	transferência de dados	não confirmado
P-EXPEDITED-DATA	transferência de dados expressos	não confirmado
P-TYPED-DATA	transfere dados fora de faixa	não confirmado
P-CAPABILITY-DATA	transferência de informação de controle	confirmado

(cont.) 4 - Primitivas OSI de Apresentação

P-TOKEN-GIVE	passa o token à outra entidade	não confirmado
P-TOKEN-PLEASE	solicita o token de outra entidade	não confirmado
P-CONTROL-GIVE	passa todos os tokens à outra entidade	não confirmado

P-SYNC-MAJOR	insere um ponto de sincronização maior	confirmado
P-SYNC-MINOR	insere um ponto de sincronização menor	confirmado
P-REYSINCHRONIZE	retorna a um ponto de sincronização	confirmado

P-ACTIVITY-START	inicia uma atividade	não confirmado
P-ACTIVITY-END	termina uma atividade	confirmado
P-ACTIVITY-DISCARD	descarta uma atividade	confirmado
P-ACTIVITY-INTERRUPT	interrompe uma atividade em curso	confirmado
P-ACTIVITY-RESUME	retorna uma atividade interrompida	não confirmado

P-U-EXCEPTION-REPORT	o usuário informa sobre uma exceção	não confirmado
P-P-EXCEPTION-REPORT	a entidade provedora informa sobre uma exceção	não confirmado

P-UNIT-DATA	transfere dados sem o estab/o de conexão	não confirmado
P-ALTER-CONTEXT	prevê mecanismo de mudança de contexto	confirmado

Alguns Exercícios

- ❶ Quais as funções da camada de apresentação ?
- ❷ Especifique os NPDUs do X.25/Camada 3 em ASN.1 .