Institute of Telecommunications
Warsaw University of Technology
2019

internet technologies and standards

- Piotr Gajowniczek
- Andrzej Bąk
- Michał Jarociński

introduction

Internet – assumptions & definition

- *1974 – assumptions (Cerf & Kahn)*
  - minimalism & autonomy – simple common protocol allows interconnecting separate networks without interfering with their internal structure
  - stateless routers
  - *best effort* service model
  - no centralized management
- *1995 - definition*

RESOLUTION: The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term "Internet".

"Internet" refers to the global information system that --

(i) is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/follow-ons;

(ii) is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and

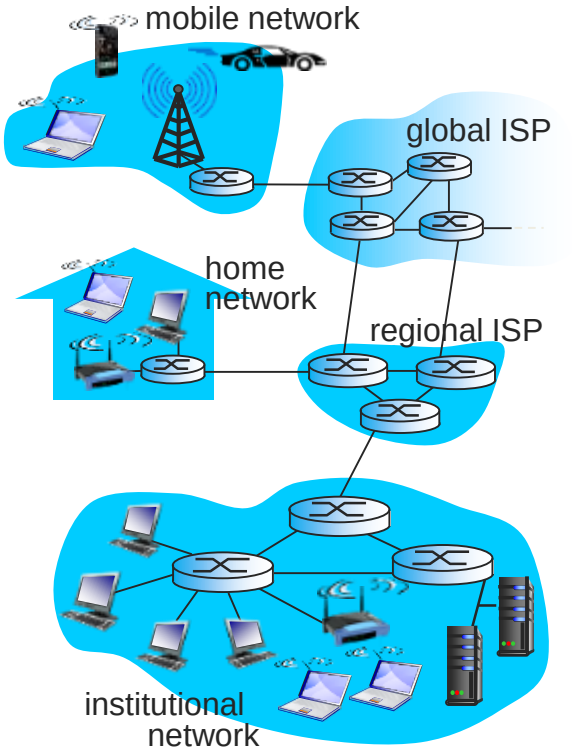
(iii) provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein.

what is the Internet? infrastructure view

- *Internet: "network of networks"*
  - interconnected ISPs
- *protocols control sending, receiving of msgs*
  - data plane
  - control plane
  - management plane

protocols define format and order of messages sent and received among network entities, and actions taken on message transmission and receipt

- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



- network (switches, routers) – access, aggregation, transport
- users (hosts, servers) – applications

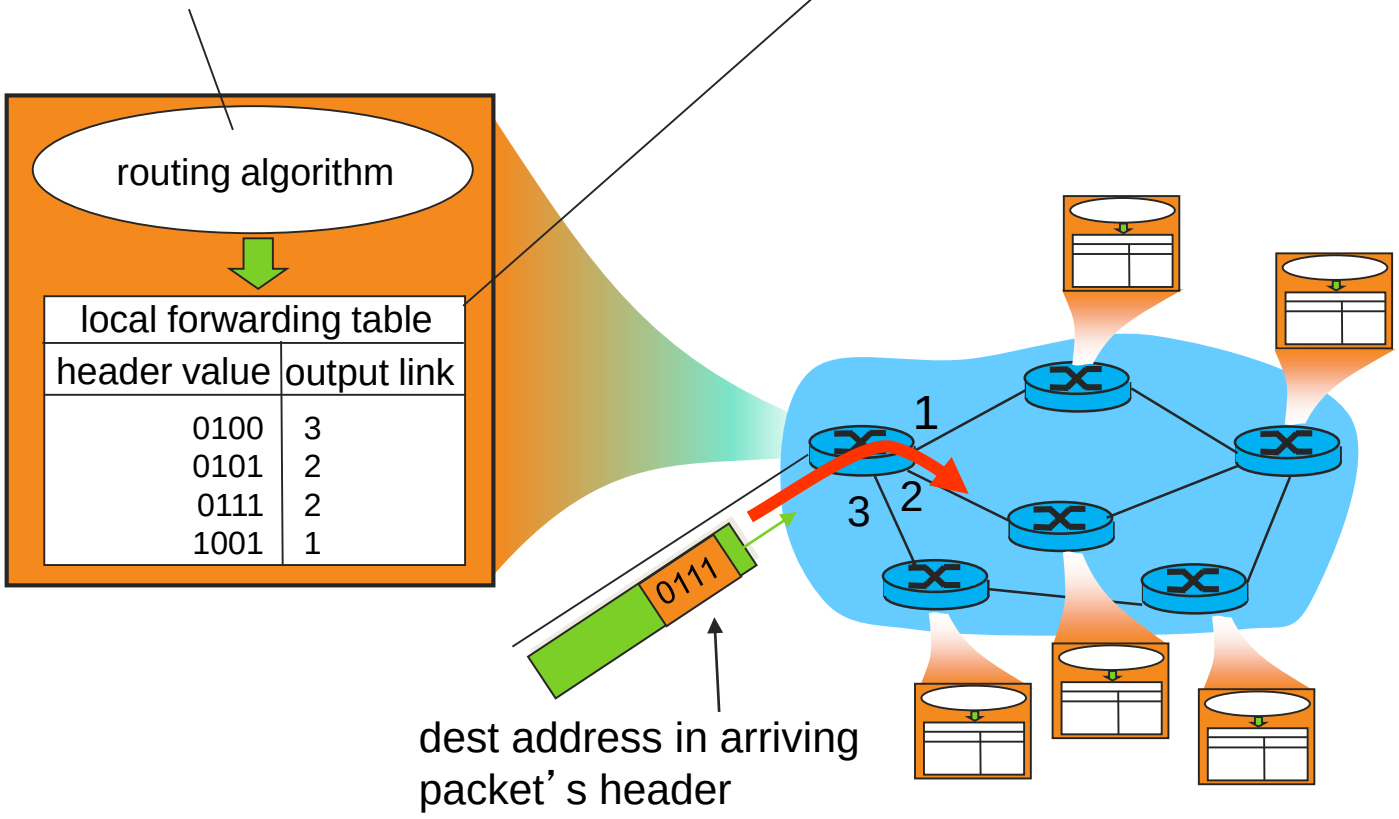
the network core

- *mesh of interconnected routers*

routing: determines source-destination route taken by packets

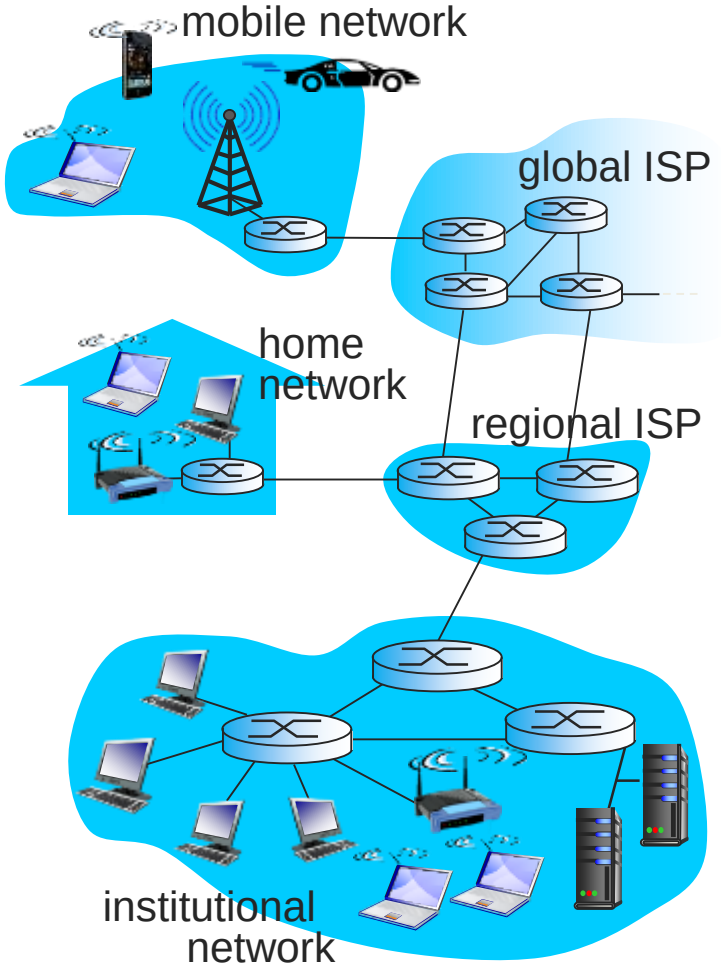
- routing algorithms

forwarding: move packets from router's input to appropriate router output



what is the Internet: a service view

- *infrastructure that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, …
- *provides programming interface to apps*
  - hooks that allow sending and receiving app programs to "connect" to Internet



Internet applications

- Main categories
  - messaging
  - data retrieval
  - real-time / continuous media
- Main architectures
  - client/server
  - peer-to-peer (P2P)
- Main interaction patterns
  - request – response
  - continuous media
  - event-based

structure of the Internet

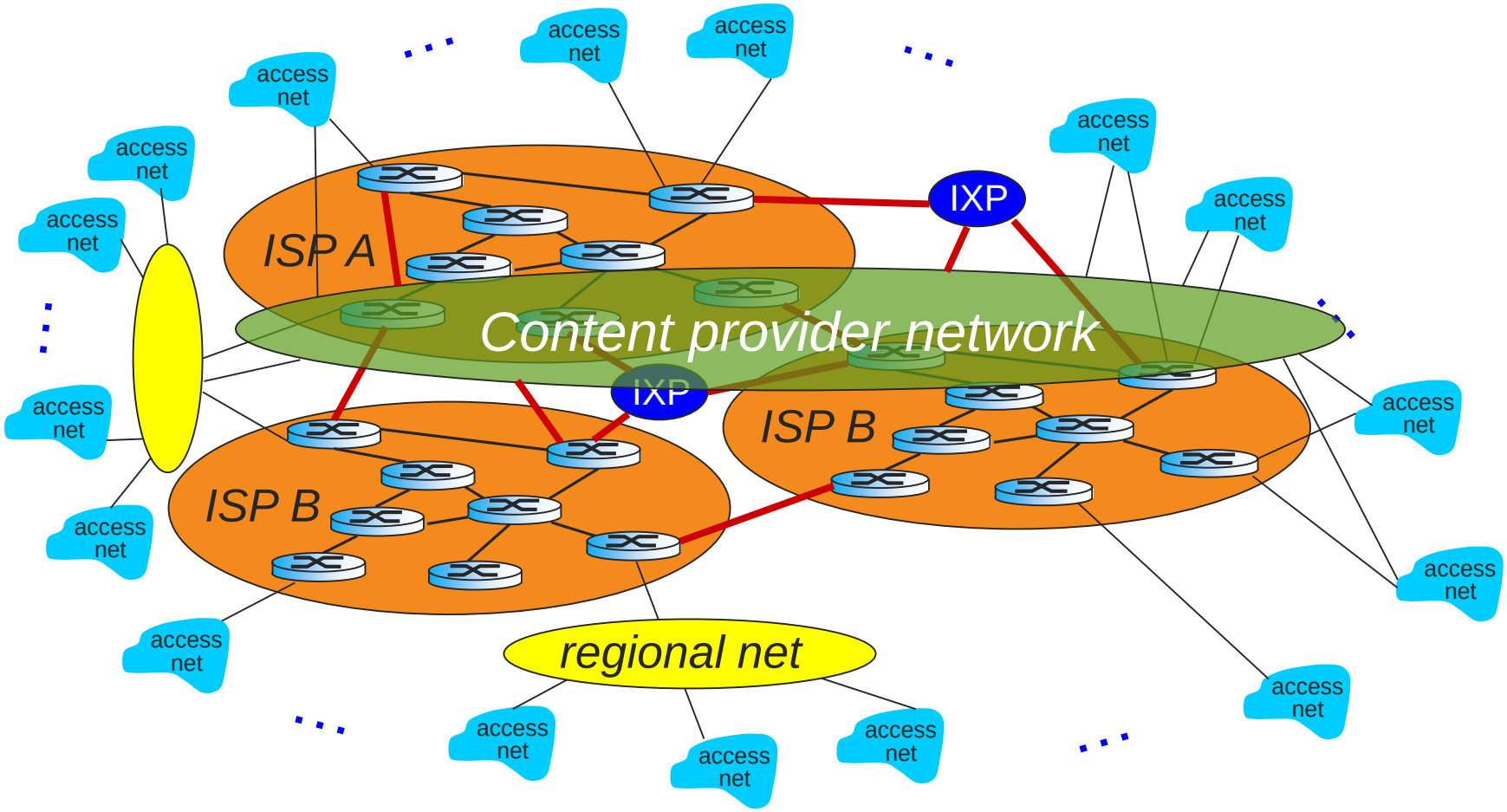
Internet structure

- *terminals / end systems connect to Internet via access ISPs (Internet Service Providers)*
  - ISPs have various scales of operation
- *access ISPs in turn have to be interconnected*
  - with each other
  - with service / content providers (data centers)
  - to facilitate global data exchange

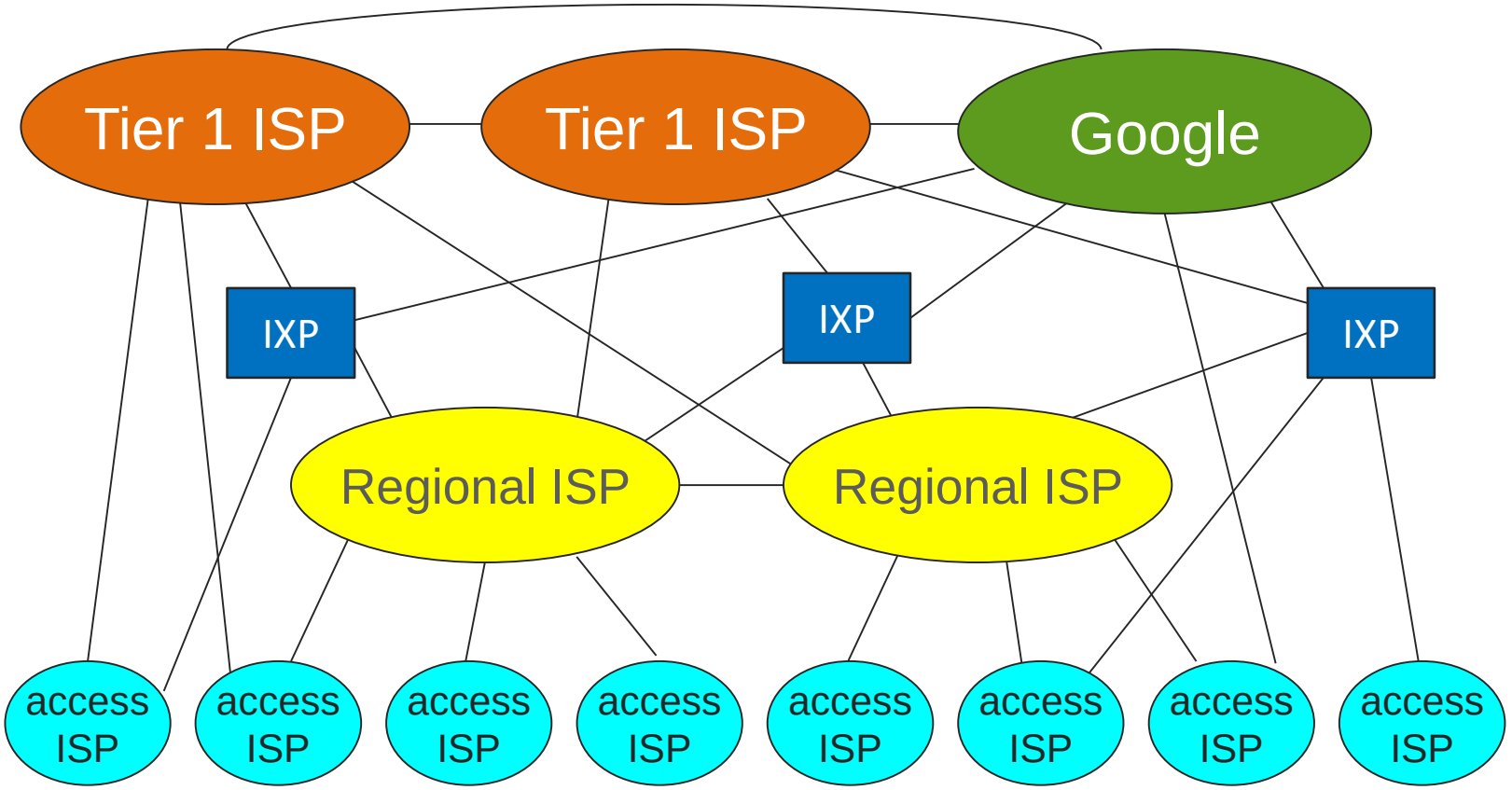
question: given millions of access ISPs, how to connect them together?

Internet structure: network of networks

- *ISPs, content providers (e.g. Google, Netflix, Facebook…), CDNs (e.g. Akamai …)*
- *public (IXP) or private peering*

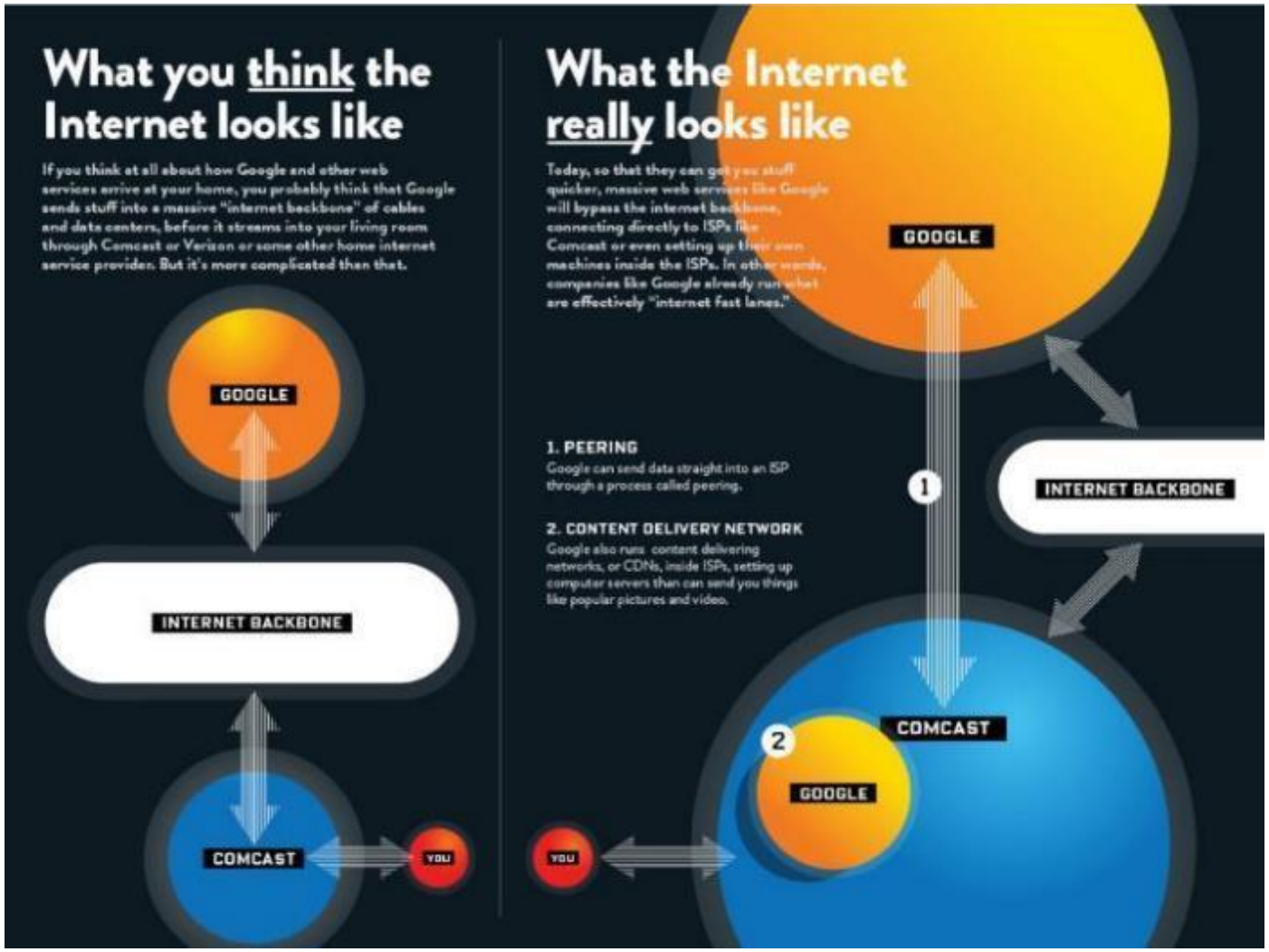


Internet structure: network of networks



- *at center: small # of well-connected large networks*
  - "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - content provider network (e.g, Google): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

content delivery and "net neutrality"



source: What Everyone Gets Wrong in the Debate Over Net Neutrality | WIRED

layered networking architecture

how to describe all this complexity?

networks are complex, with many "pieces":

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

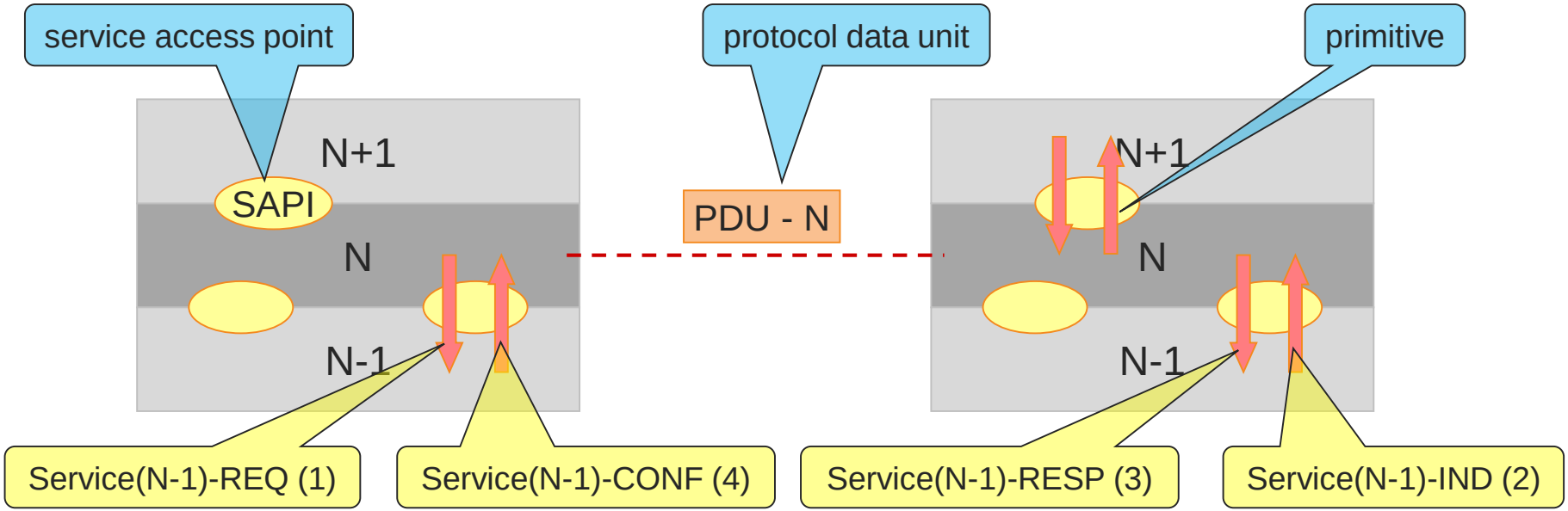
question:

is there any hope of organizing structure of network?

…. or at least our discussion of networks?

the concept of layering (aka protocol stack)

- *functions are split into hierarchical groups called layers*
- *solution often used in engineering or service systems*
  - a layer offers a set of services
  - services of upper layer are created on the base of services offered by a lower layer (protocol stack)
  - layer protocol – rules of cooperation between same layer entities in remote systems
  - protocol standardization – PDUs and data exchange scenarios



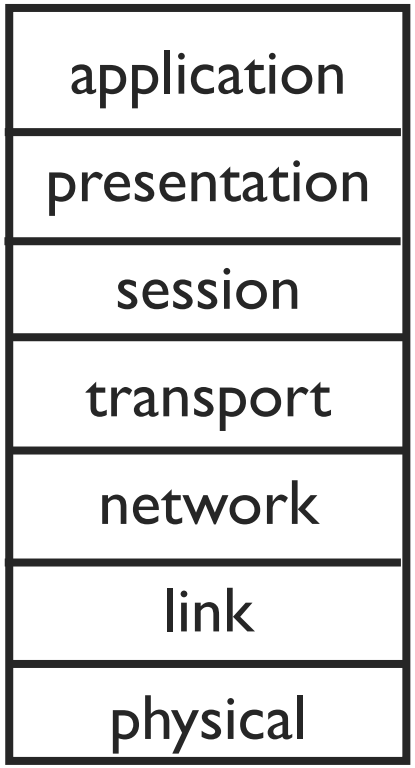
why a stack?

- layer = functionality + data structure(s)
- benefits
  - harnessing complexity by splitting a task into subproblems
  - allows implementing changes inside a layer without influencing others and the way how the whole system behaves
    - limiting data exchange between protocols
    - « black box »
  - allows to define "stable" interfaces
    - interface defines a layer "vertically"
- how many layers?
  - OSI model (reference) – 7 layers
  - TCP/IP (Internet) stack – 4 layers
  - "programmer" view – 2 layers

ISO/OSI reference model

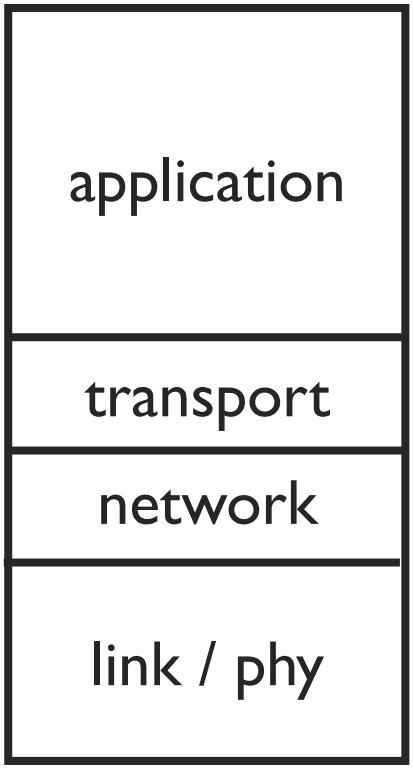
- *presentation: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions*
- *session: synchronization, checkpointing, recovery of data exchange*
- *Internet stack has these layers "incorporated" into application layer*

- traditionally:
  - L2 (link)
  - L3 (network = IP)
  - L4 (transport = TCP)
  - L7 (applications)
  - L2.5 = ?

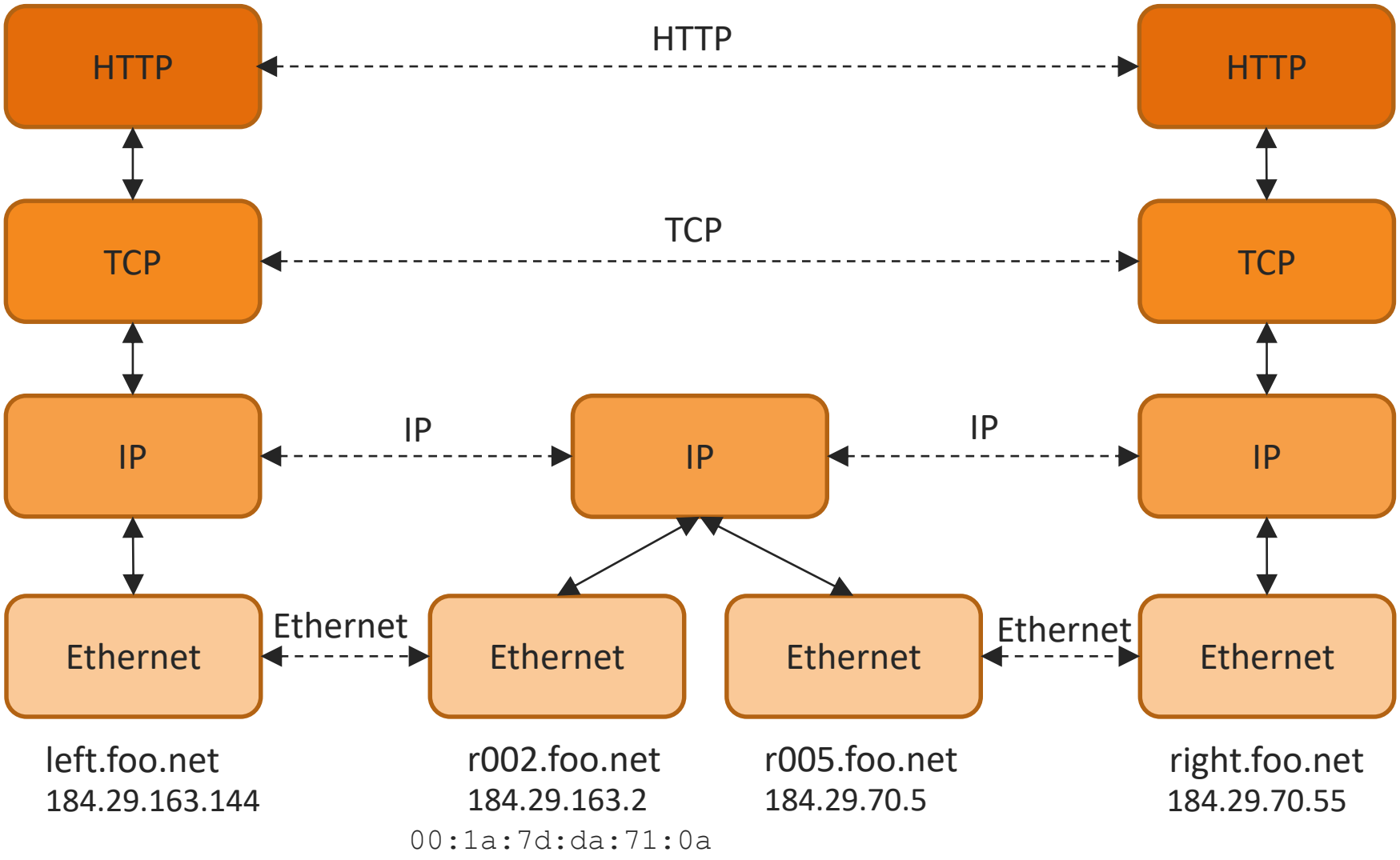


Internet protocol stack

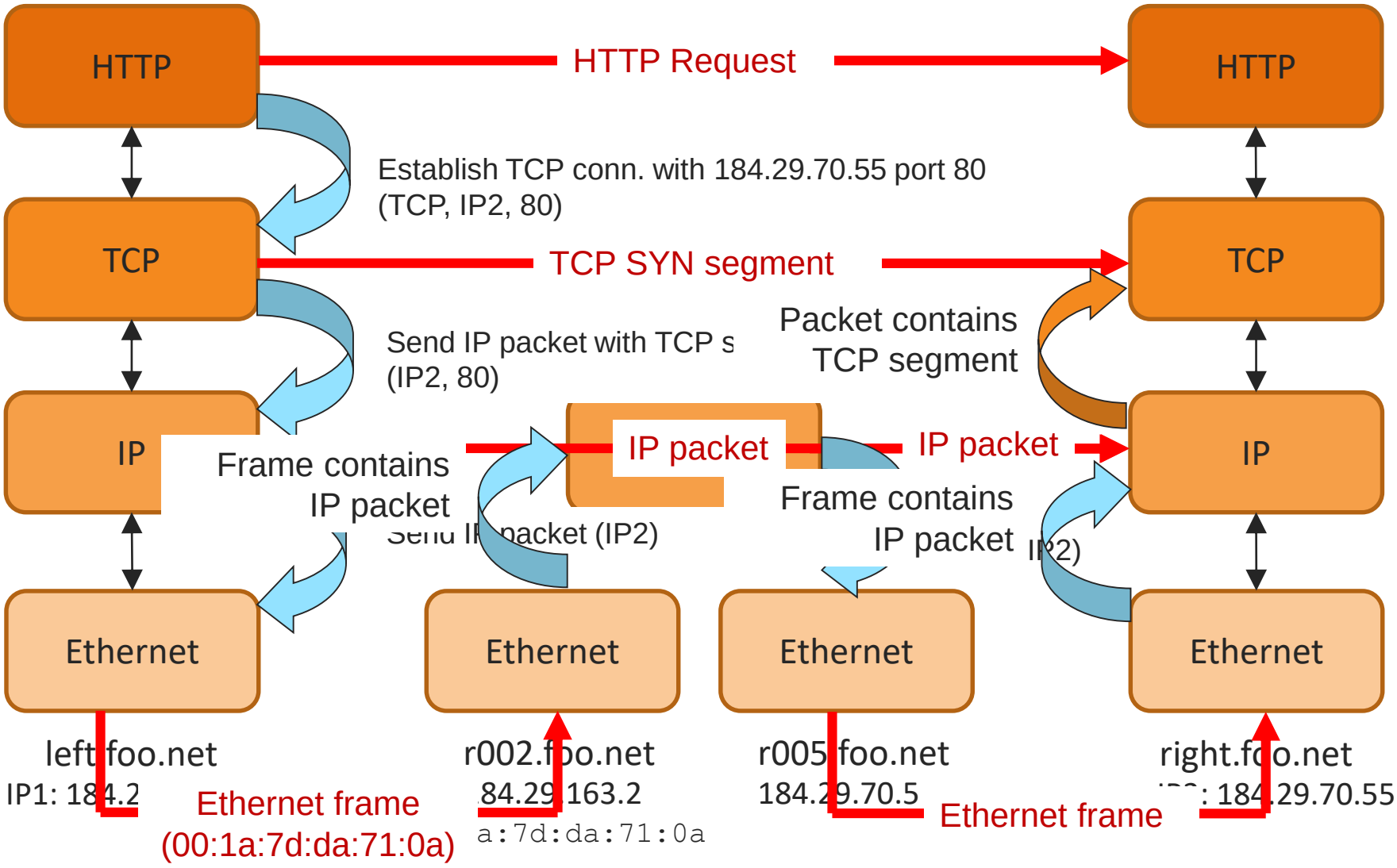
- *application: supporting network applications*
  - FTP, SMTP, HTTP
- *transport: process-process data transfer*
  - TCP, UDP
- *network: routing of datagrams from source to destination*
  - IP, routing protocols
- *link / phy: data transfer between neighboring network elements*
  - Ethernet, 802.111 (WiFi), PPP
  - bits "on the wire"



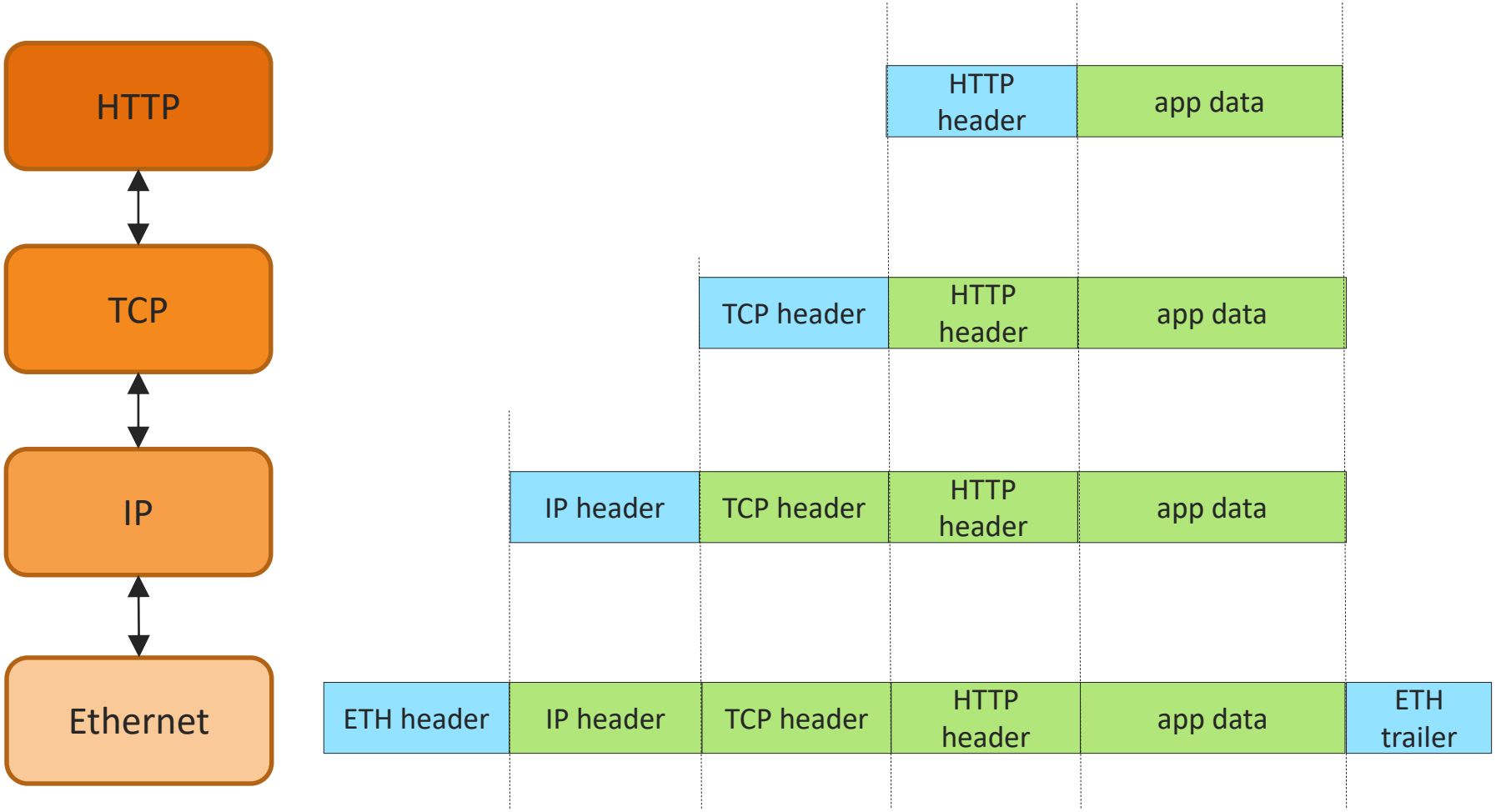
example


HTTP
HTTP
HTTP
TCP
TCP
TCP
IP
IP
IP
IP
Ethernet
Ethernet
Ethernet
Ethernet
Ethernet
left.foo.net
184.29.163.144
r002.foo.net
184.29.163.2
00:1a:7d:da:71:0a
r005.foo.net
184.29.70.5
right.foo.net
184.29.70.55

example


HTTP
HTTP Request
HTTP
Establish TCP conn. with 184.29.70.55 port 80
(TCP, IP2, 80)
TCP
TCP SYN segment
TCP
Packet contains
TCP segment
Send IP packet with TCP s
(IP2, 80)
IP
IP packet
IP packet
IP
Frame contains
IP packet
Frame contains
IP packet
Ethernet
Ethernet
Ethernet
Ethernet
left.foo.net
IP1: 184.2
r002.foo.net
84.29.163.2
a:7d:da:71:0a
r005.foo.net
184.29.70.5
right.foo.net
Ethernet frame
(00:1a:7d:da:71:0a)
Ethernet frame

encapsulation



- *each protocol adds control information, creating new PDU*