

*Arquitecturas de Redes*  
*Modelos Arquitectónicos*

*FEUP/DEEC*  
*Redes de Computadores*  
*MIEEC – 2008/09*  
*José Ruela*

## *Arquitecturas de Redes*

---

- » Uma Arquitectura de Rede é um modelo abstracto que permite descrever a organização e o comportamento de uma rede e dos sistemas que a constituem
  - um modelo arquitectónico baseia-se num conjunto de princípios gerais e define regras de comportamento
  - o carácter abstracto permite descrever de forma clara e concisa conceitos e relações essenciais entre os componentes de uma rede
  
- » O modelo deve ter um carácter essencialmente funcional e permitir
  - identificar as funções necessárias à comunicação
  - organizar as funções em componentes (decompor / agrupar funções de acordo com diferenças / semelhanças ou por se basearem em mecanismos comuns)
  - relacionar (estruturar) os componentes funcionais
  - definir regras de comportamento e relações entre os sistemas e os seus componentes para efeitos de comunicação e cooperação
  
- » A partir dum modelo geral é possível especificar e desenvolver soluções particulares e concretas baseadas no modelo

## *Necessidade e objectivos*

---

- » As funções a realizar numa rede têm um elevado grau de complexidade devido a vários factores
  - grande diversidade de funções a realizar
    - ◆ as funções são de natureza e complexidade muito diferentes (do nível físico ao nível de aplicação) e podem ser realizadas de formas muito diversas, dependendo dos sistemas, das tecnologias e dos serviços disponíveis em cada momento
  - grande diversidade de aplicações a suportar, com características (tipo e volume de informação, padrões de tráfego) e requisitos (e.g., desempenho) muito diferentes
    - ◆ para garantir flexibilidade e adaptabilidade a um ambiente em permanente evolução, uma rede de uso genérico não pode (ou não deve) ser projectada e otimizada tendo em conta serviços ou aplicações particulares
  - evolução tecnológica
    - ◆ novas soluções tecnológicas devem poder ser exploradas para evitar obsolescência, melhorar o desempenho e / ou reduzir custos e devem poder ser incorporadas sem necessidade de introduzir alterações radicais (e com custos elevados) nos sistemas
  - ambiente de comunicação
    - ◆ caracterizado por sistemas heterogéneos e dispersos, que geram tráfego assíncrono (*bursty*), e que são ligados por sistemas de transmissão não totalmente fiáveis

## *Arquitecturas em Camadas*

---

- » Uma arquitetura de rede não pode ser baseada num modelo monolítico, por várias razões
  - dificuldade de concepção e de desenvolvimento
  - dificuldade de manutenção e de alteração (e.g, para acompanhar a evolução tecnológica)
  - inflexibilidade (dificuldade de aplicar a situações diferentes ou a novas situações)
  
- » A solução consiste em decompor o problema global e complexo num conjunto de problemas mais simples e tratáveis (modularidade), permitindo assim uma abordagem sistemática, com elevado grau de flexibilidade e adaptabilidade
  
- » Os modelos arquitectónicos que têm sido adoptados em redes baseiam-se na organização das funções em módulos e na sua estruturação hierárquica, de que resultam Arquitecturas em Camadas (*Layered Architectures*)

## *Arquitecturas em Camadas – Princípios*

---

Arquitecturas em Camadas baseiam-se em três princípios

- » Independência entre camadas – uma camada encapsula as funções que realiza, não sendo visível do exterior da camada a forma como essas funções são realizadas (mas apenas o serviço que oferece)
- » Camadas adjacentes comunicam através duma interface – a camada inferior oferece um serviço à camada superior através da interface
- » Valorização dos serviços – o serviço oferecido por uma camada à camada superior acrescenta valor ao serviço recebido da camada inferior

## *Arquitecturas em Camadas – Vantagens*

---

- » Redução da complexidade de concepção, desenvolvimento e manutenção
- » Possibilidade de desenvolvimentos independentes das várias camadas, o que pressupõe a definição das interfaces entre camadas (e os serviços associados)
- » Flexibilidade de implementação, visto ser possível escolher as tecnologias e os algoritmos de controlo mais adequados a cada função ou grupo de funções
- » Possibilidade de introduzir alterações numa camada (e.g., para explorar novas tecnologias entretanto disponíveis ou algoritmos de controlo mais eficientes)
- » Possibilidade de suportar diferentes aplicações com base num número reduzido de interfaces (serviços) comuns
- » Concepção e análise de sistemas complexos com diferentes graus de abstracção
- » Adopção de *standards*, o que permite a produção em massa (com a consequente redução de custos) e o suporte de produtos por diferentes fabricantes (o que aumenta a diversidade de escolha e a flexibilidade das soluções)

## *Regras de Comunicação – Protocolos*

---

- » Numa arquitectura em camadas os sistemas que constituem a rede aparecem funcionalmente organizados e estruturados em camadas – uma camada atravessa transversalmente todos os sistemas
- » Uma camada é constituída por entidades (processos, recursos) responsáveis pela realização das funções específicas dessa camada
- » Entidades de uma mesma camada (*peer entities*) residentes em sistemas diferentes cooperam para construir o serviço oferecido pela camada – o que requer a troca de mensagens de controlo e de sincronização, para além de mensagens que contêm dados (de camadas superiores)
- » Esta comunicação pressupõe regras, isto é, um Protocolo
- » Numa arquitectura em camadas, os Protocolos aparecem igualmente estruturados em camadas – a comunicação entre dois sistemas pode então ser decomposta e descrita com base na comunicação que ocorre em cada camada

## *Protocolo, Interface e Serviço*

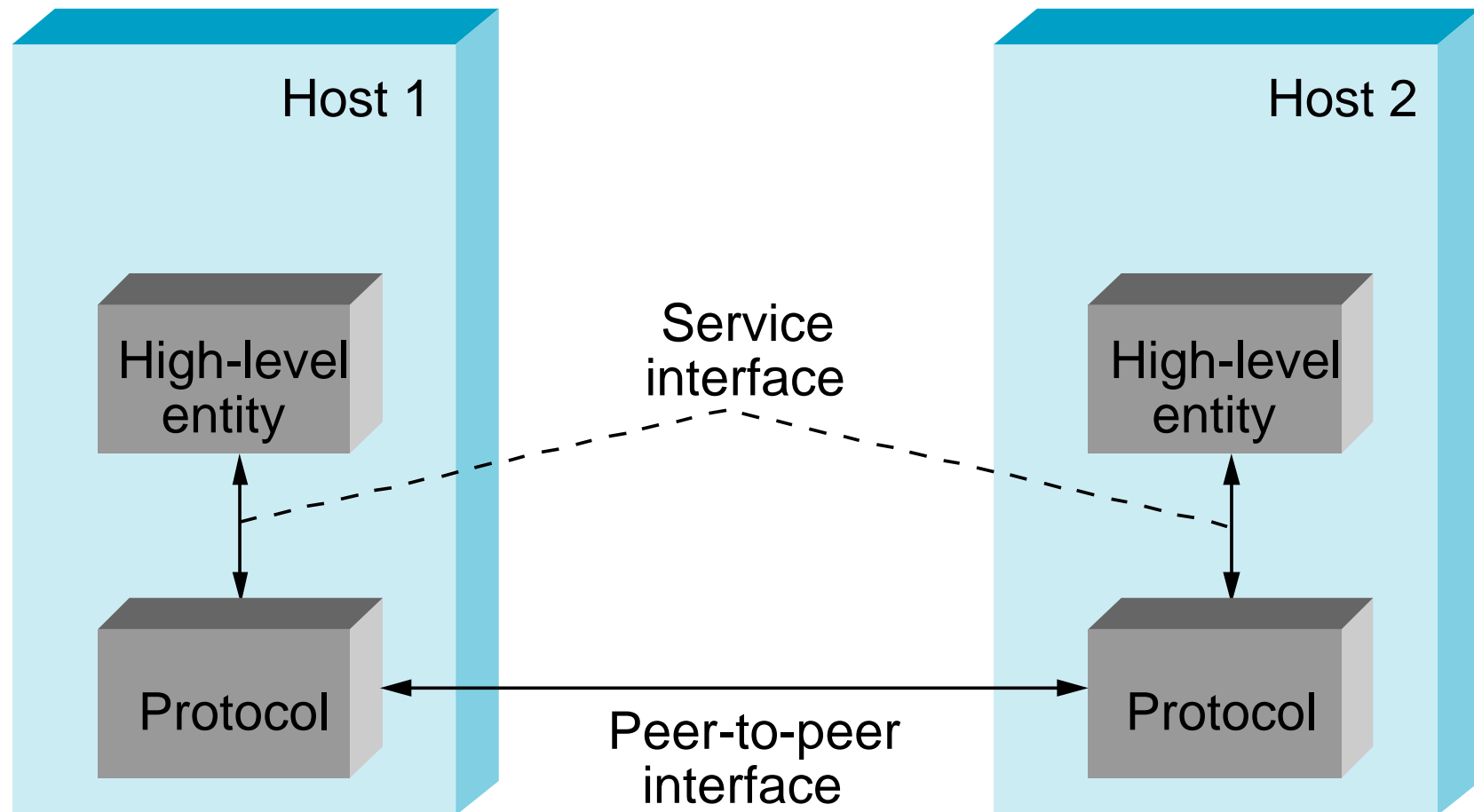
---

- » Um Protocolo define um conjunto de regras de comunicação entre entidades homólogas (*peer entities*) que residem em sistemas diferentes – um protocolo define uma interface (*peer interface*) entre estas entidades
  - Com excepção da camada protocolar mais baixa, esta comunicação (horizontal) é lógica (ou virtual), pois as entidades de uma camada comunicam usando o serviço da camada inferior (e este princípio é aplicado recursivamente)
- » Elementos de um protocolo
  - Um protocolo inclui elementos de natureza sintática, semântica e temporal
  - Um protocolo deve definir a sintaxe (formatos) e a semântica (significado) das mensagens trocadas entre as entidades protocolares, fornecer mecanismos de sincronização (que permitam garantir um determinado comportamento temporal) e especificar as acções (procedimentos) a executar aquando da ocorrência de acontecimentos, tendo em atenção o estado do sistema (um protocolo pode ser descrito por uma máquina de estados)
- » Um serviço é disponibilizado através de uma interface (*service interface*) e requer a interacção entre entidades residentes em camadas adjacentes no mesmo sistema
  - Esta comunicação (vertical) corresponde ao fluxo real da informação no sistema
- » Uma arquitectura fica caracterizada pelo tipo de estruturação e pela definição das funções, dos protocolos e dos serviços de cada camada



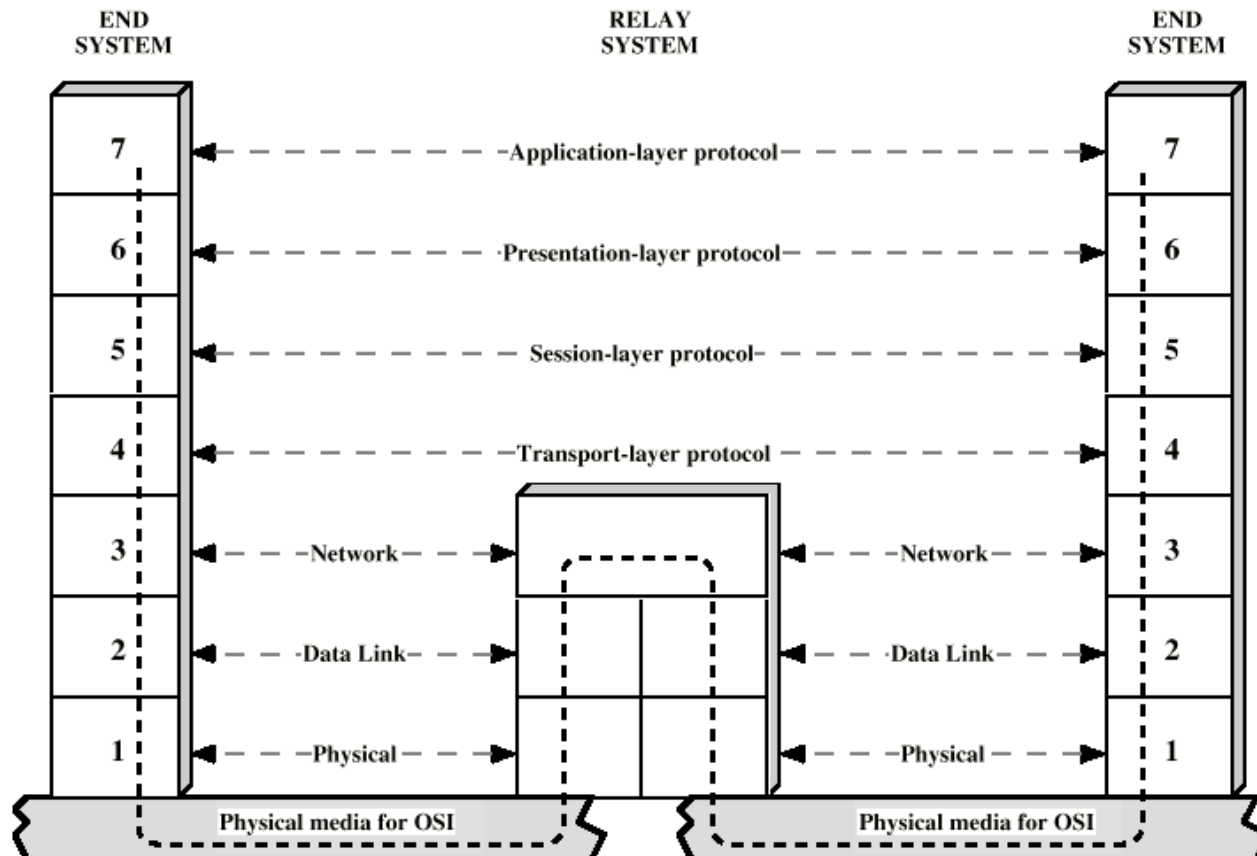
# *Protocolo, Interface e Serviço*

---



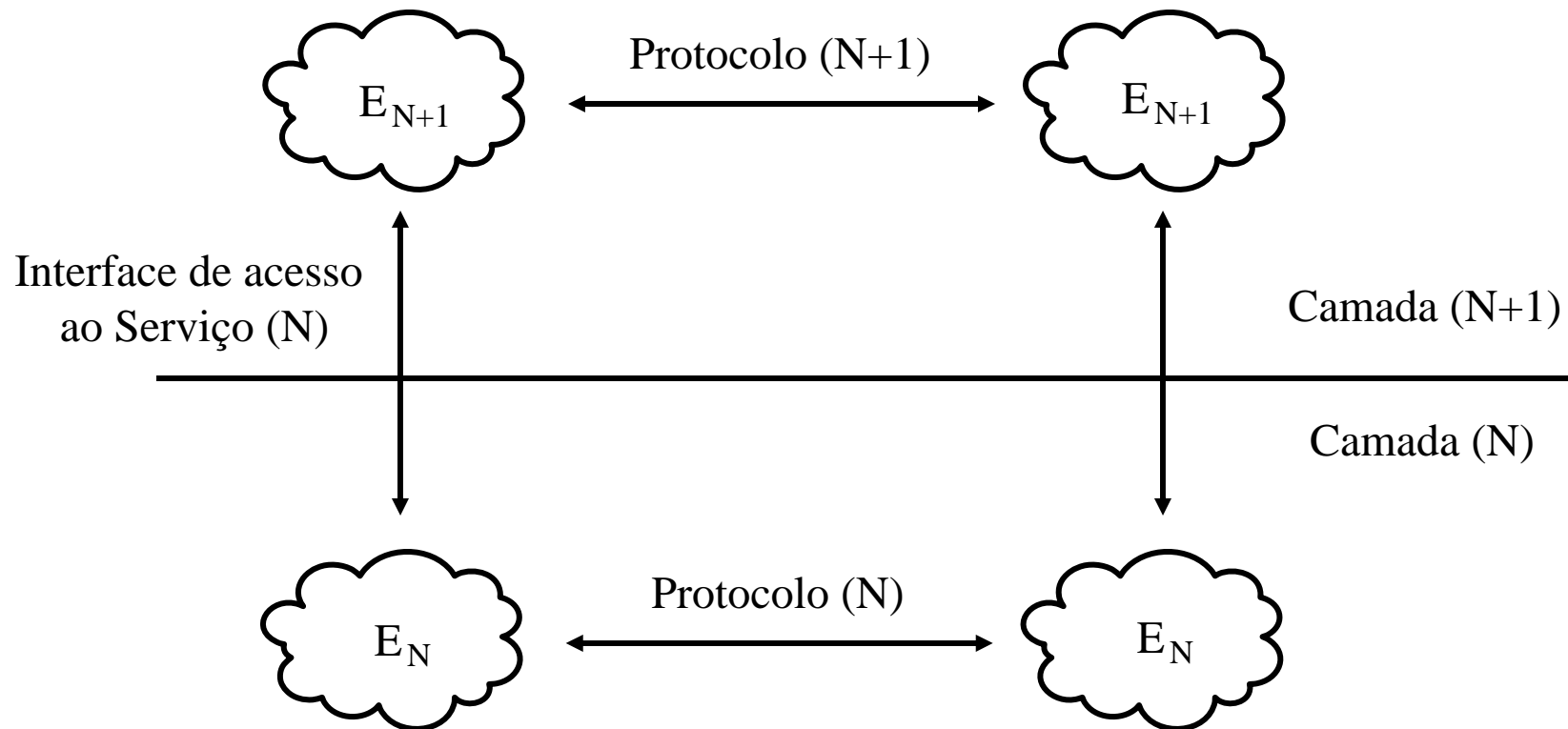
## *Fluxo de dados e comunicação protocolar*

Tomando como referência um modelo em camadas (OSI), a figura ilustra o fluxo real dos dados na comunicação entre dois sistemas (através de um sistema intermédio) e a comunicação lógica entre entidades das várias camadas



# *Modelo de comunicação*

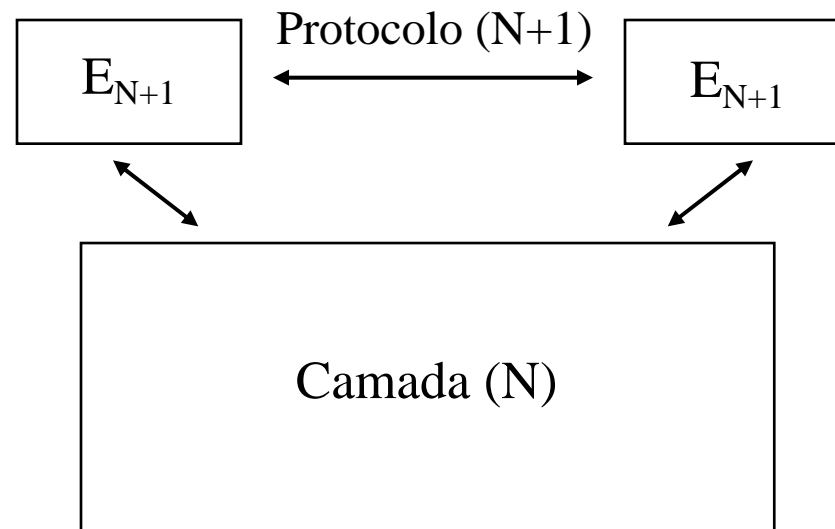
---



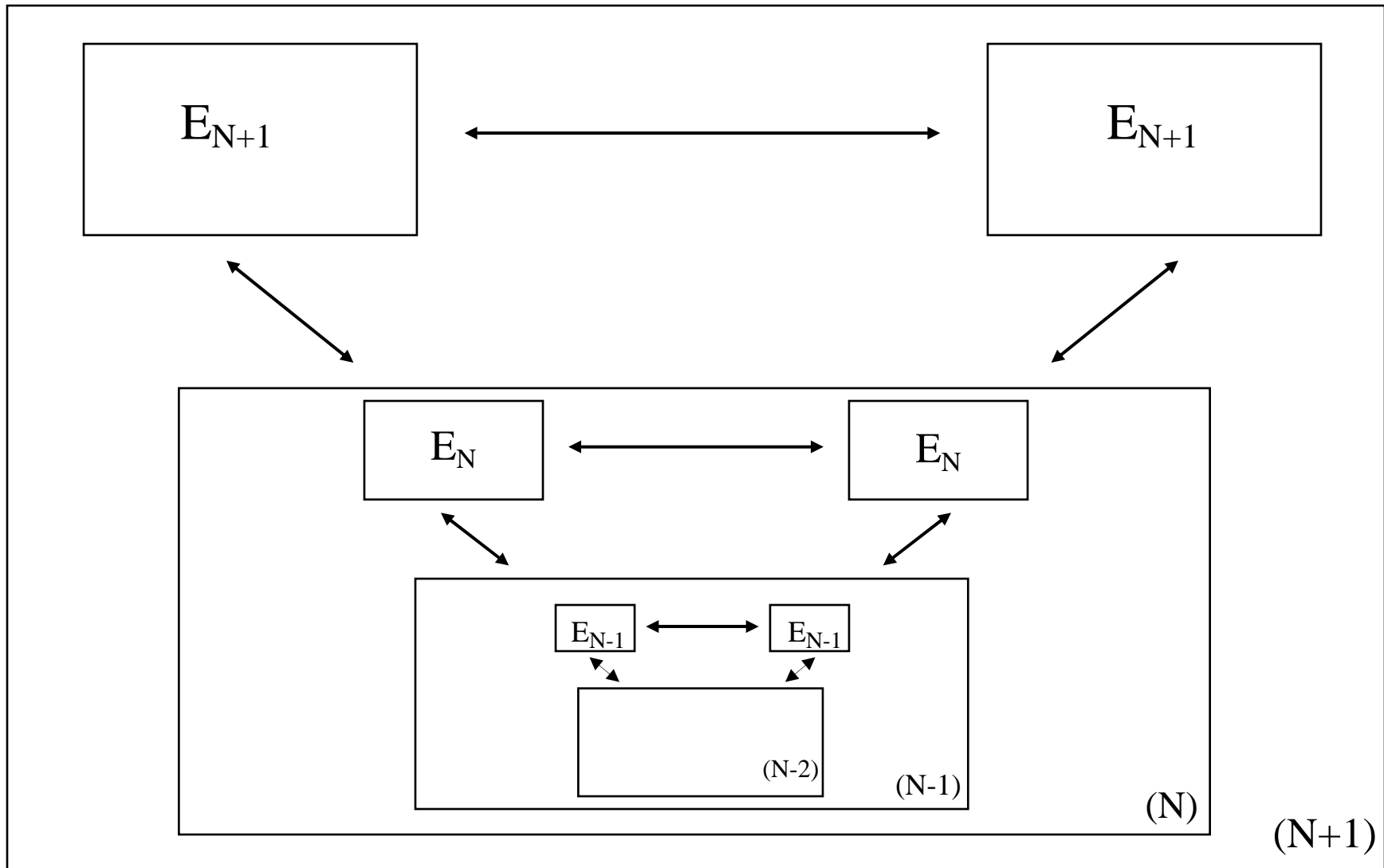
## *Serviço*

---

- » Entidades (N+1) comunicam por meio dum protocolo (N+1) usando o serviço disponibilizado pela camada (N) através da interface entre as duas camadas
- » A forma como o serviço é realizado não é visível do exterior da camada (N)
- » Os serviços são usados recursivamente pelas várias camadas



# Utilização Recursiva de Serviços



## *Modelo de Referência – necessidade*

---

- » Os fabricantes de computadores começaram por desenvolver arquitecturas próprias com o objectivo de permitir a ligação em rede dos seus sistemas
  - embora baseadas em princípios e conceitos semelhantes caracterizavam-se por diferenças irreconciliáveis, no que se refere ao número de camadas, funções e protocolos de cada camada, tipo de controlo e serviços disponibilizados
- » Por outro lado, começaram a implantar-se redes públicas de comunicação de dados, baseadas em diferentes tecnologias, protocolos de acesso e serviços
- » Esta situação tornava os utilizadores muito dependentes das soluções de um único fabricante e tornava difícil explorar os serviços entretanto oferecidos pelos operadores de redes (em alternativa ao aluguer de circuitos)
- » Alternativas a soluções específicas de cada fabricante (fechadas) exigiam
  - recurso a redes públicas de dados, usando protocolos e interfaces (serviços) normalizados
  - interligação de equipamentos de diferentes fabricantes usando protocolos universais
- » Surgiu assim naturalmente a necessidade de um Modelo Arquitectónico de Referência – papel que veio a ser desempenhado pelo Modelo de Referência de Sistemas Abertos (Modelo OSI) desenvolvido pela ISO

## *Modelo de Referência de Sistemas Abertos (OSI)*

---

- » O Modelo de Referência de Sistemas Abertos (*Open Systems Interconnection Reference Model*) define regras gerais de interacção entre sistemas abertos, isto é, sistemas que obedecem a normas universais de comunicação (por oposição a sistemas fechados) e cujo comportamento externo está de acordo com o prescrito pelo modelo (princípio da visibilidade restrita)
- » O Modelo OSI cria as bases para a especificação e aprovação de *standards* por organizações de normalização reconhecidas internacionalmente – embora os *standards* não façam parte do modelo
- » O Modelo OSI define princípios, conceitos e relações entre componentes – é um modelo abstracto da descrição da comunicação entre sistemas (e não um modelo de implementação)
- » O Modelo OSI é geral e flexível – embora definido no contexto das redes de computadores que se desenvolveram durante a década de 1970, continua a ser usado como modelo de descrição de redes e serviços que se desenvolveram desde então

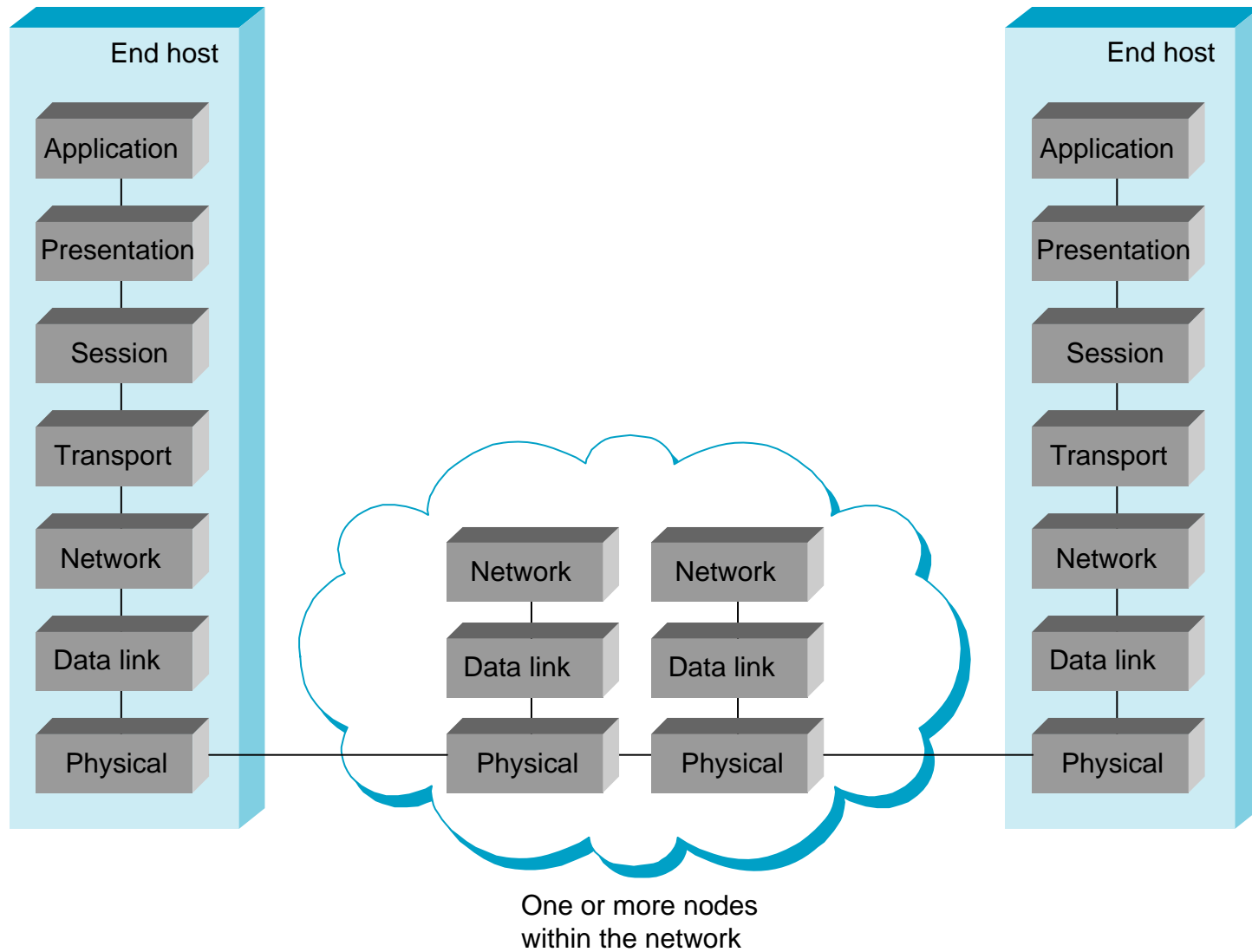
## *Modelo OSI – princípios e conceitos*

---

- » O Modelo OSI propõe uma organização funcional em sete camadas, de acordo com os seguintes princípios
  - As funções são decompostas e organizadas em camadas
  - Cada camada realiza um conjunto de funções relacionadas, suportadas num protocolo
  - Cada camada fornece serviços à camada superior escondendo-lhe os detalhes de implementação
  - Cada camada usa serviços da camada inferior
  - Mudanças internas numa camada não implicam mudanças nas outras camadas
  
- » O Modelo OSI não se pode reduzir a esta visão simplificada de sete camadas protocolares – pois inclui um conjunto extremamente rico de conceitos e princípios, nomeadamente
  - Princípios de estruturação em camadas
  - Modelo e Tipos de Serviço
  - Descrição das Funções a suportar pelos Protocolos das diferentes camadas
  - Princípios de Endereçamento



# Camadas OSI



## *Camadas OSI (1-3)*

---

- ◆ Física
  - » Características mecânicas, eléctricas e funcionais da interface física entre sistemas (conectores, níveis de sinal, códigos de transmissão, sincronização, etc.)
  - » Exemplos: RS-232, V.24, X.21
  
- ◆ Ligação de dados
  - » Estabelecimento, manutenção e terminação de uma ligação de dados
  - » Encapsulamento de dados em tramas para transmissão
  - » Controlo de Fluxo e Controlo de Erros (no caso de ligação fiável)
  - » Exemplos: HDLC, LAPB (X.25), LAPD (Canal D / RDIS), LAPF (*Frame Relay*), PPP (IP), LLC (LANs)
  
- ◆ Rede
  - » Transferência de informação (multiplexagem e comutação) entre nós da rede
  - » Encaminhamento de pacotes através da rede
  - » Serviço independente da tecnologia e dos serviços nativos de subredes físicas
  - » Exemplos: X.25, IP (*internetworking*)

## *Camadas OSI (4-7)*

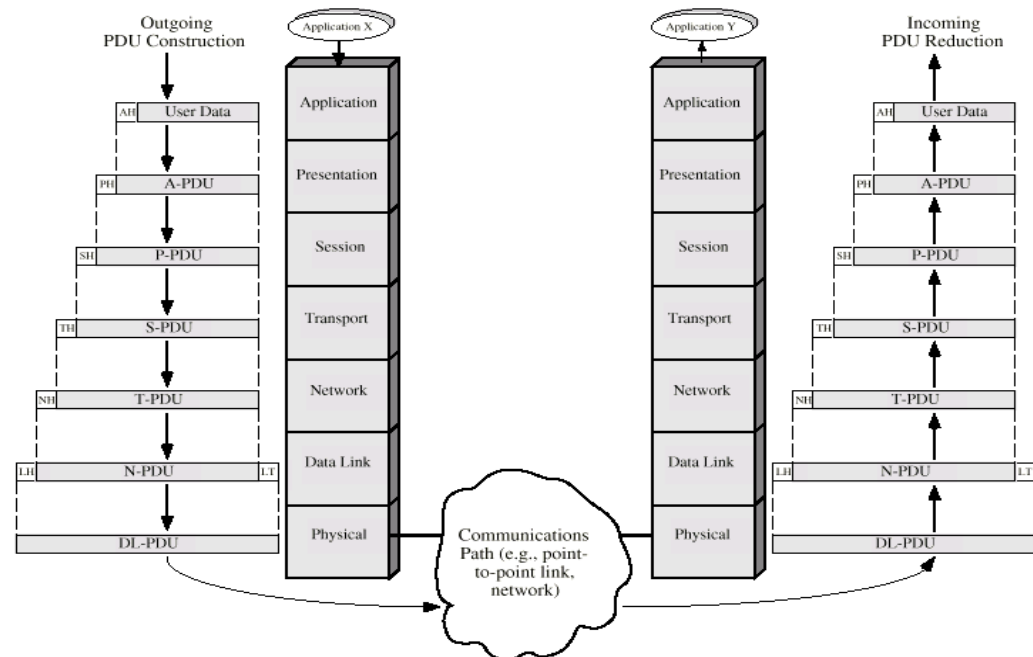
---

- ◆ Transporte
  - » Transferência de informação extremo a extremo entre equipamentos terminais
  - » Serviço independente do serviço de Rede (ou dos serviços nativos de subredes)
  - » Adaptação ao serviço de Rede (fragmentação, multiplexagem de fluxos de dados)
  - » Eventualmente Controlo de Erros (serviço fiável) e Controlo de Fluxo
  - » Exemplos: TCP (fiável), UDP (não fiável)
- ◆ Sessão
  - » Controlo do diálogo entre processos e mecanismos de sincronização
- ◆ Apresentação
  - » Representação de informação (formatos, códigos) independente do conteúdo
  - » Resolução de diferenças sintácticas e negociação da sintaxe de transferência
- ◆ Aplicação
  - » Criação do ambiente para comunicação entre aplicações (aspectos semânticos)
  - » Aplicações genéricas (transferência de ficheiros, correio electrónico, etc.)
  - » Funções de gestão

## *Comunicação em ambiente OSI*

A comunicação entre uma Aplicação X e uma Aplicação Y em sistemas diferentes pode ser descrita pela seguinte sequência

- » Para comunicar com a Aplicação Y, a Aplicação X usa os serviços da camada 7
- » As entidades da camada 7 de X comunicam com as entidades da camada 7 de Y usando um protocolo da camada 7
- » O protocolo da camada 7 usa os serviços da camada 6
- » ... e assim sucessivamente



## *Modelo de Serviço*

---

- » Um serviço é definido de forma abstracta como um conjunto de capacidades disponibilizadas por uma camada (Fornecedora do serviço) à camada adjacente superiora (Utilizadora do serviço)
  - a descrição do serviço inclui apenas os aspectos semânticos do serviço e não a forma como é realizado
  
- » Um serviço é descrito por um conjunto de atributos e a sua descrição inclui
  - a interacção através da interface
  - os dados associados à interacção
  - a relação entre eventuais interacções nos vários sistemas envolvidos no serviço
  
- » A interacção entre Utilizadores e Fornecedores de um serviço é descrita por meio de Primitivas de Serviço (operações elementares e indivisíveis) que
  - indicam uma acção (realizada ou a realizar) ou o seu resultado
  - fornecem parâmetros (endereços, Qualidade de Serviço, controlo de fluxo, negociação de facilidades, etc.)

# Primitivas de Serviço

## *Request (Pedido)*

- » Invocada pelo utilizador do serviço
- » Invoca um serviço (acção, procedimento) especificado por meio de parâmetros

## *Indication (Indicação)*

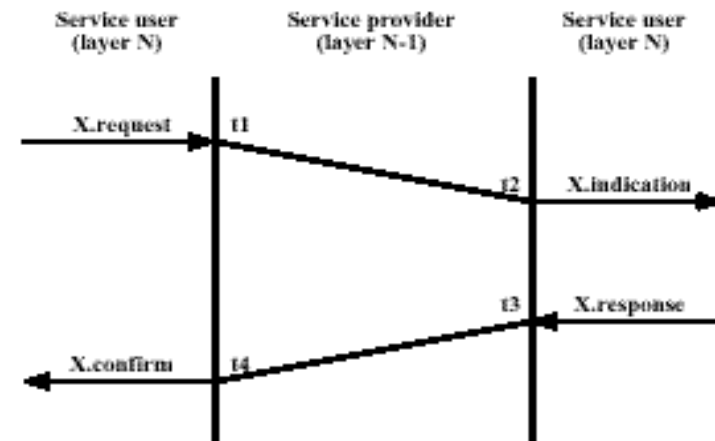
- » Invocada pelo fornecedor do serviço
- » Indica que um serviço foi invocado pelo utilizador remoto (*peer*) ou notifica o utilizador de um acontecimento ou de uma acção iniciada pelo fornecedor do serviço

## *Response (Resposta)*

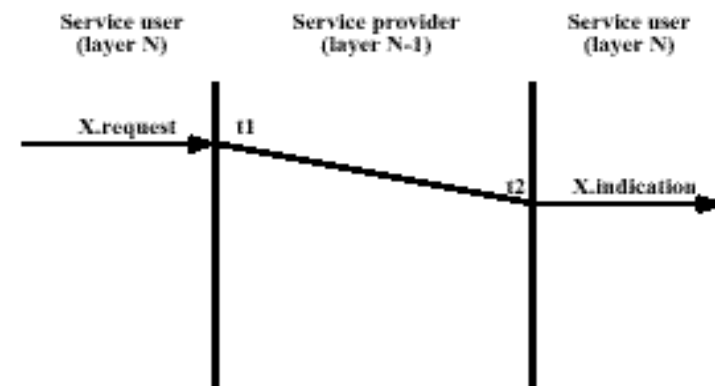
- » Invocada pelo utilizador do serviço
- » Resposta a uma Primitiva *Indication*

## *Confirmation (Confirmação)*

- » Invocada pelo fornecedor do serviço
- » Indica que foi accionado ou se completou um serviço invocado pelo utilizador (*Request*)



(a) Confirmed Service



(b) Nonconfirmed Service

## *Pontos de Acesso ao Serviço (SAPs)*

---

- » Os serviços da camada N são oferecidos a Entidades (N+1) em Pontos de Acesso ao Serviço – (N)-SAPs (*Service Access Points*)
  
- » Um (N)-SAP constitui a interface lógica entre Entidades (N) e (N+1)
  - um (N)-SAP é servido por uma e uma só Entidade (N) e é usado por uma e uma só Entidade (N+1)
  - uma Entidade (N) pode servir vários (N)-SAPs e uma Entidade (N+1) pode usar vários (N)-SAPs
  
- » Um (N)-SAP é identificado por um endereço-(N) – (N)-*address* – que o identifica univocamente na interface entre as camadas (N) e (N+1)

## *Tipos de Serviço*

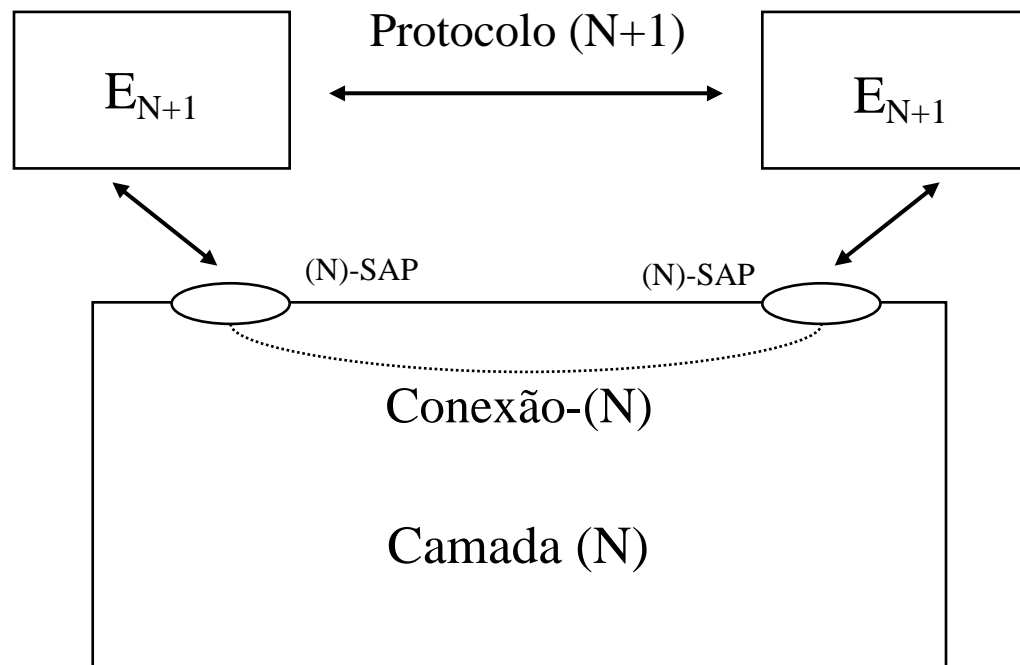
---

- » O modelo OSI define dois tipos de serviço
  - Serviços orientados à conexão (CO – *Connection Oriented*)
  - Serviços sem conexão (CL – *Connectionless*)
  
- » Uma conexão-(N) – (N)-*connection* – é uma associação estabelecida pela camada (N) para a transferência de dados entre duas ou mais Entidades (N+1)
  - uma conexão pode ser descrita como uma associação lógica entre (N)-SAPs
  
- » Um conexão-(N) é estabelecida, mantida e terminada por meio de um protocolo da camada (N)
  - a conexão tem um identificador único atribuído, que tem de estar presente nas mensagens trocadas pelas entidades protocolares da camada (e.g., VCI)
  - uma conexão é terminada localmente num SAP – essa terminação designa-se por *Connection End Point* (CEP); várias conexões podem terminar no mesmo SAP
  - uma conexão é univocamente identificada na interface entre camadas por um par de identificadores (SAPI, CEPI)

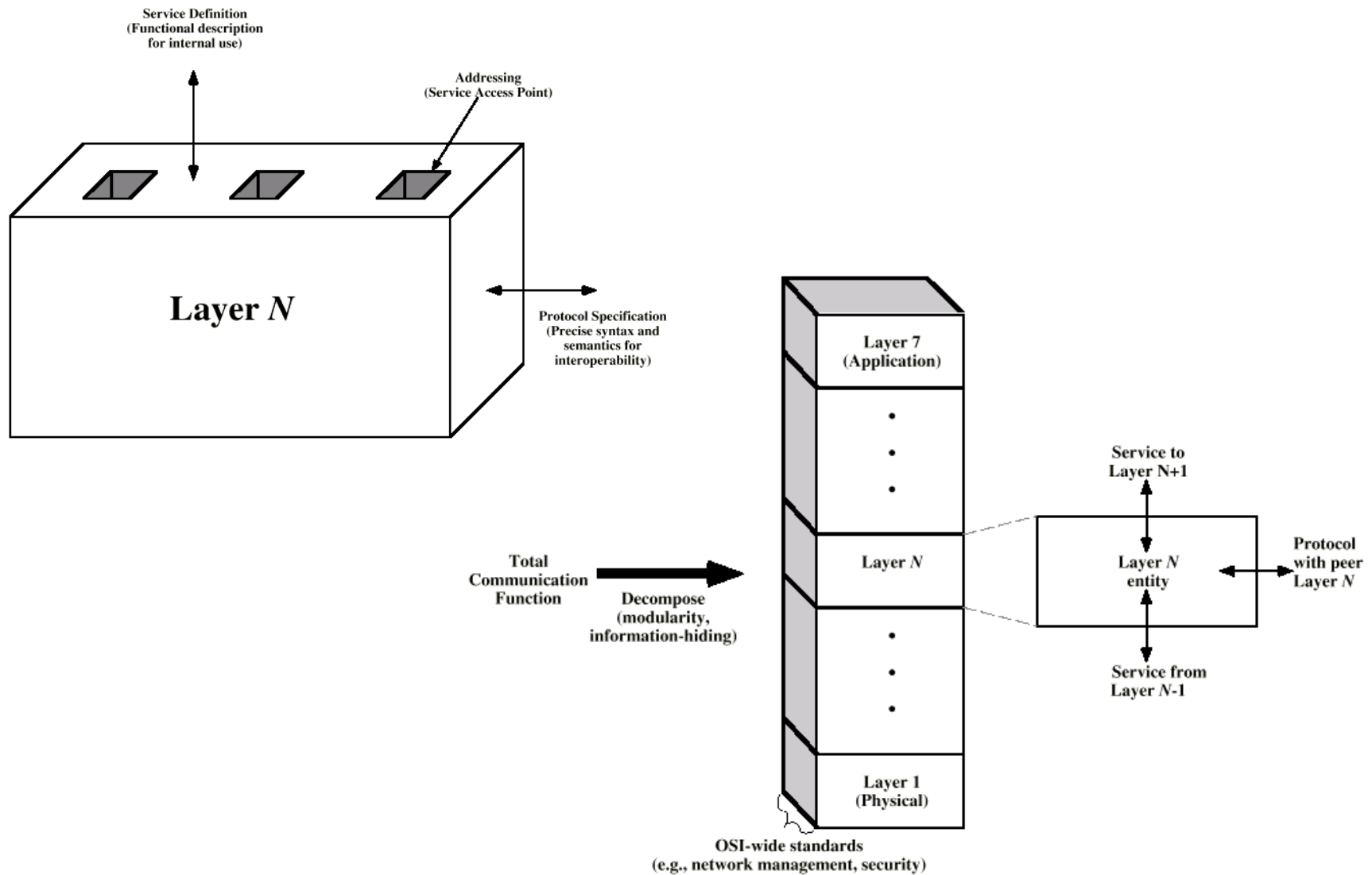


# *SAPs e Conexões*

---



# Modelo OSI – Protocolos e Serviços



## *Funções Protocolares*

---

As Funções a realizar pelos Protocolos são naturalmente diferentes de camada para camada, podendo no entanto haver funções idênticas realizadas em mais do que uma camada, embora em contextos e com objectivos diferentes

### Funções típicas

- » Encapsulamento de dados
- » Segmentação e reassemblagem de dados
- » Controlo de ligações (conexões)
- » Entrega ordenada de dados
- » Controlo de erros
- » Controlo de fluxo
- » Controlo de congestionamento
- » Endereçamento
- » Multiplexagem

## *Unidades de Dados*

---

- » As unidades de dados transferidas através da interface entre a camada (N+1) e a camada (N) designam-se por Unidades de Dados de Serviço-(N) e estão relacionadas com as necessidades dos utilizadores do serviço

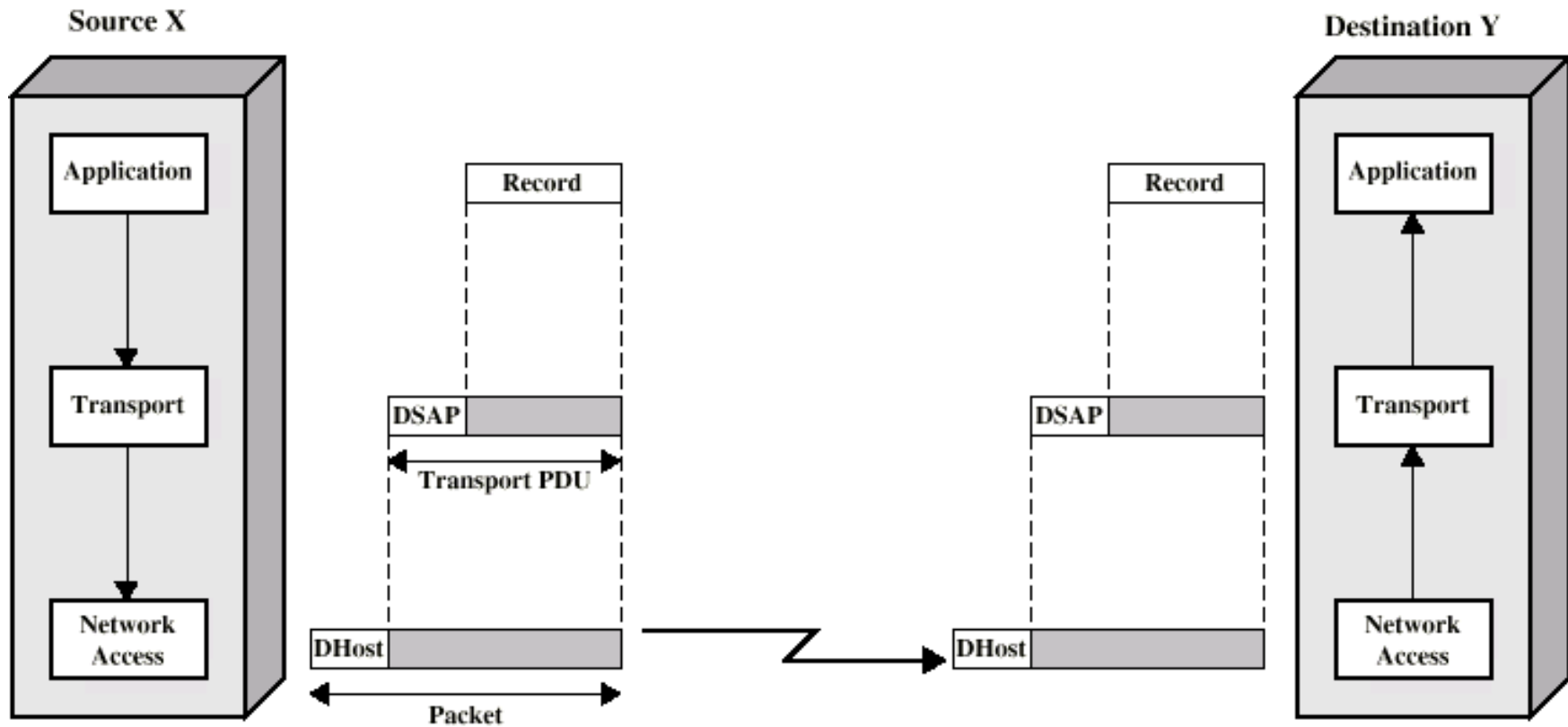
(N)-SDU – *Service Data Unit*

- » As unidades de dados trocadas entre entidades protocolares na camada (N) designam-se por Unidades Protocolares de Dados-(N) e estão relacionadas com a operação do protocolo

(N)-PDU – *Protocol Data Unit*

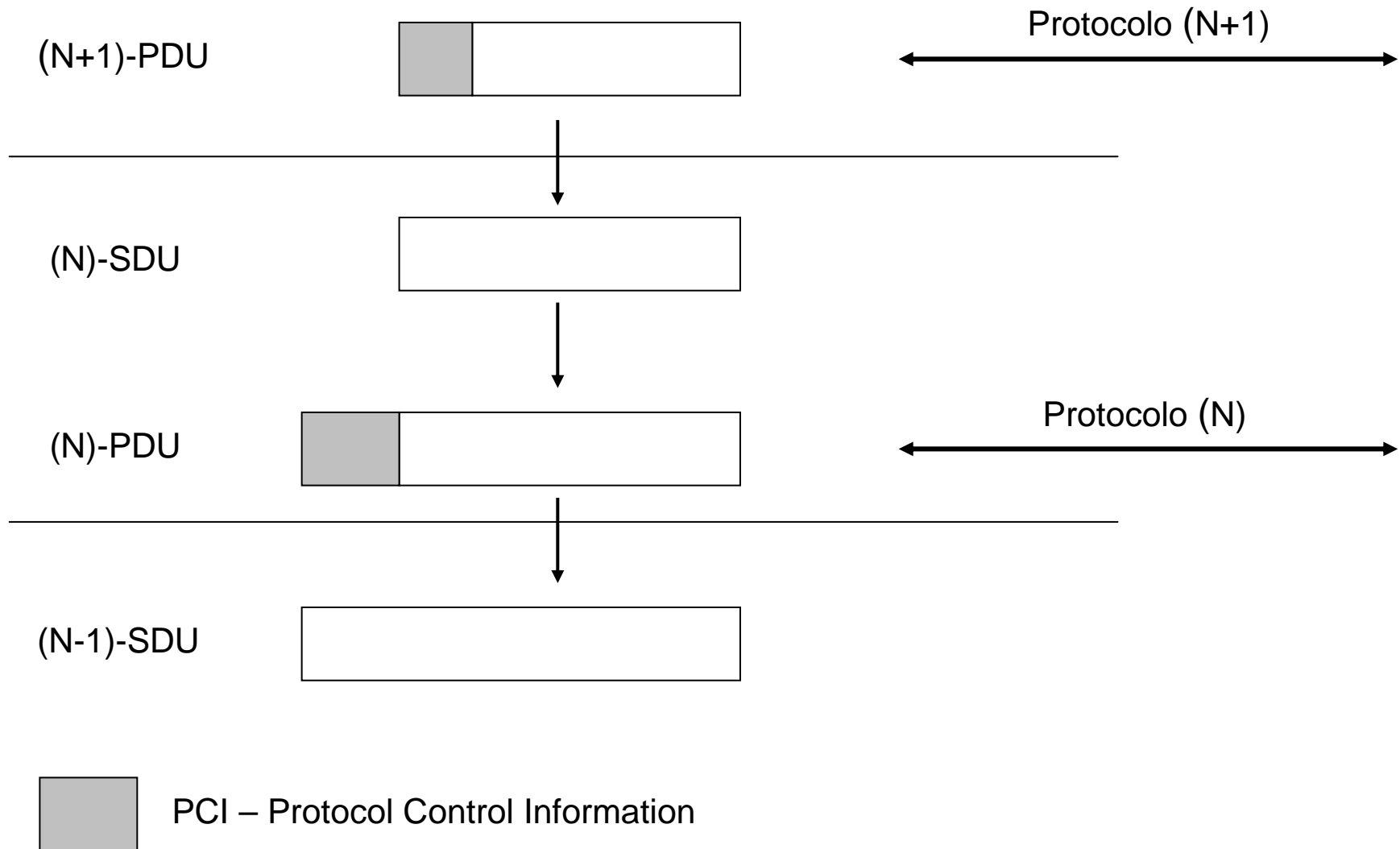
- » No caso mais simples a um (N)-SDU corresponde um (N)-PDU – o protocolo da camada (N) forma um (N)-PDU encapsulando o (N)-SDU com informação adicional, que pode incluir
  - informação de controlo do protocolo (PCI – *Protocol Control Information*)
  - endereços (ou outros identificadores)
  - código para detecção de erros

# Encapsulamento



# *PDU*s e *SDU*s

---



## *Relações possíveis entre PDUs e SDUs*

---

- » No caso básico exemplificado, um (N)-SDU é encapsulado com (N)-PCI, formando um único (N)-PDU
- » Um (N)-SDU pode ser segmentado e encapsulado em vários (N)-PDUs
  - É necessário providenciar informação no (N)-PCI de cada (N)-PDU de forma a permitir a reconstituição no destino do (N)-SDU original
  - Estas operações designam-se *Segmentation e Reassembly*
- » Vários (N)-SDUs podem ser agrupados e transportados num único (N)-PDU
  - É necessário providenciar informação no (N)-PCI do (N)-PDU de forma a permitir a extracção no destino dos (N)-SDU originais
  - Estas operações designam-se *Blocking e Deblocking*
- » Em ambos os casos, estas operações têm lugar numa única camada protocolar, ao contrário do que ocorre num terceiro caso em que vários (N)-PDUs podem ser concatenados para formar um (N-1)-SDU
  - Estas operações designam-se *Concatenation e Separation*

## *Segmentação e Reassemblagem*

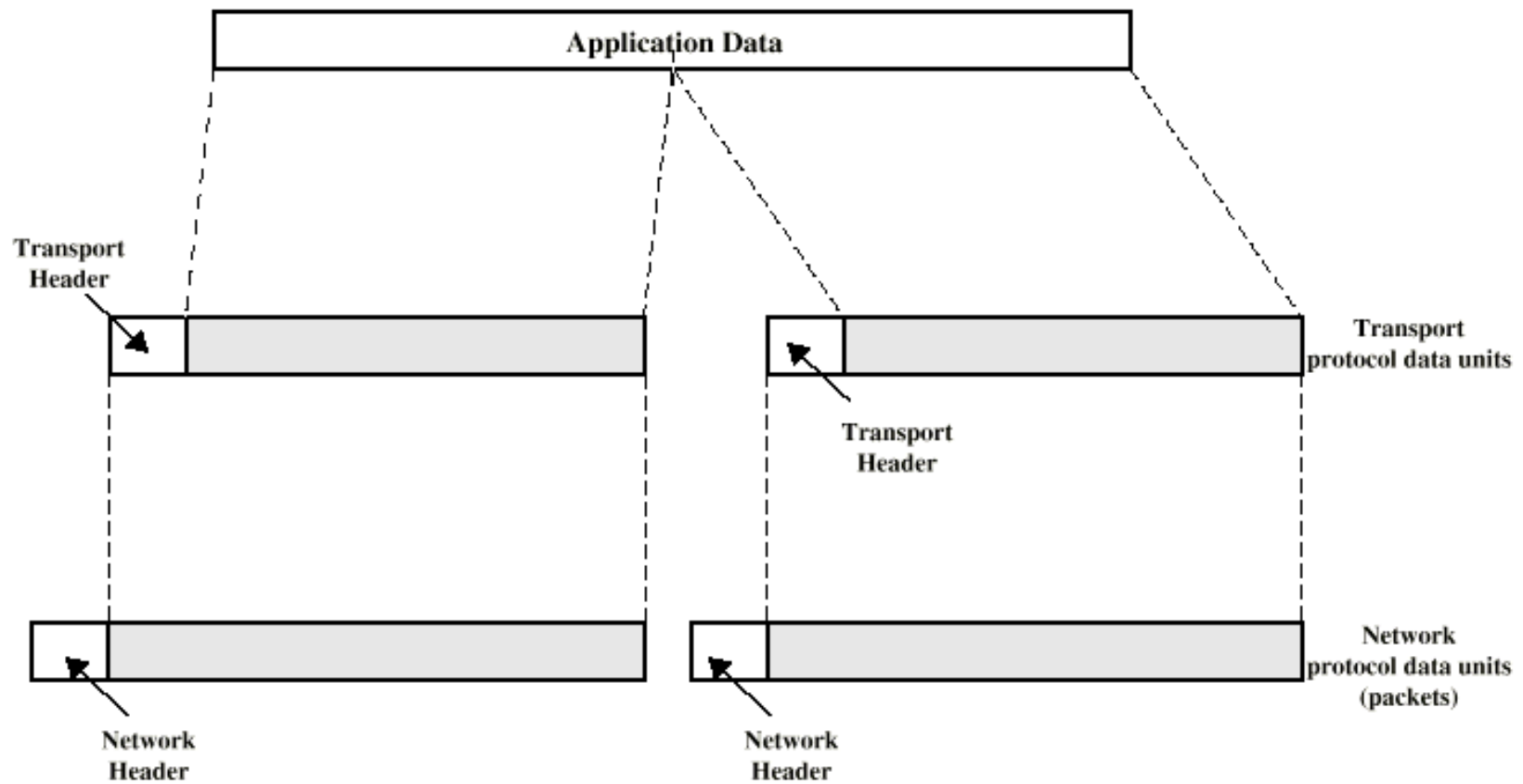
---

- » Uma camada protocolar pode ter necessidade de fragmentar (segmentar) as unidades de dados (SDUs) recebidas da camada superior, transportando-as, após encapsulamento, em vários PDUs
- » Para ser possível reconstituir no destino as unidades de dados iniciais (SDUs) é necessário que a informação de controlo (PCI) acrescentada pelo protocolo permita relacionar os segmentos (por exemplo, por meio de números de sequência e indicação do último segmento)
- » Razões para segmentar
  - Controlo de erros mais eficiente
  - Acessos mais equilibrados à rede
  - Atrasos menores na rede
  - *Buffers* mais pequenos nos nós da rede
- » Desvantagens
  - *Overheads* adicionais (maior número de PDUs e mais informação de controlo por PDU)
  - Mais interrupções nos processadores (tipicamente uma por segmento)
  - Tempos de processamento superiores (funções adicionais a realizar)



# Segmentação e Reassemblagem

---



## *Modos de operação dos protocolos*

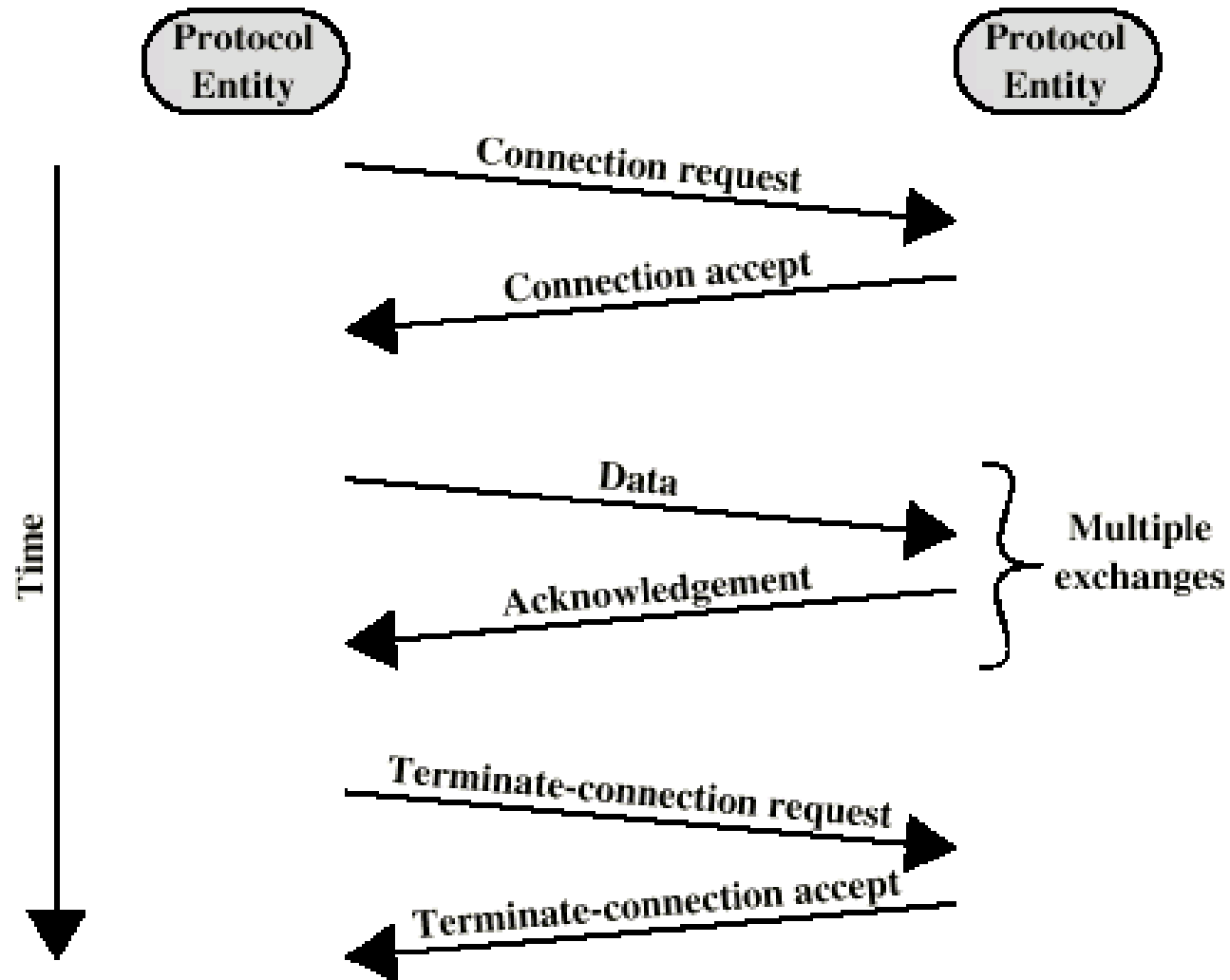
---

A comunicação entre entidades protocolares na mesma camada pode ser realizada de dois modos

- » Não-orientado à conexão (*connectionless*) – as unidades de dados (PDUs) são transportadas de forma independente, sem necessidade de estabelecimento prévio de qualquer associação lógica (conexão) entre as entidades protocolares
- » Orientado à conexão (*connection oriented*) – a transferência de unidades de dados (PDUs) só é possível após o estabelecimento de uma conexão entre as entidades protocolares
  - Uma conexão-(N) é estabelecida pelo protocolo da camada (N), usando PDUs de controlo (*Connection Request*), após invocação de uma Primitiva de Serviço do tipo (N)-*Connect Request* por uma entidade da camada (N+1)
  - Fases de uma conexão: estabelecimento, transferência de dados, terminação

## *Fases de uma conexão*

---



# *Ordenação de PDUs, Controlo de Fluxo e de Erros*

---

## » Ordenação de PDUs

- Protocolos orientados à conexão garantem que os PDUs chegam ordenados ao destino
- Se os PDUs seguirem trajectos diferentes na rede podem chegar ao destino desordenados, podendo ser reordenados, se necessário
- PDUs são numerados sequencialmente para se poder garantir a sua (re)ordenação

## » Controlo de Fluxo

- Objectivo: limitar (controlar) o débito do emissor
- Pode ser realizado pelo receptor (e.g., *window control*) ou pelo emissor (e.g., *rate control*)
- Pode ser necessário em várias camadas protocolares (nó a nó ou extremo a extremo)

## » Controlo de Erros

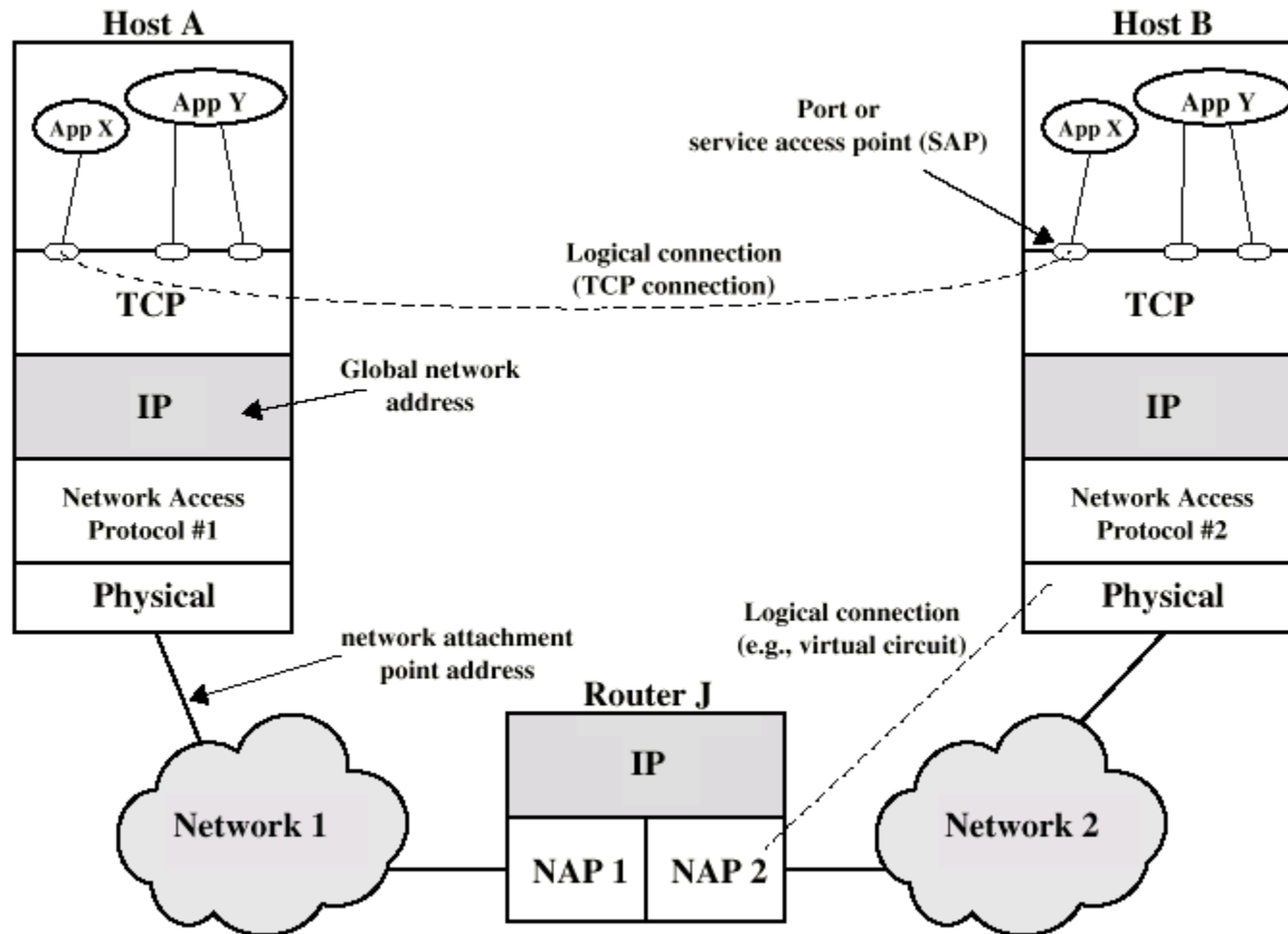
- Protecção contra perda ou corrupção de PDUs
- Implica detecção de erros e retransmissão de PDUs não aceites pelo receptor
- Pode ser necessário em várias camadas protocolares (nó a nó ou extremo a extremo)

# Endereçamento

---

- ◆ Numa rede é necessário identificar não só os sistemas que a constituem (*hosts, routers, etc.*) mas igualmente entidades protocolares, aplicações, etc., o que requer diferentes tipos e níveis de identificação ou endereçamento
  - » Endereços de sistemas
    - Normalmente trata-se de endereços lógicos, definidos na camada de Rede
    - Exemplos: endereços IP, endereços NSAP (*Network Service Access Point*)
  - » Endereços de interfaces a subredes (SNPA – *Subnetwork Point of Attachment*)
    - Endereços físicos que identificam pontos de acesso a uma subrede (interface física)
    - Exemplos: endereços MAC (LANs), endereços X.25
  - » Identificadores de processos / aplicações
    - Identificadores internos, normalmente concatenados com um endereço do sistema
    - Exemplos: porta TCP, TSAP (*Transport Service Access Point*)
  - » Identificadores de conexão
    - Em protocolos orientados à conexão, evitam a necessidade de usar endereços nos PDUs
    - Exemplos: identificadores de ligação de dados ou de circuito virtual
- ◆ Modos de endereçamento
  - » *Unicast, multicast, broadcast*

## *Exemplo de Endereços*



## *Multiplexagem*

---

- ◆ Um protocolo orientado à conexão pode suportar múltiplas conexões simultâneas
- ◆ Um serviço de uma camada (CO ou CL) suporta-se num serviço da camada adjacente inferior (CO ou CL), sendo possíveis as quatro combinações
  - » CO / CO – exemplo: Circuito Virtual X.25 sobre Ligação de Dados LAPB
  - » CO / CL – exemplo: TCP sobre IP
  - » CL / CL – exemplos: IP sobre serviço LAN (MAC); UDP sobre IP
  - » CL / CO – exemplo: IP sobre ATM; IP sobre *Frame Relay*
- ◆ No caso de um serviço CO construído sobre um serviço CO é possível a multiplexagem de conexões entre níveis
  - » Um para um
  - » Multiplexagem ascendente – múltiplas conexões numa camada partilham uma conexão na camada inferior (e.g., múltiplos circuitos virtuais X.25 sobre uma ligação LAPB)
  - » Multiplexagem descendente (*Inverse Multiplexing* ou *splitting*) – uma conexão numa camada é construída sobre múltiplas conexões na camada inferior

# *Arquitectura Protocolar TCP/IP*

---

## ◆ Arquitectura dominante

- » Desenvolvida inicialmente no âmbito da ARPANET, que começou por ser uma rede experimental financiada pelo Departamento de Defesa dos EUA, e que ligava universidades e centros de investigação
- » Os protocolos da família TCP/IP foram especificados e implementados antes da maior parte dos protocolos baseados no modelo OSI
- » Um grande número de serviços e aplicações disponíveis actualmente usa TCP/IP

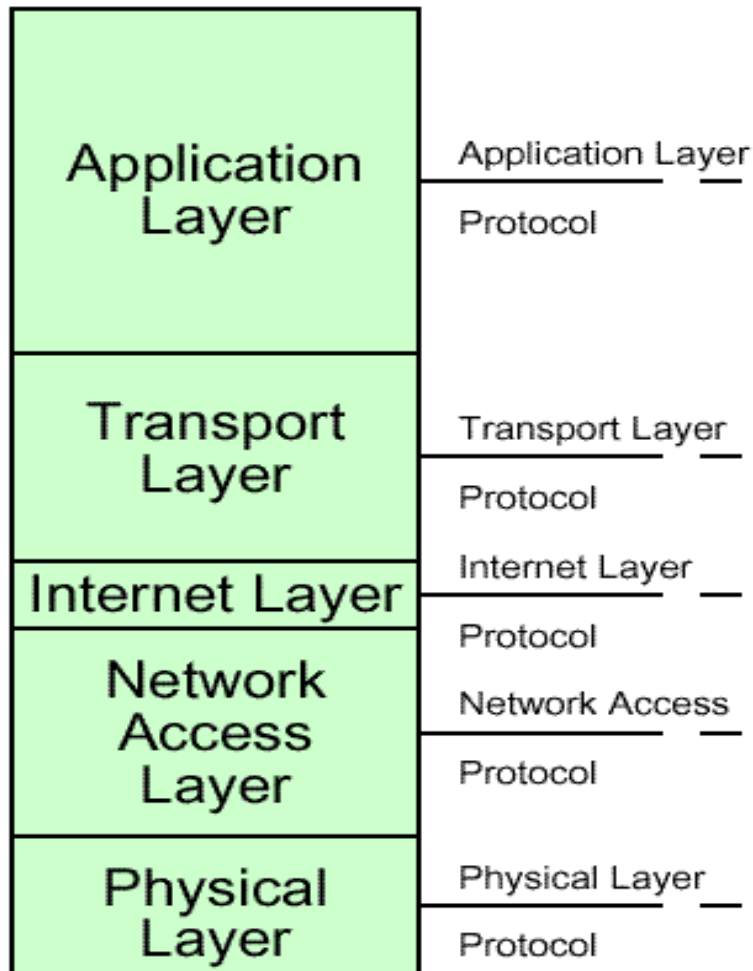
## ◆ Princípios

- » As funções de comunicação são estruturadas em módulos
- » Entidades comunicam com entidades homólogas (*peer*) noutros sistemas
- » Num sistema uma entidade
  - Usa serviços de outras entidades
  - Fornece serviços a outras entidades
  - Serviços podem ser fornecidos a camadas não adjacentes (ao contrário do modelo OSI)



# Arquitectura Protocolar TCP/IP

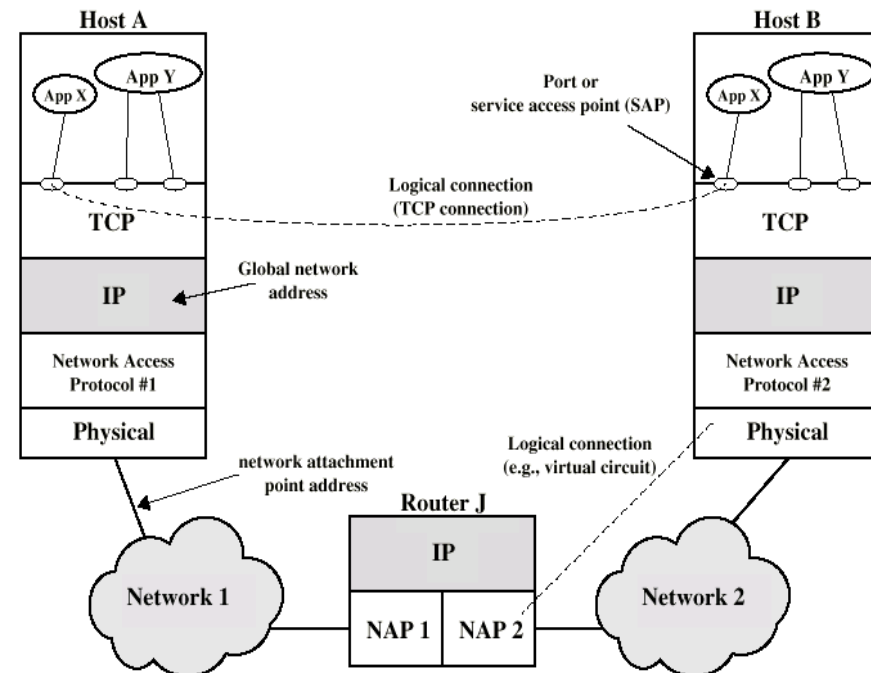
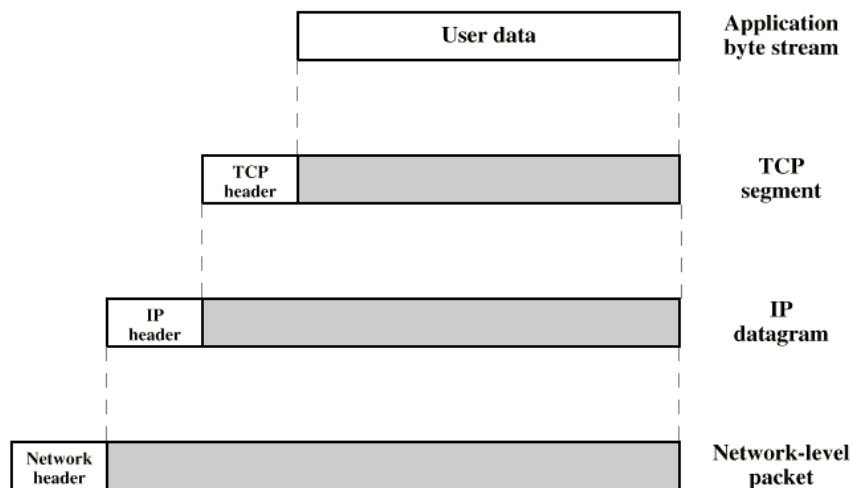
---



- » Aplicação – serviços de utilizador
  - Comunicação entre processos ou aplicações
  - Modelo cliente-servidor
  - HTTP, FTP, telnet
- » Transporte (TCP/UDP)
  - Transmissão de mensagens extremo a extremo
  - Independente do serviço de sub(redes) físicas
  - Transferência fiável (TCP) ou não fiável (UDP)
- » Internet (IP)
  - Encaminhamento através de múltiplas (sub)redes interligadas (*internetworking*)
  - Implementado em computadores (*hosts*) e nós intermédios (*routers*)
- » Acesso a uma rede (subrede)
  - Acesso a uma (sub)rede e comunicação entre estações (*hosts / routers*) ligadas à mesma (sub)rede física
- » Interface física
  - Características eléctricas e mecânicas do acesso à (sub)rede (níveis de sinal, débitos de transmissão, conectores, etc.)

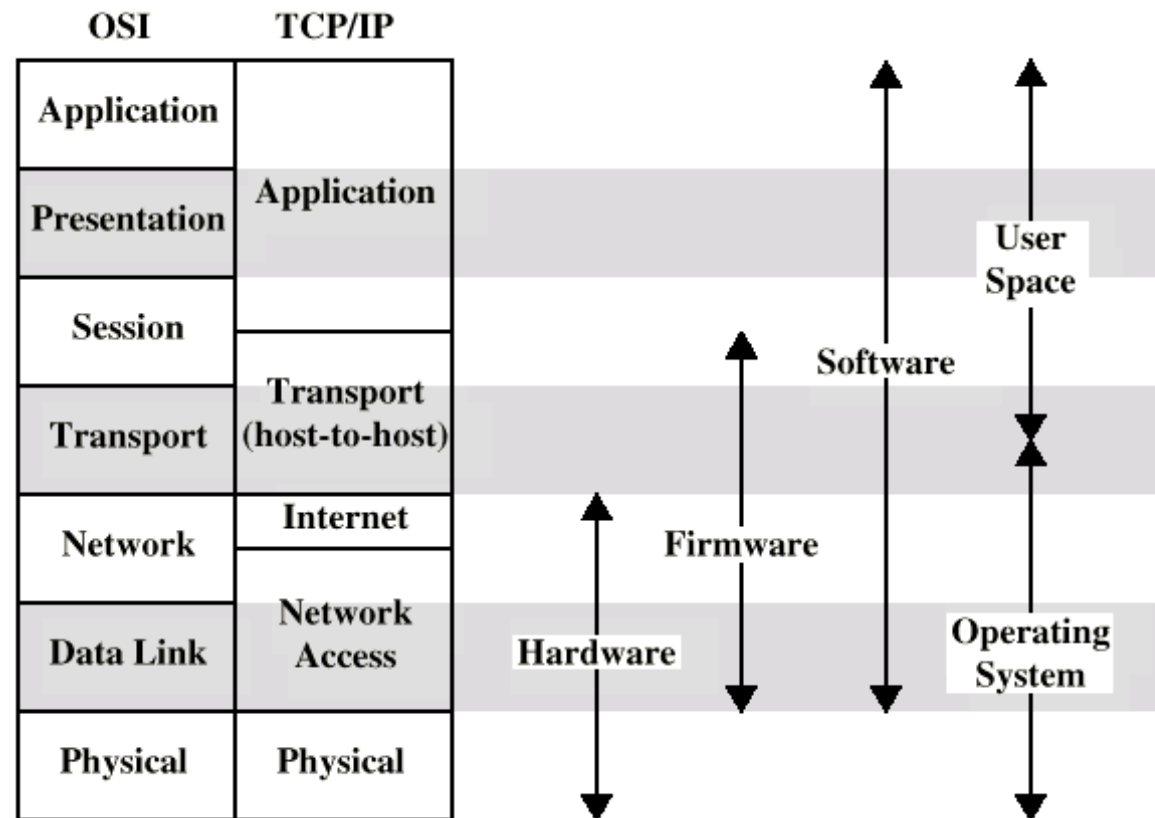
## Algumas Características do TCP/IP

- ◆ O IP (*Internet Protocol*) é implementado em todos os computadores (*hosts*) e *routers*
- ◆ Cada computador tem um endereço IP único em cada subrede a que pertence
- ◆ Cada processo num computador tem um endereço único (porta)



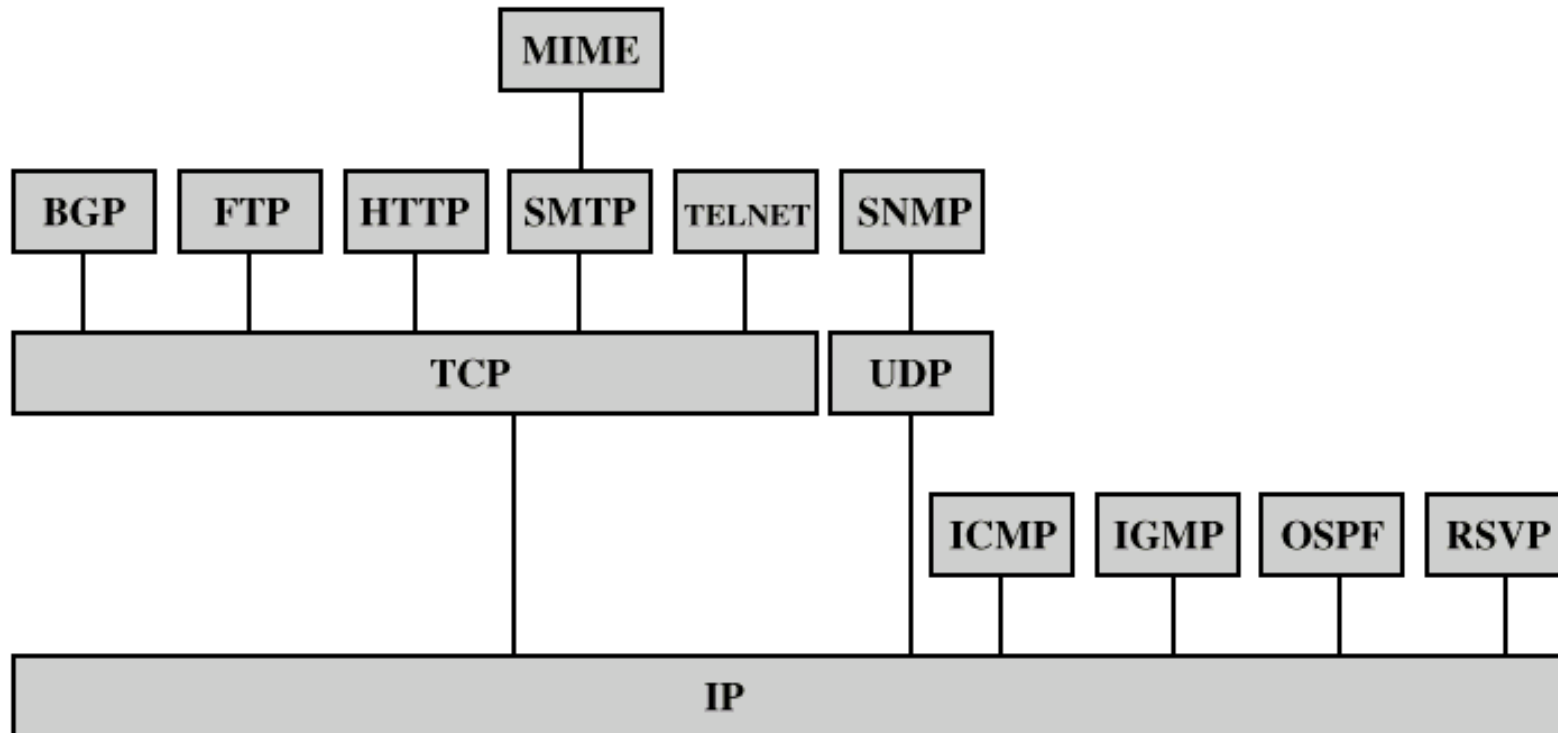
# *OSI vs TCP/IP*

---



# *Família de Protocolos TCP/IP*

---



**BGP** = Border Gateway Protocol  
**FTP** = File Transfer Protocol  
**HTTP** = Hypertext Transfer Protocol  
**ICMP** = Internet Control Message Protocol  
**IGMP** = Internet Group Management Protocol  
**IP** = Internet Protocol  
**MIME** = Multi-Purpose Internet Mail Extension

**OSPF** = Open Shortest Path First  
**RSVP** = Resource ReSerVation Protocol  
**SMTP** = Simple Mail Transfer Protocol  
**SNMP** = Simple Network Management Protocol  
**TCP** = Transmission Control Protocol  
**UDP** = User Datagram Protocol