Course on Computer Communication and Networks

Lecture 12
Continuously evolving Internet-working
Part 2: QoS, traffic engineering, SDN, IoT

EDA344/DIT 420, CTH/GU

Based on the book Computer Networking: A Top Down Approach, Jim Kurose, Keith Ross, Addison-Wesley.

Recall: Internet protocol stack layers

Application: protocols supporting *n/W applications*

http (*web*), smtp (*email*),

p2p, media streaming, CPS/IoT apps

Transport: end2end data transfer protocols

UDP, TCP

Network: routing of datagrams, connecting different physical networks

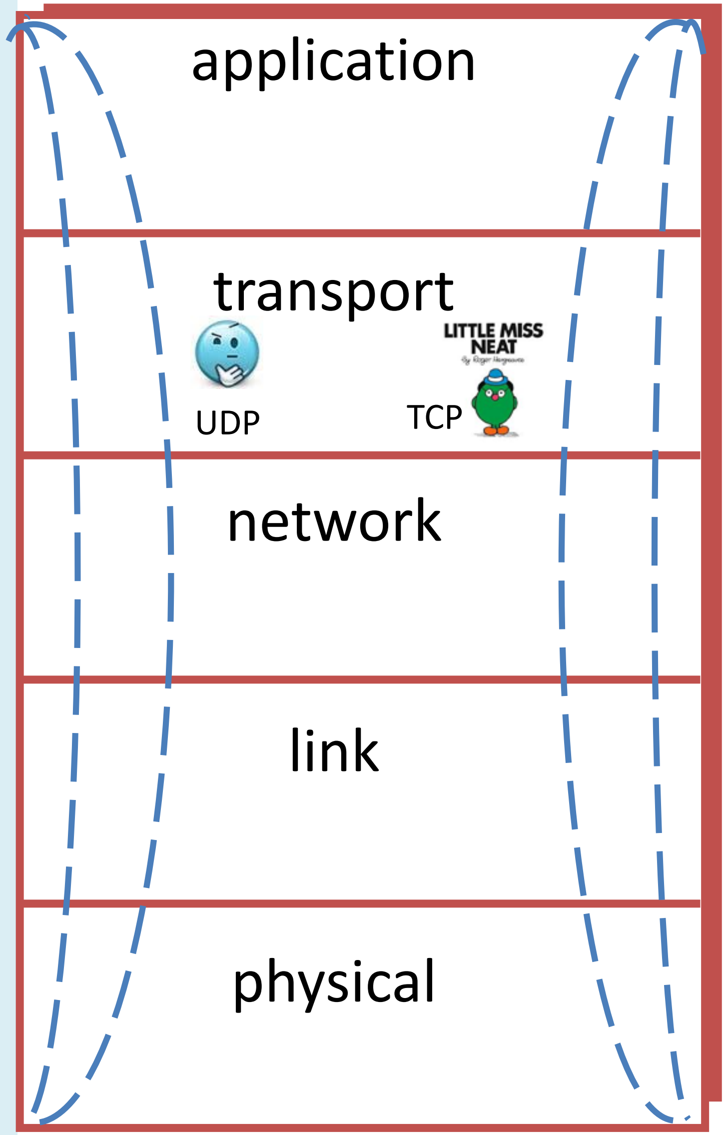
IP addressing, routing,

Virtualization, traffic engineering, Software Defined Networks

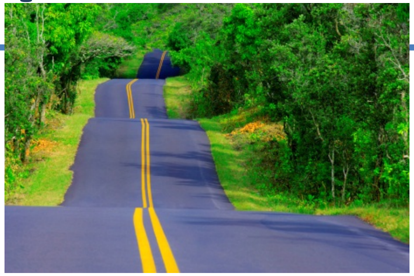
link: data transfer between neighboring nodes

Ethernet, WiFi, …

physical: protocols for bit-transmission/receipt on the physical medium between neighboring nodes



Let's hit the road again☺: Roadmap



NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*

Timing/bandwidth guarantees in networks

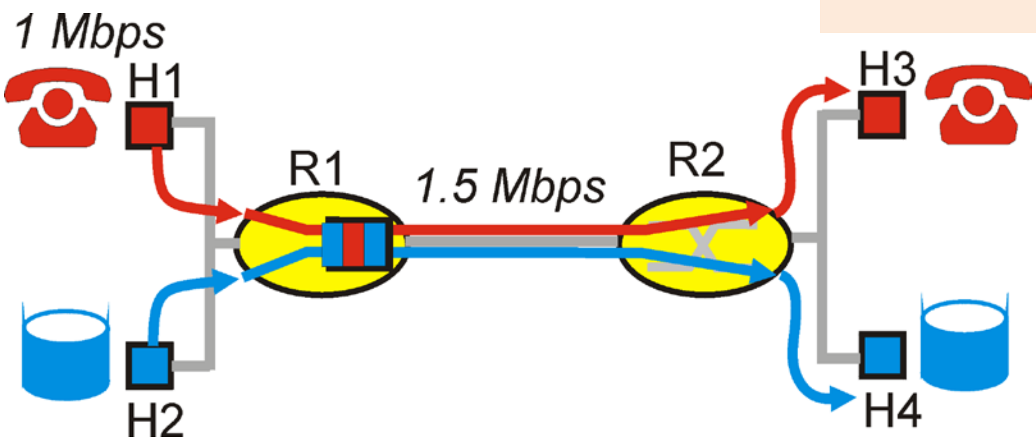
aka Quality of Service (QoS): agreement on

- Traffic characteristics (packet rate, sizes, …)
- Network service guarantees (delay, jitter, loss rate, …)

model for resource sharing and congestion studies:
questions/principles for QoS in Network Core

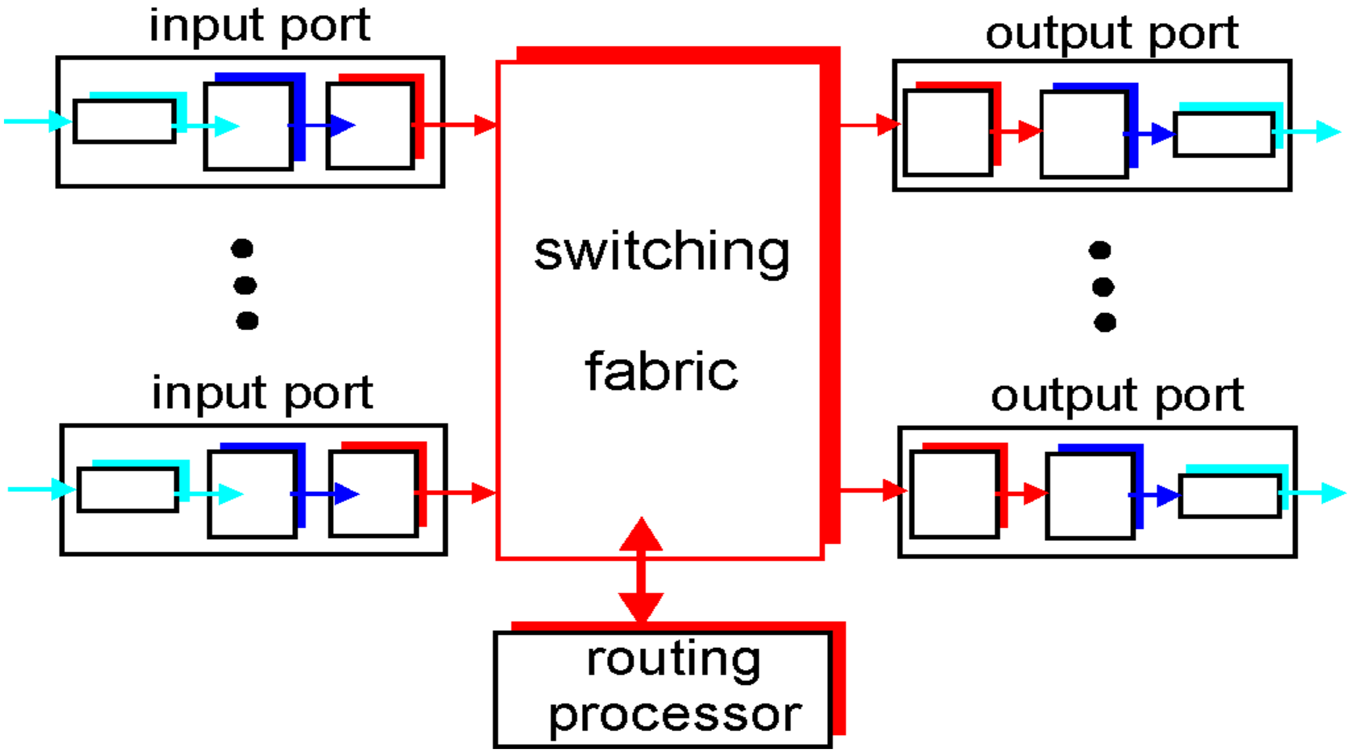
- Distinguish traffic?
- Control offered load? (isolate different ”streams”?)
- Resources? (utilization)
- Control acceptance of new sessions?

- Packet classification & scheduling (bandwidth allocation)
- Traffic shaping/policing (enforce contract terms)
- Admission control



Where does this go in?

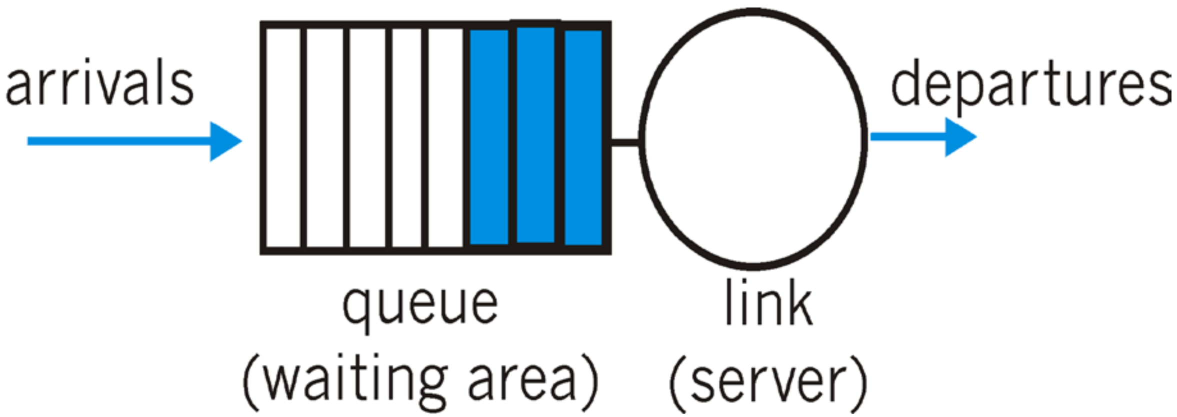
Scheduling = choosing the next packet for transmission on a link (= allocate bandwidth)



Packet Scheduling: FIFO

FIFO: in order of arrival to the queue

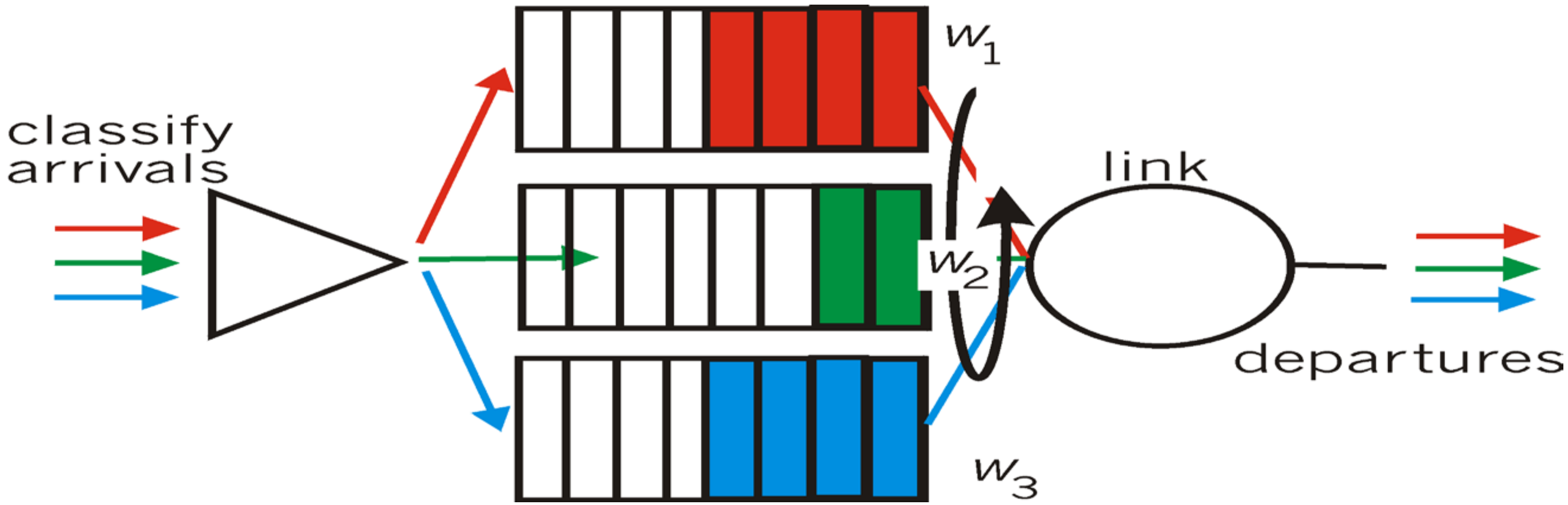
- ❐ if buffer full: a discard policy determines which packet to discard among the arrival and those already queued



Packet Scheduling: Weighted Fair Queueing

Weighted Fair Queuing: generalized Round Robin, including priorities (weights)

- provide each class with a differentiated amount of service
- class i receives a fraction of service $w_i/\sum(w_j)$



- There are a lot more decision options about packet scheduling: work-conserving policies, delays, …

Policing Mechanisms

Idea: *shape* the packet traffic :network provider does *traffic policing,* ie enforces the ”shape” agreed.

- Traffic shaping, to limit transmission rates:
  - (Long term) **Average Rate** (e.g.100 pkts/sec or 6000 packets per min)
  - **Peak Rate**: e.g.1500 pkts/sec peak
  - (Max.) **Burst Size**: Max. number of packets sent consecutively, ie over a very short period of time

Policing Mechanisms: Pure *Leaky Bucket*

Idea: eliminates bursts completely; may cause unnecessary packet losses

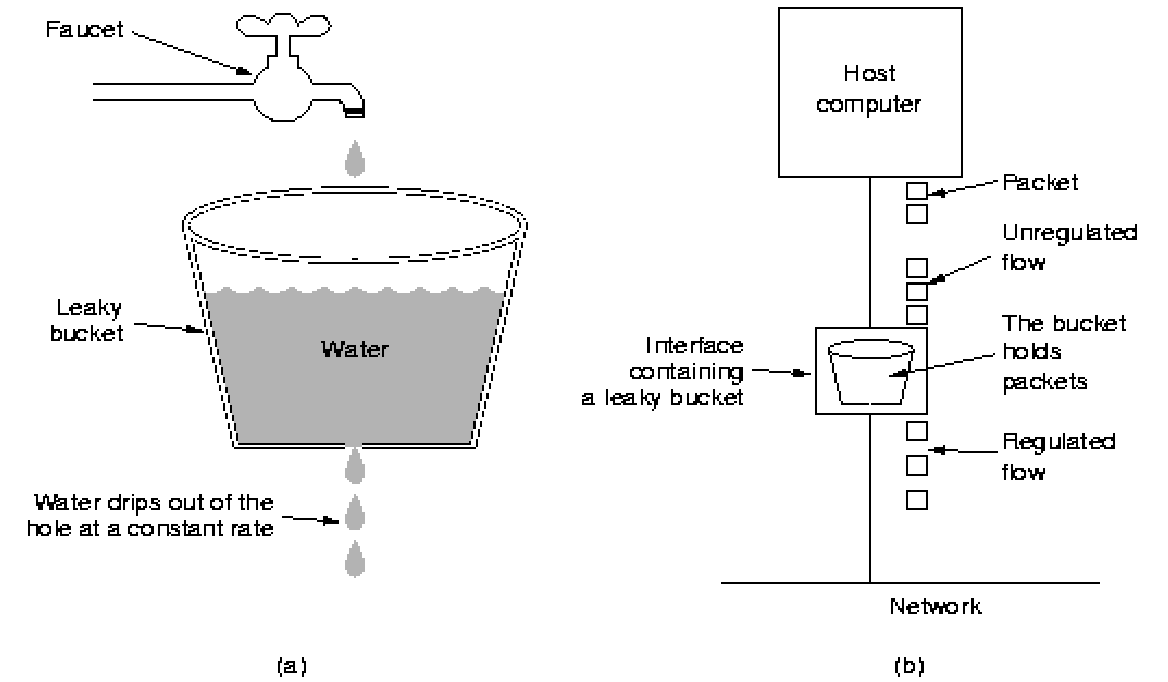


Fig. 5-24. (a) A leaky bucket with water. (b) A leaky bucket with packets.

Policing Mechanisms:Leaky*Token Bucket*

Idea: packets sent by consuming tokens produced at constant rate r

- limit input to specified Burst Size (b= bucket capacity) and Average Rate (max admitted #packets over time period t is b+rt).
- to avoid still much burstiness, put a leaky bucket -with higher rate; after the token bucket)

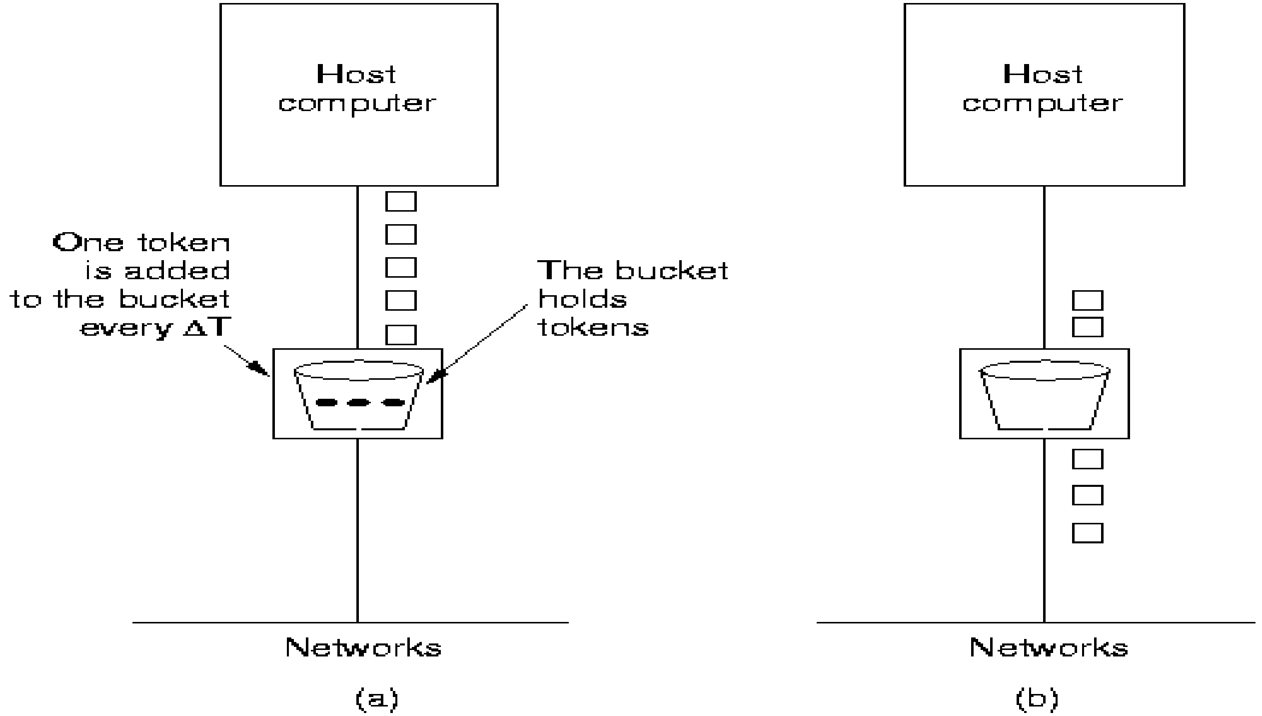
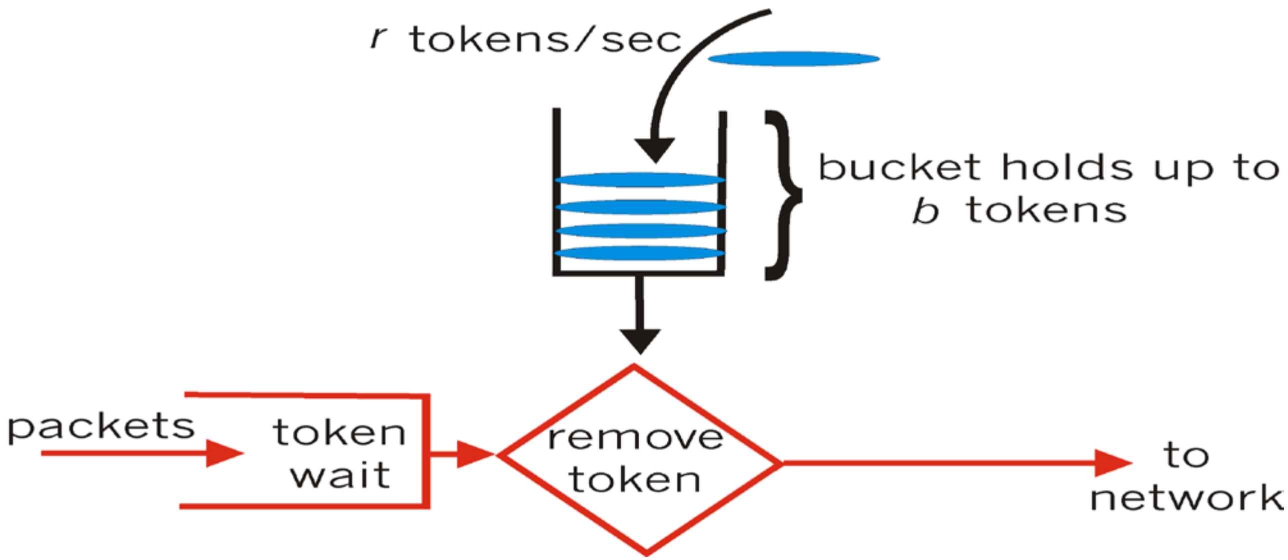


Fig. 5-26. The token bucket algorithm. (a) Before. (b) After.

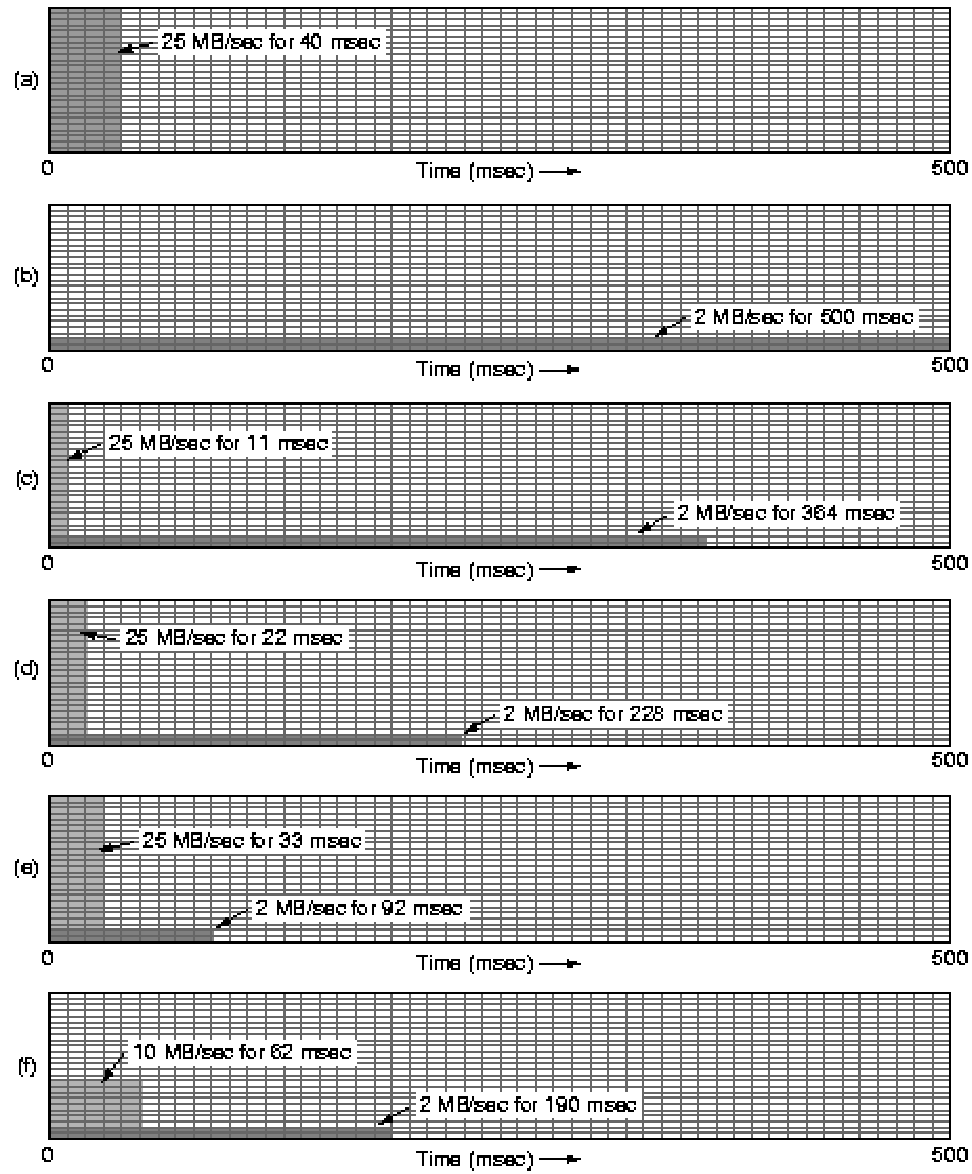
Policing Mechanisms: token bucket

Another way to illustrate token buckets:

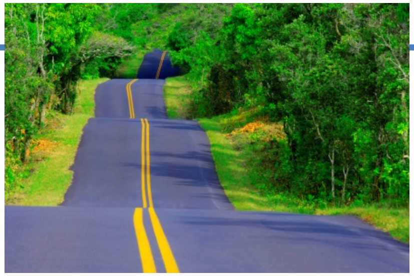


Policing: the effect of buckets

- input
- output pure leaky bucket, 2MBps
- output token bucket 250KB, 2MBps
- output token bucket 500KB, 2MBps
- output token bucket 750KB, 2MBps
- output token bucket 500KB, 2MBps, feeding 10MBps leaky bucket



Roadmap



NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*

Virtual Circuit example:
ATM: Asynchronous Transfer Mode nets

Internet 's IP:

- today's *de facto* standard for global data networking

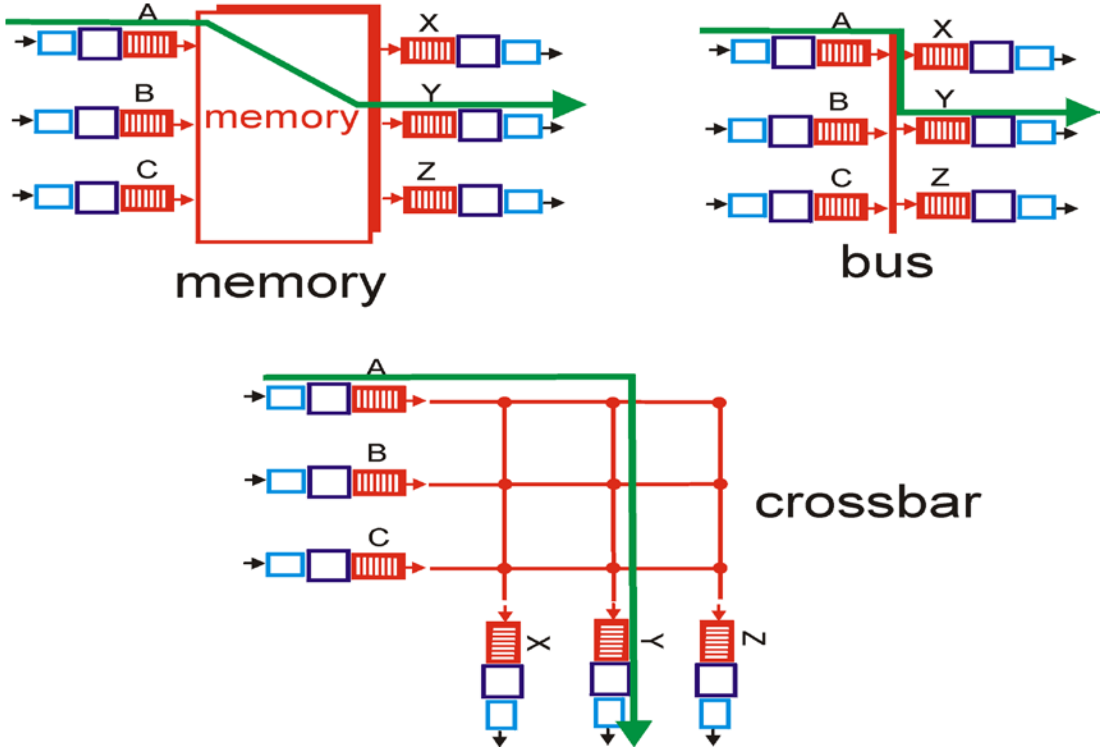
1980's:

- telco's develop ATM specifications: competing network standard for carrying high-speed voice/data

ATM principles:

- virtual-circuit networks: switches maintain state for each "call"
- small (48 byte payload, 5 byte header) fixed length *cells* (like packets)
  - fast switching
  - small size good for voice
- well-defined interface between "network" and "user" (think of classic telecom)

Recall: switching fabrics



- ATM switches: VC technology
  - support for virtual channels, virtual circuits (based on Banyan crossbar switches)

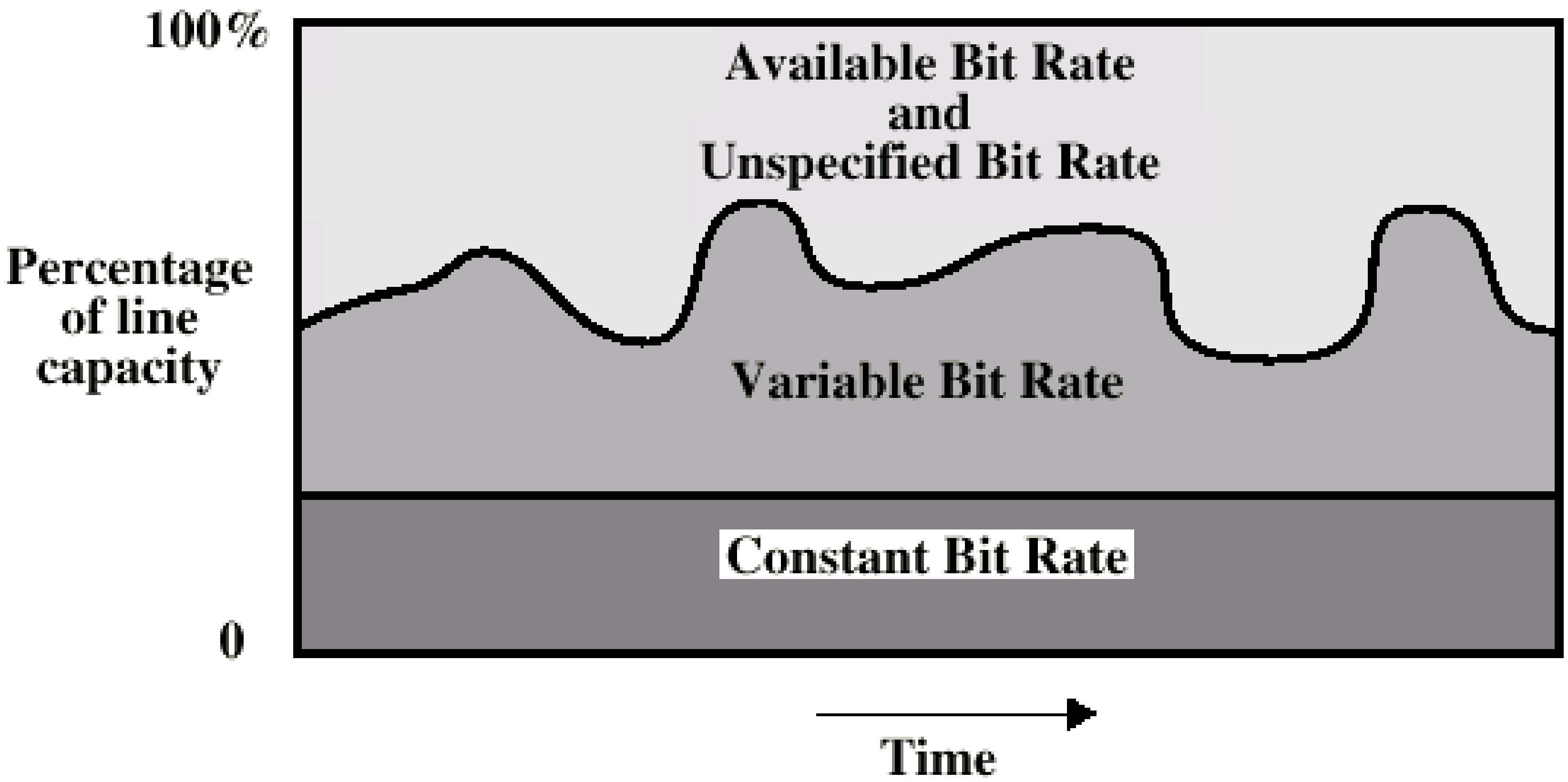
- ATM routing: as train travelling for routers (hence no state for each ”stream/passenger”, but for each ”train” = virtual path)

Example VC technology
ATM Network service models:

Service Model	Example	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion control
Constant Bit Rate	voice	constant rate	yes	yes	yes	Admission control
VariableBR (RT/nRT)	Video/ "streaming"	guaranteed rate	yes	yes	yes	Admission control
Available BR	www-browsing	guaranteed minimum	no	yes	no	Yes, feedback
Undefined BR	Background file transfer	none	no	yes	no	discard pkts

With ABR you can get min guaranteed capacity and better, if possible; with UBR you can get better, but you may be thrown out in the middle ☹

ATM Bit Rate Services



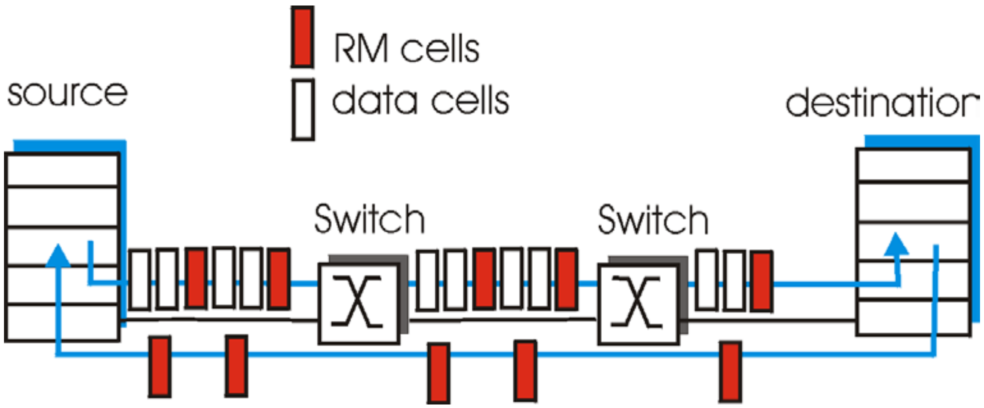
ATM (VC) Congestion Control
Several different strategies:

Admission control and resource reservation (CBR and VBR traffic*:*

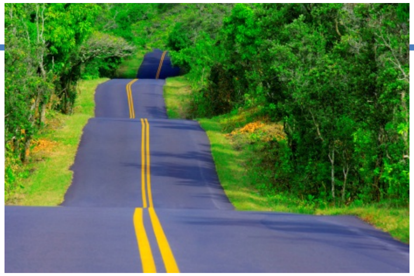
reserve resources when opening a VC; traffic shaping and policing *(use bucket-like methods)*

Rate-based congestion control*:*
(ABR traffic)

- **idea** = feedback to the sender and intermediate stations on the *available (= max. acceptable) rate* on the VC.
- similar to "choke packets" (option provided in ICMP, which is not used in implementations…)



Roadmap



NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*

Internet bandwidth guarantee support possibilities?

Diffserv approach:

- provide functional components to build service classes
  - Network core: stateless, simple
  - Combine flows into aggregated flows
  - Classification, shaping, admission at the network edge

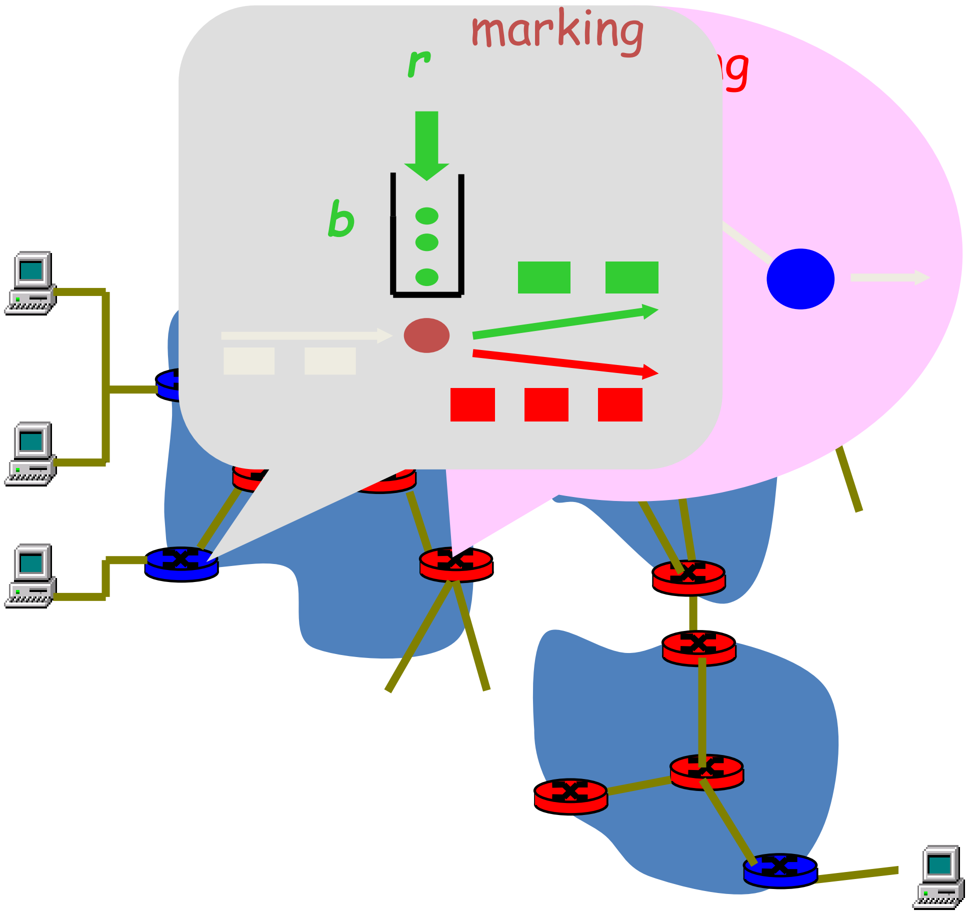
Diffserv Architecture

Edge router:

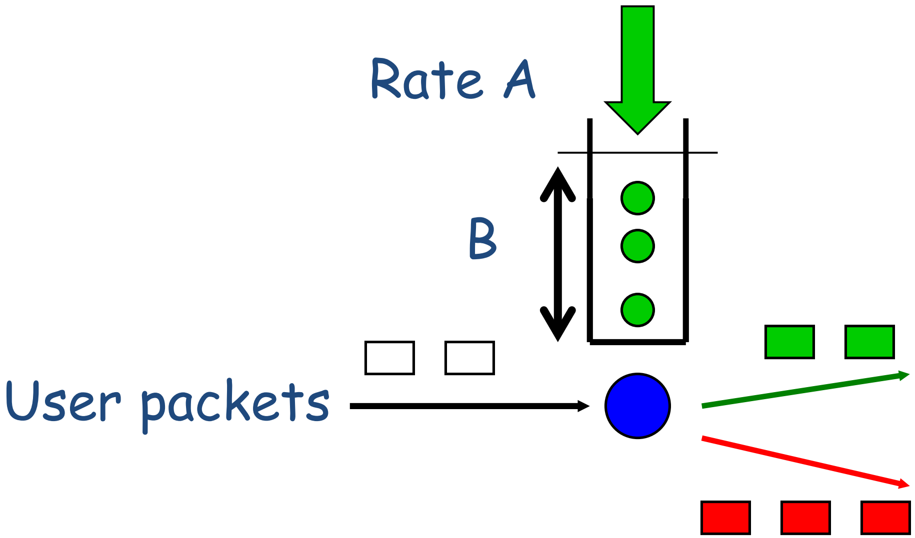
- per-flow traffic management
- marks packets as in-profile and out-profile

Core router:

- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets



Edge-router Packet Marking



-Class-based marking: packets of different classes marked differently

Profile within class: pre-negotiated rate A, bucket size B

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6

DiffServ Core Functions

- **Forwarding**: according to "Per-Hop-Behavior" (PHB) specified for the particular packet class; PHB is strictly based on classification marking
  - PHB does not specify what mechanisms to use to ensure required PHB performance behavior
  - Examples:
    - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
    - Class A packets leave before packets from class B

- Advantage:

  No state info to be maintained by routers

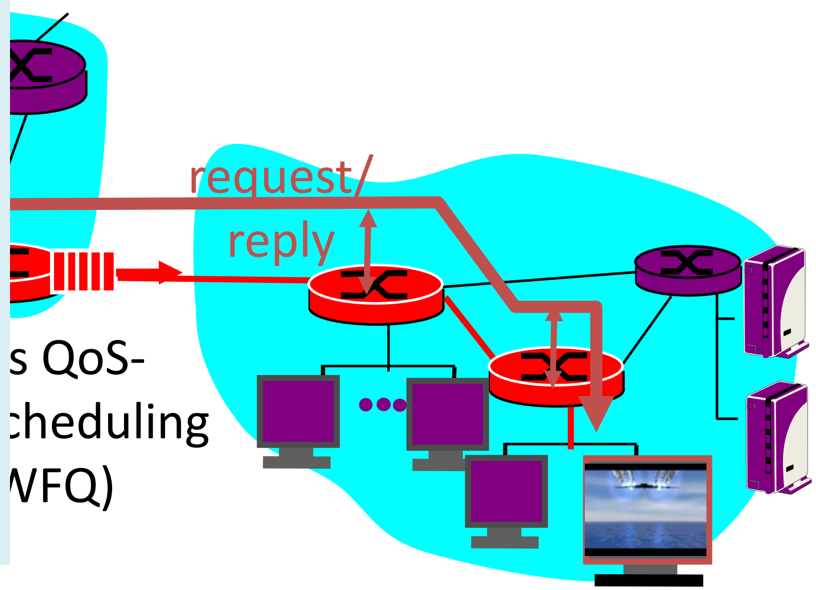
Another approach:

Intserv: QoS guarantee scenario

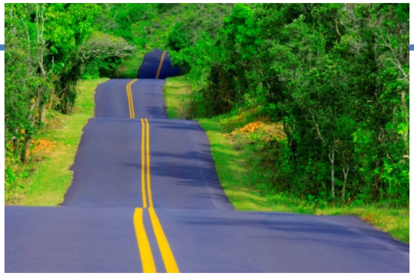
Resource reservation per individual application session (admission control, continuous)

- setup, signaling (RSVP)
- Maintains state a la VC (but soft state, ie times out)
  - responsibility at the client to renew reservations

s QoS-
cheduling
WFQ)



Roadmap



NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*

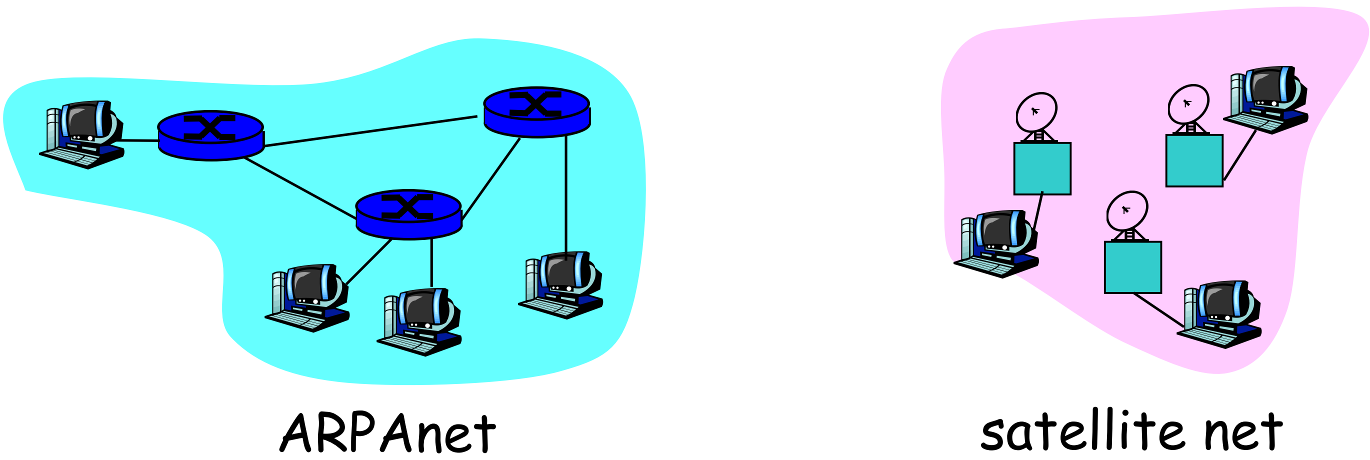
Recall:
the Internet concept: virtualizing networks

1974: multiple unconnected nets

- ARPAnet
- data-over-cable networks
- packet satellite network (Aloha)
- packet radio network

… differing in:

- addressing conventions
- packet formats
- error recovery
- routing



"A Protocol for Packet Network Intercommunication", V. Cerf, R. Kahn, IEEE Transactions on Communications, May, 1974, pp. 637-648.

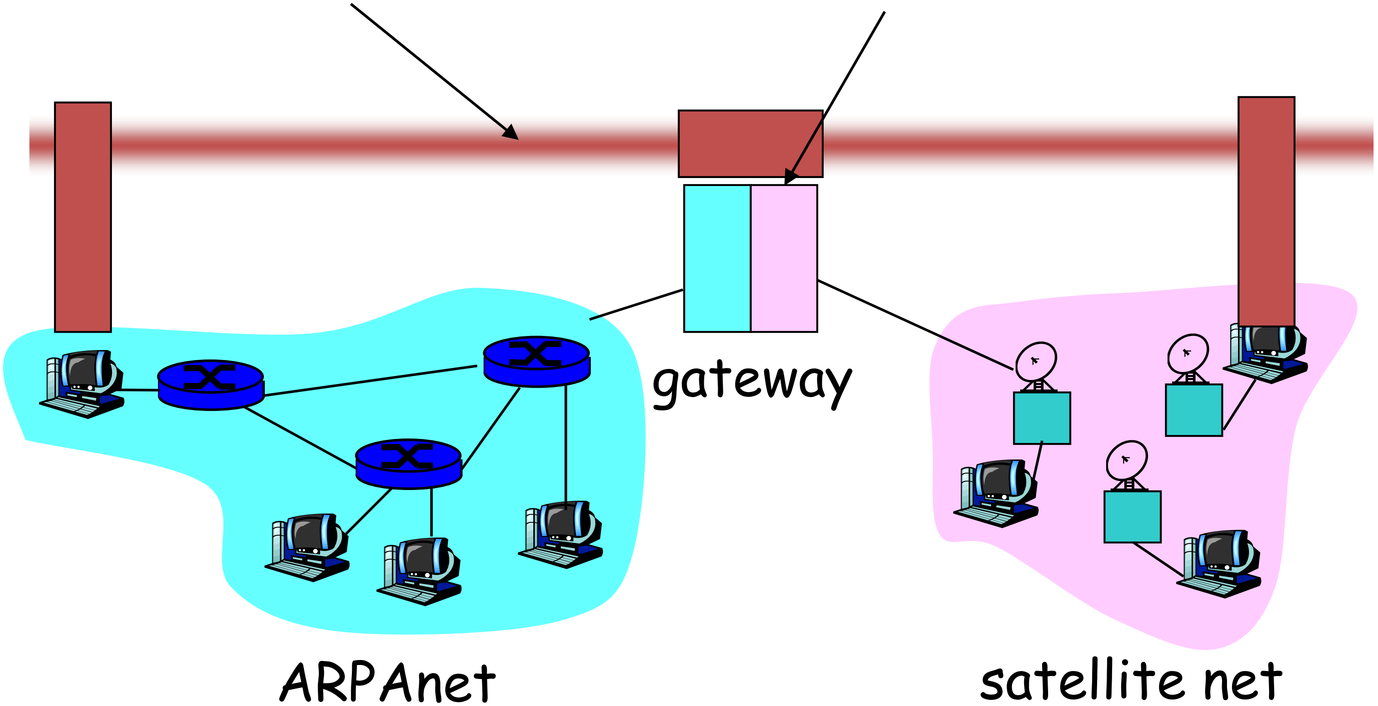
The Internet: virtualizing networks

Internetwork layer (IP):

- addressing: internetwork appears as single, uniform entity, despite underlying local network heterogeneity
- network of networks

Gateway:

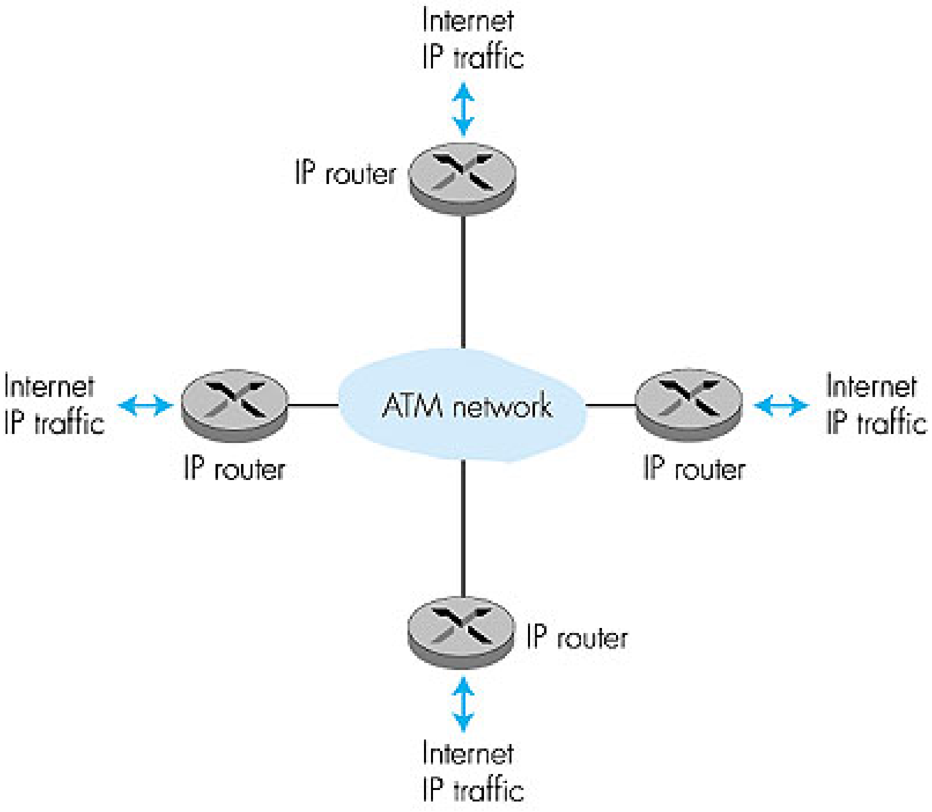
- "embed internetwork packets in local packet format"
- route (at internetwork level) to next gateway



ATM: network or link layer?

Vision: end-to-end transport: "ATM from desktop to desktop"
- ATM *is* a network technology

Reality: used to connect IP backbone routers
- "IP over ATM"
- ATM as switched link layer, connecting IP routers



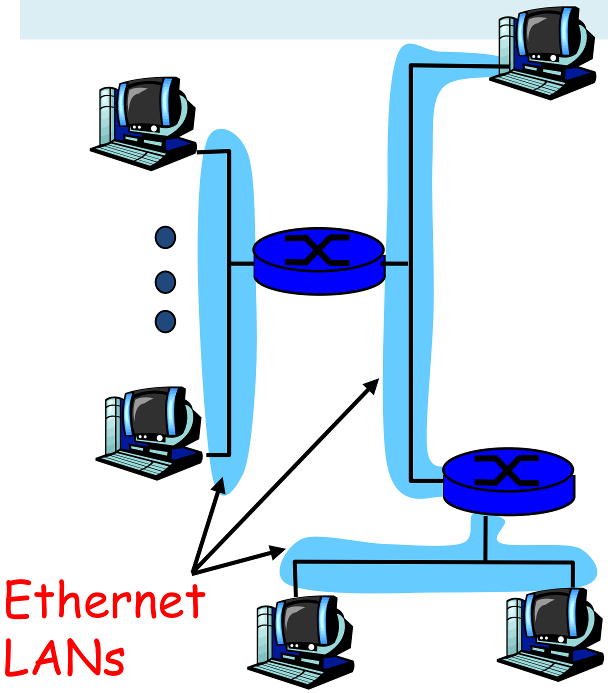
e.g. IP-Over-ATM

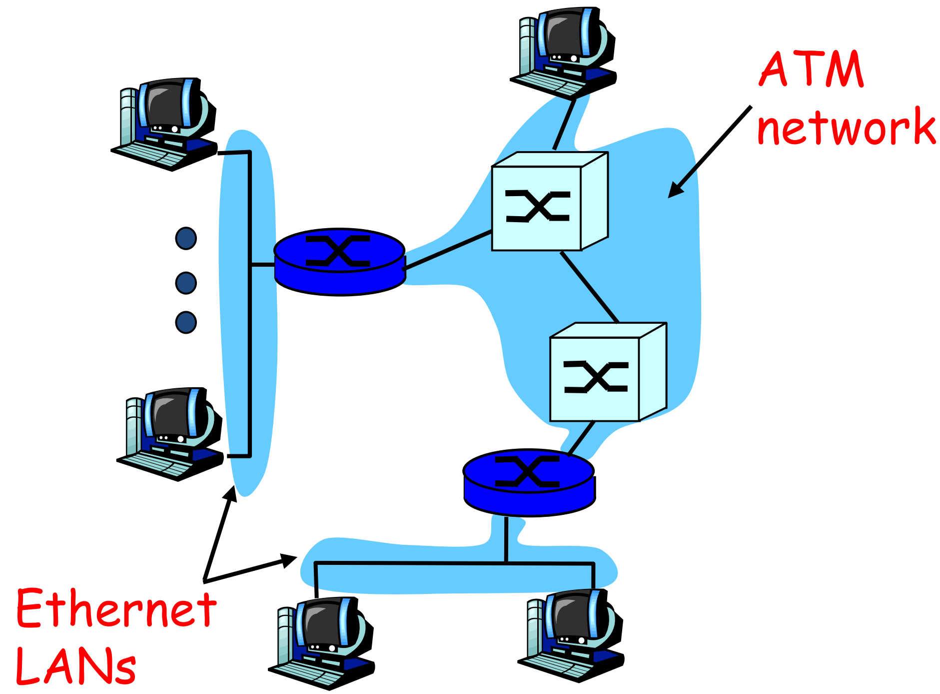
"Classic" IP over eg Ethernet

- 3 "networks" (e.g., LAN segments)
- MAC and IP addresses

IP over ATM

- ❒ replace "network" (e.g., LAN segment) with ATM network, (ATM + IP addresses)
- ❒ Run datagram routing on top of virtual-circuit routing ….





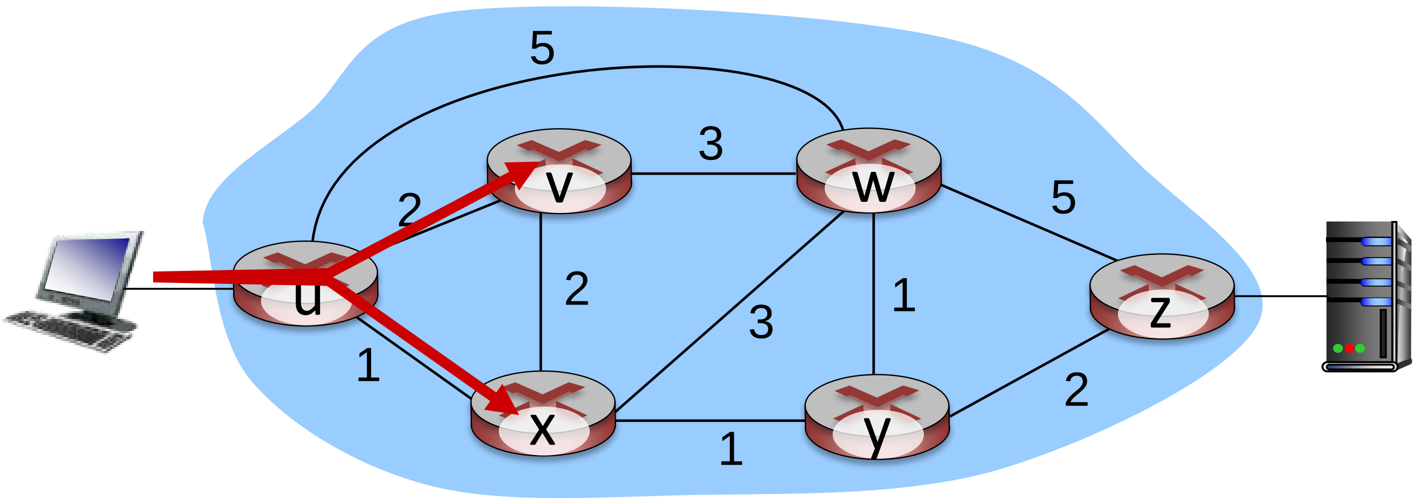
Cerf & Kahn's Internetwork Architecture

What is virtualized?

- two layers of addressing: internetwork and local network
- new layer (IP) makes everything homogeneous at internetwork layer
- underlying local network technology
  - Cable, satellite, 56K telephone modem
  - Ethernet, other LAN
  - ATM
  - **NEWER: MPLS (Multiprotocol Label Switching Protocol): for traffic engineering**

… "invisible" at internetwork layer. Looks like a link layer technology to IP

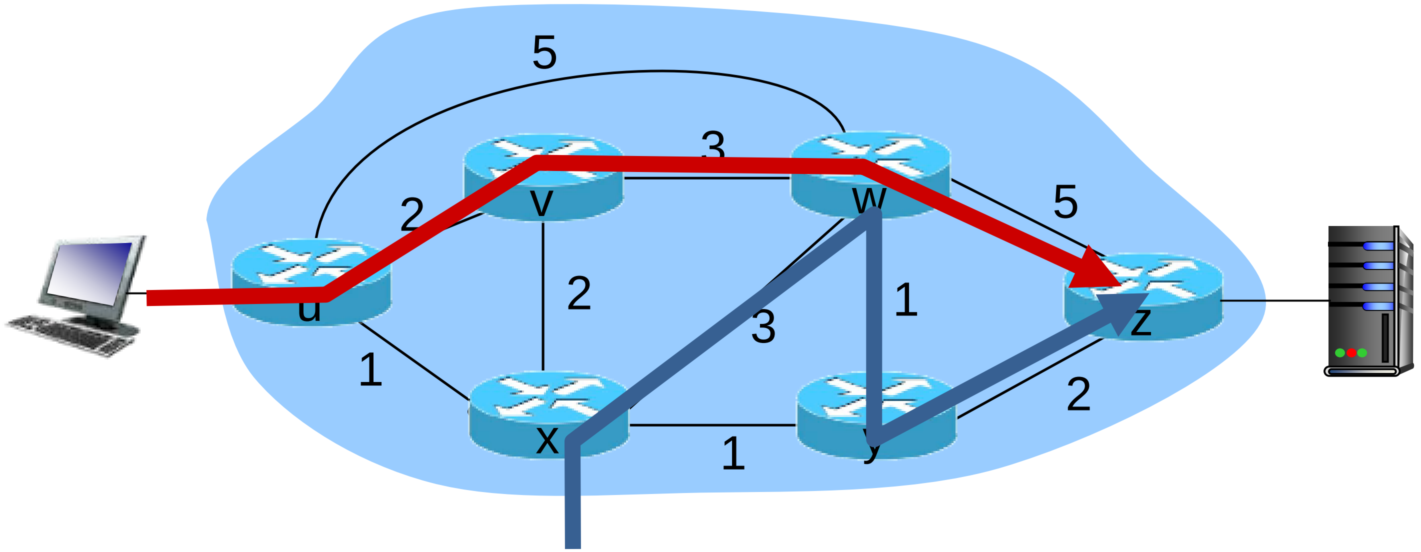
Traffic engineering: difficulties with traditional Internet routing



Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?

A: can't do it (or need a new routing algorithm)

Traffic engineering: difficulties with traditional Internet routing

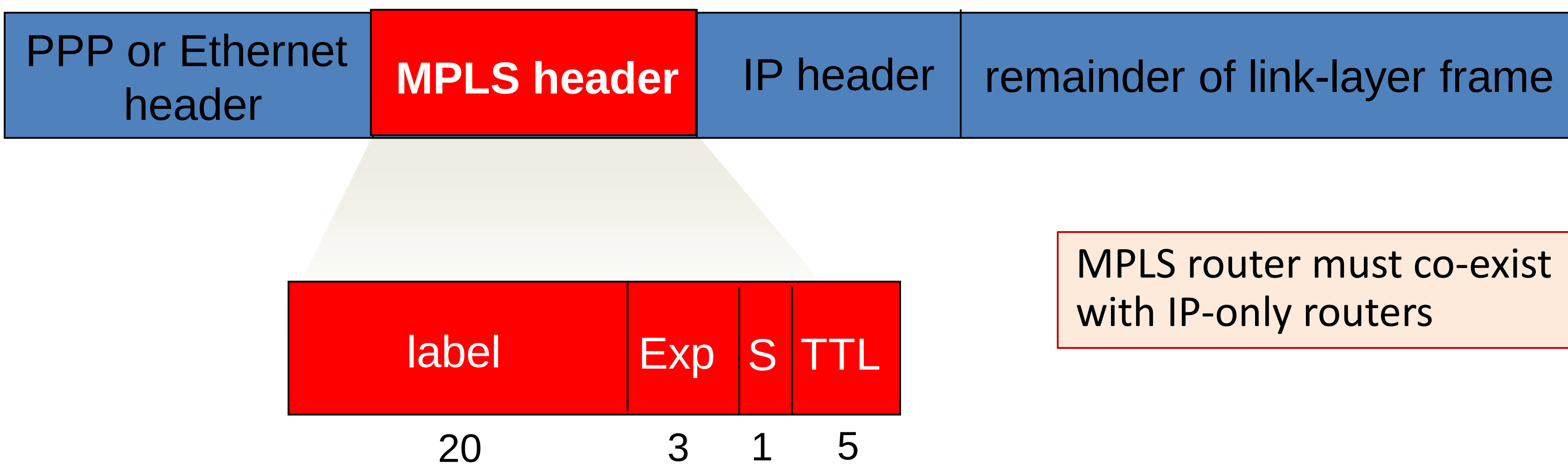


Q: what if w wants to route blue and red traffic differently?

A: can't do it (with destination based forwarding, and LS, DV routing)

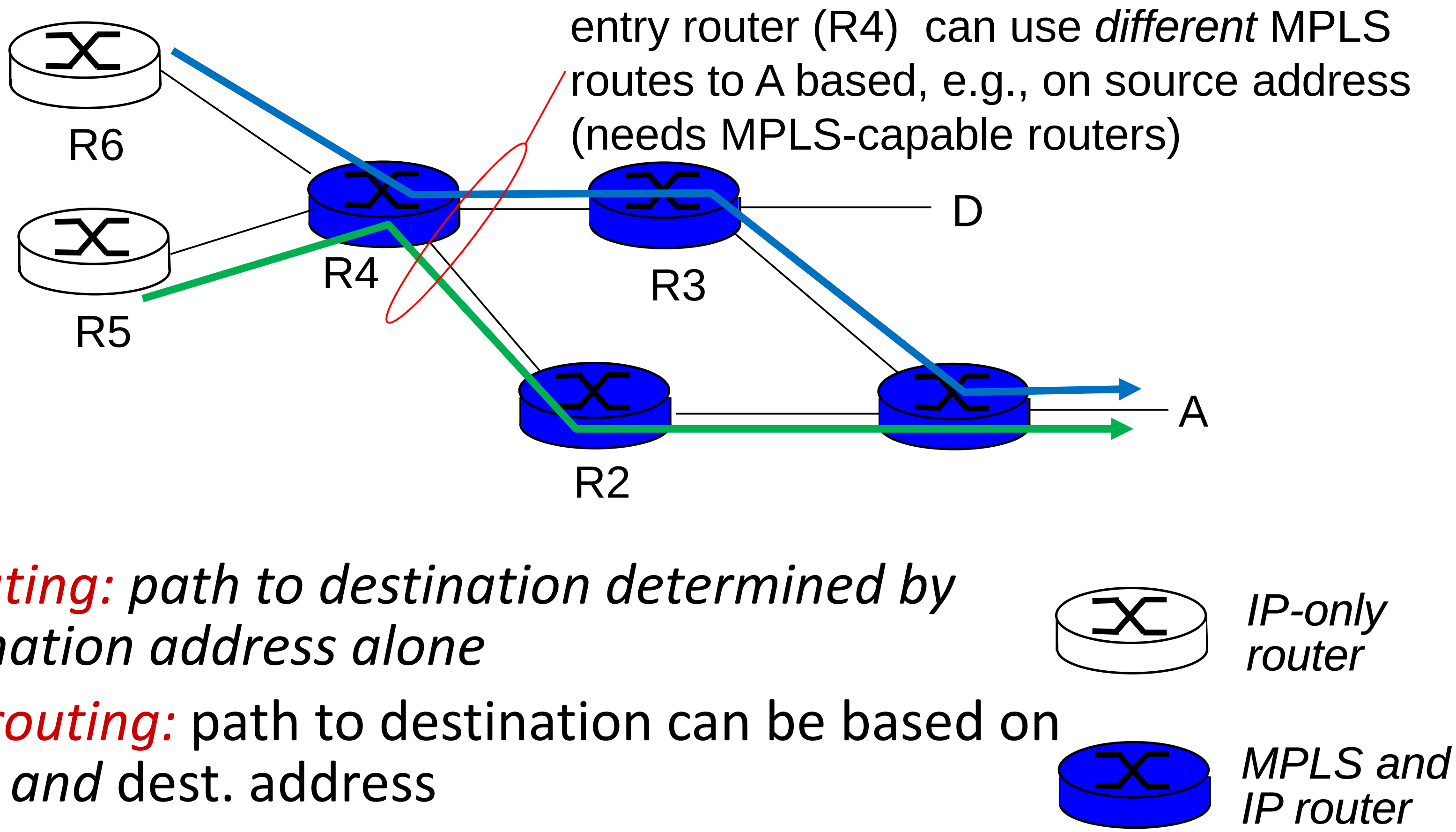
Multiprotocol label switching (MPLS) in IP networks: VC-inspired

- goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding; utilize multiple S-T paths simultanaoulsy
  - borrow ideas from Virtual Circuit (VC) approach but IP datagram still keeps IP address
- label-switched router
  - forwards packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS protocol's forwarding table distinct from IP forwarding tables



MPLS router must co-exist with IP-only routers

MPLS versus IP paths



IP routing: path to destination determined by destination address alone

MPLS routing: path to destination can be based on source *and* dest. address

fast reroute: precompute backup routes in case of link failure or congestion (eg for CDN distribution)

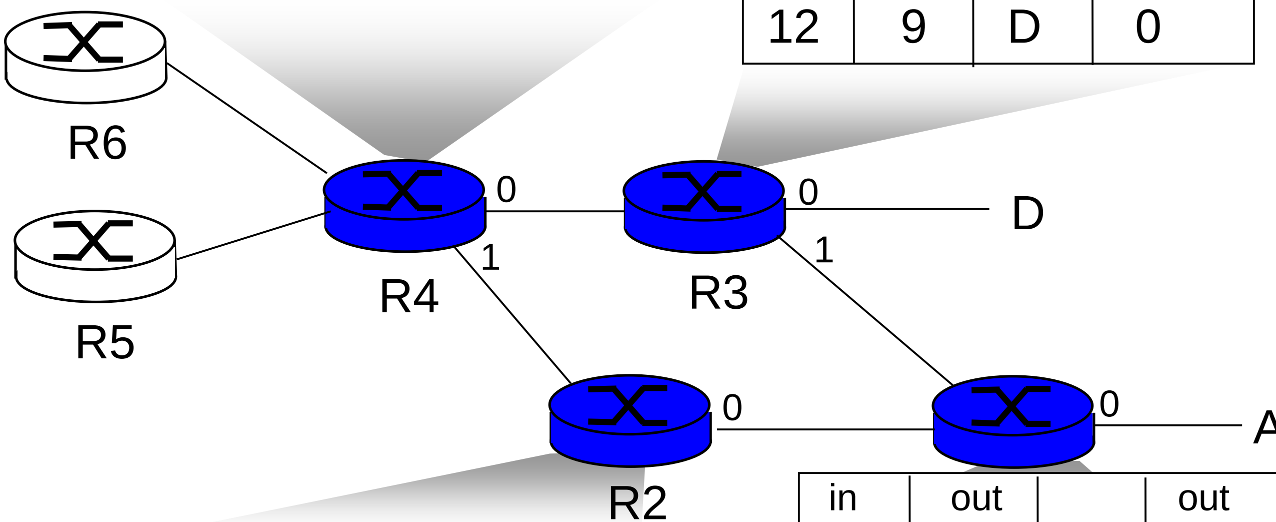
MPLS forwarding tables

PPP or Ethernet header	MPLS header	IP header	remainder of link-layer frame

Flexibilty allows Traffic Engineering (adapting Routing to suit different purposes; relates to Software-defined Networks)

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

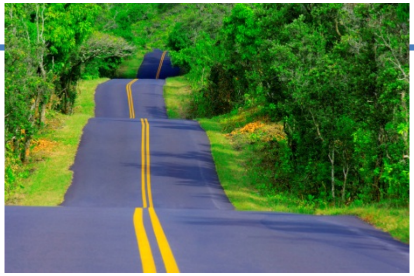
in label	out label	dest	out interface
10	6	A	1
12	9	D	0



in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	-	A	0

Roadmap

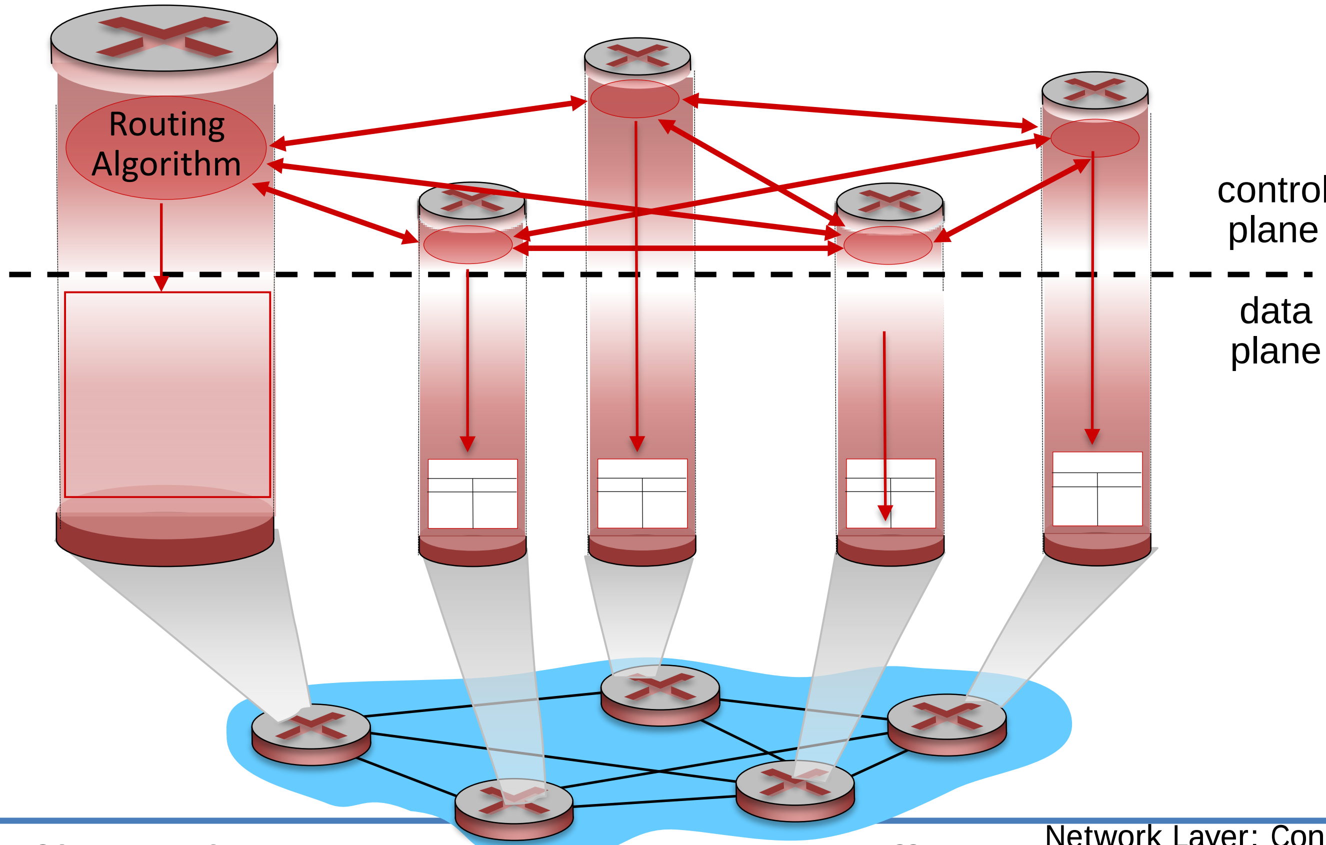


NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*

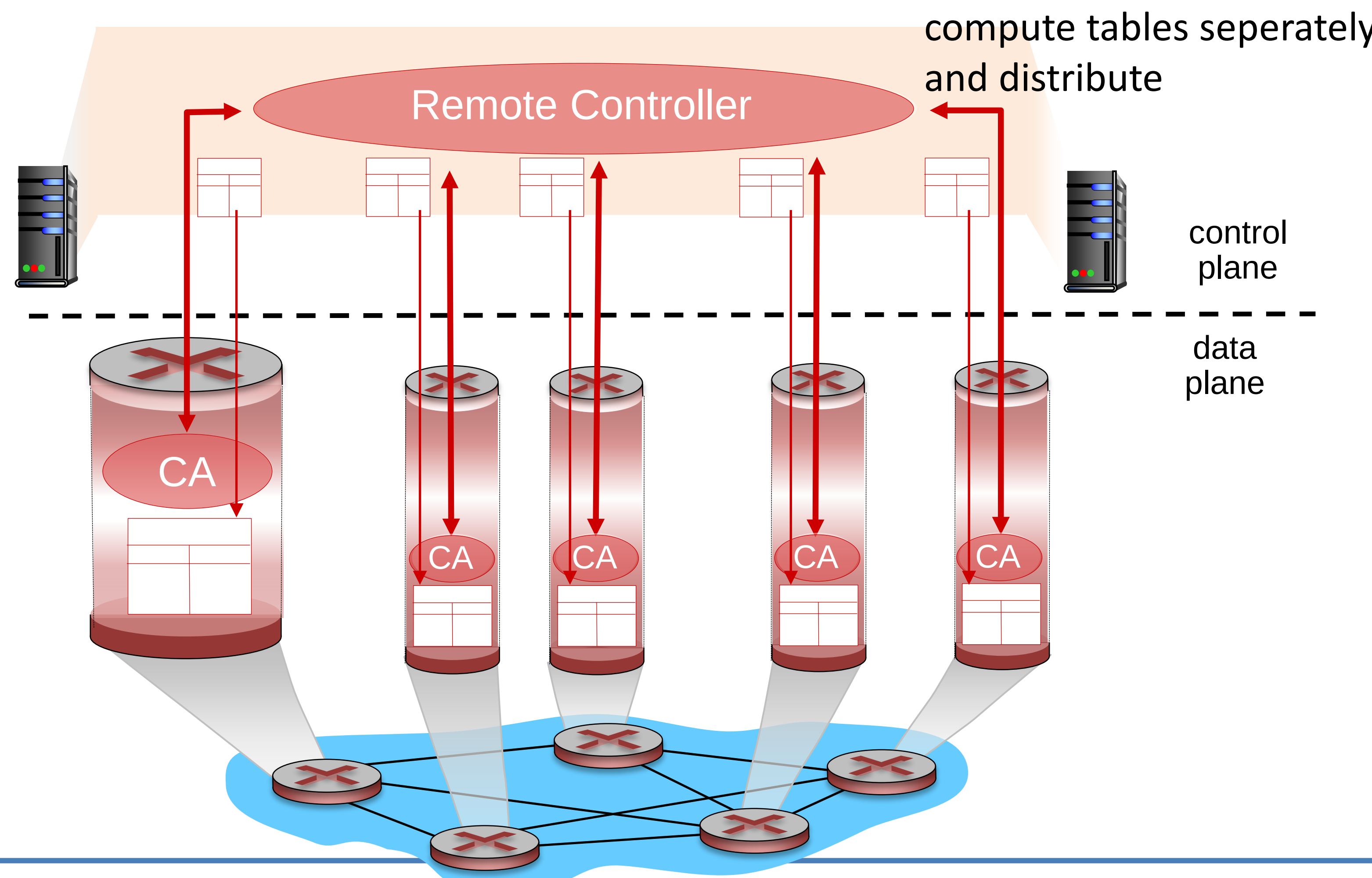
Recall: Traditional Internet, per-router control plane

Individual routing algorithm components *in each and every router* interact with each other in control plane to compute forwarding tables

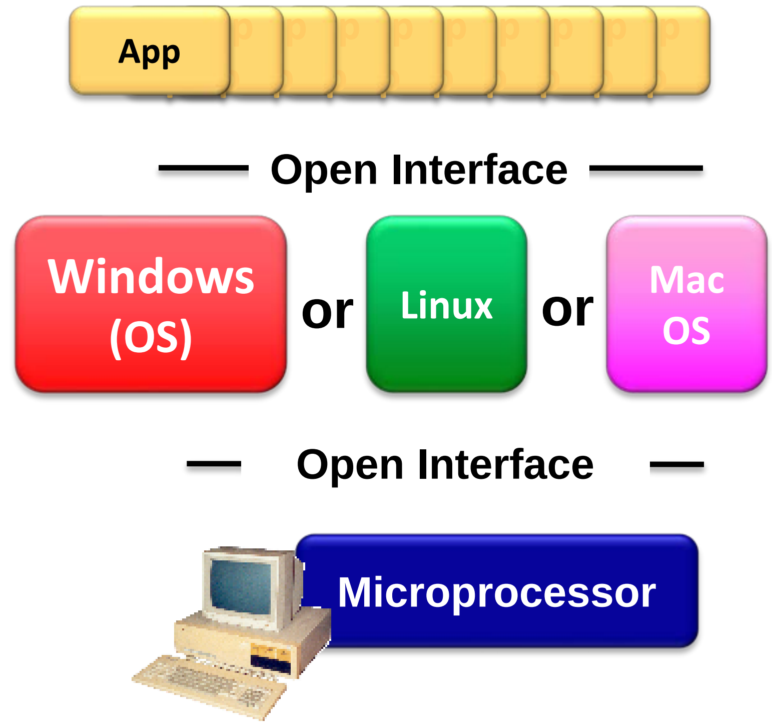


Recall: logically separated control plane

A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables



Analogy: mainframe to PC evolution*



Vertically integrated
Closed, proprietary
Slow innovation
Small industry

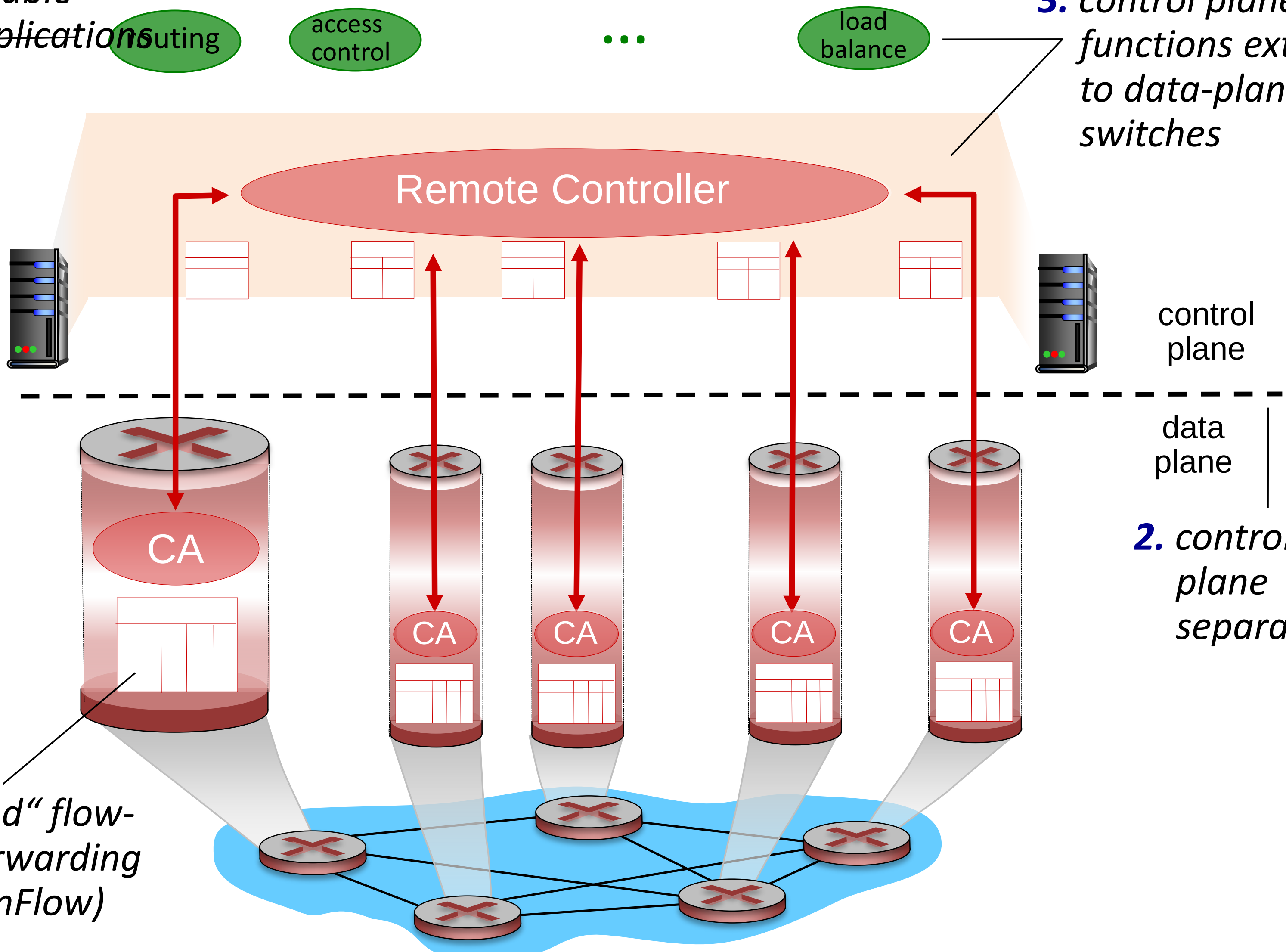
Horizontal
Open interfaces
Rapid innovation
Huge industry

* Slide courtesy: N. McKeown

Software defined networking (SDN)

4. *programmable control ~~applications~~*

3. *control plane functions external to data-plane switches*



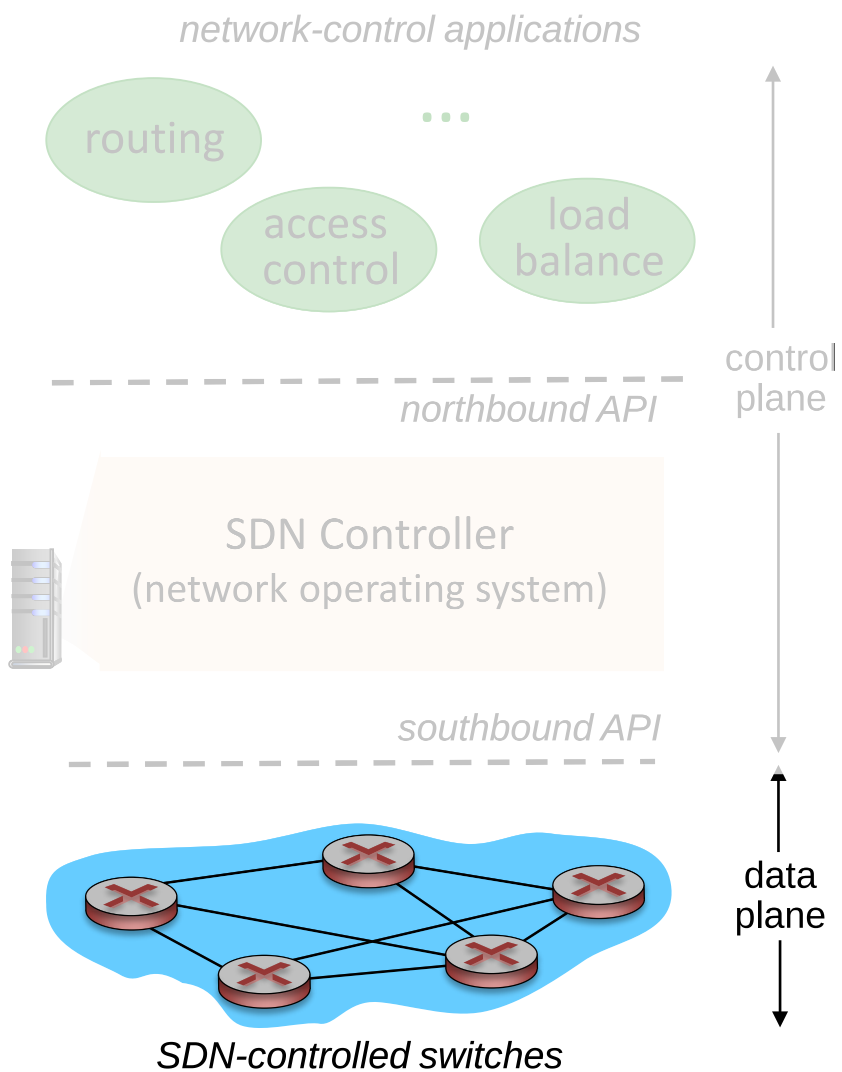
2. *control, data plane separation*

1: *generalized" flow-based" forwarding (e.g., OpenFlow)*

SDN perspective: data plane switches

Data plane switches

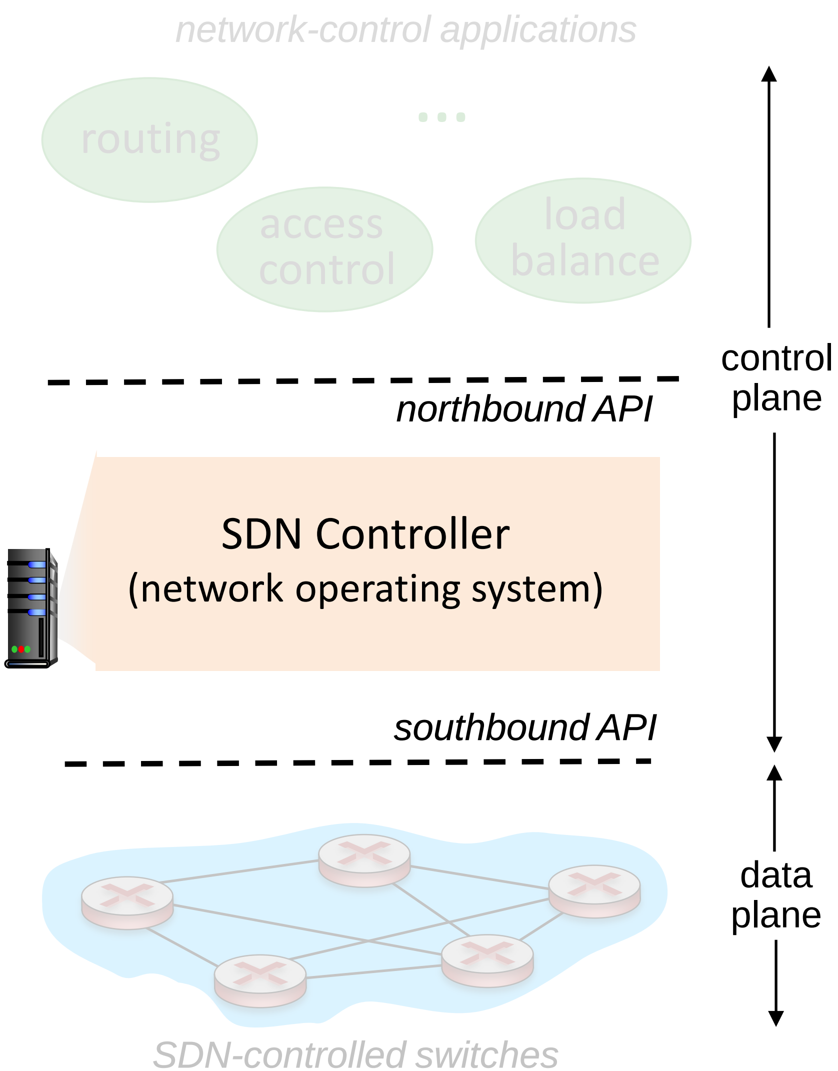
- fast, simple implementing generalized data-plane forwarding in hardware
- switch flow table computed by controller
- API for table-based switch control (e.g., OpenFlow)
- protocol for communicating with controller (e.g., OpenFlow)



SDN perspective: SDN controller

SDN controller (network OS):

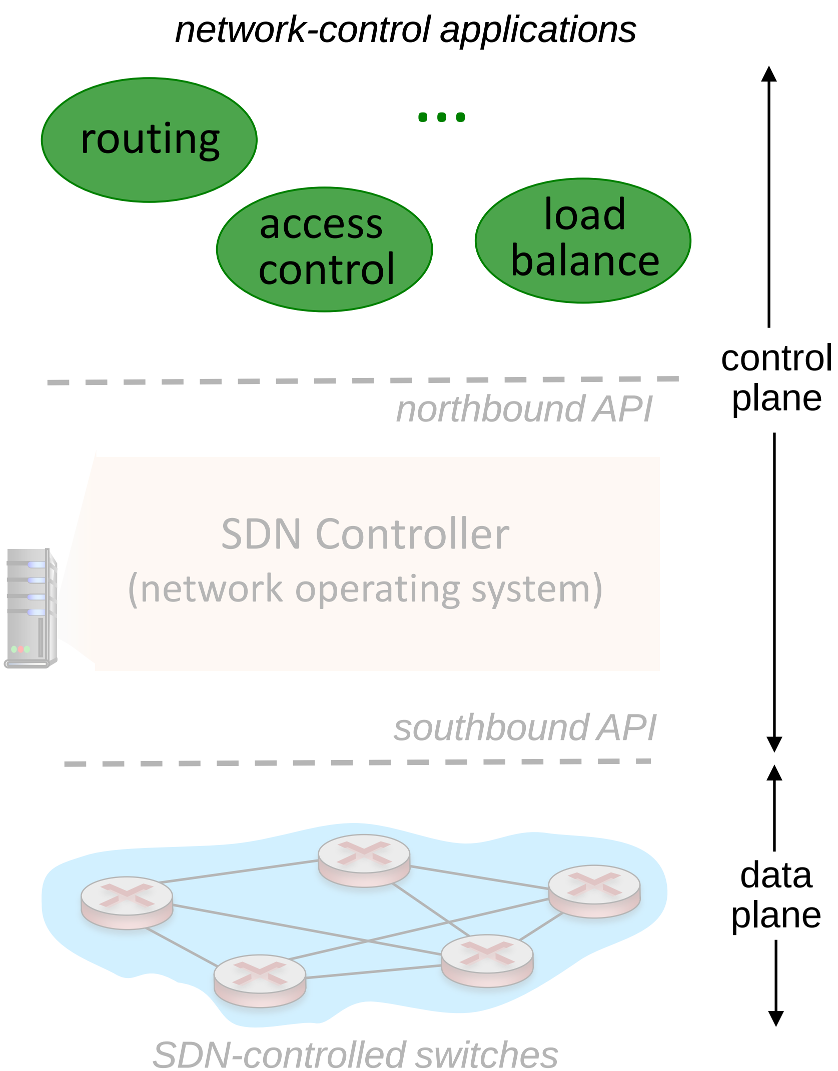
- maintain network state information
- interacts with network control applications "above" via northbound API
- interacts with network switches "below" via southbound API
- implemented as ***distributed system*** for performance, scalability, fault-tolerance, robustness



SDN perspective: control applications

network-control apps:

- "brains" of control: implement control functions using lower-level services, API provided by SDN controller
- *unbundled:* can be provided by 3rd party: distinct from routing vendor, or SDN controller

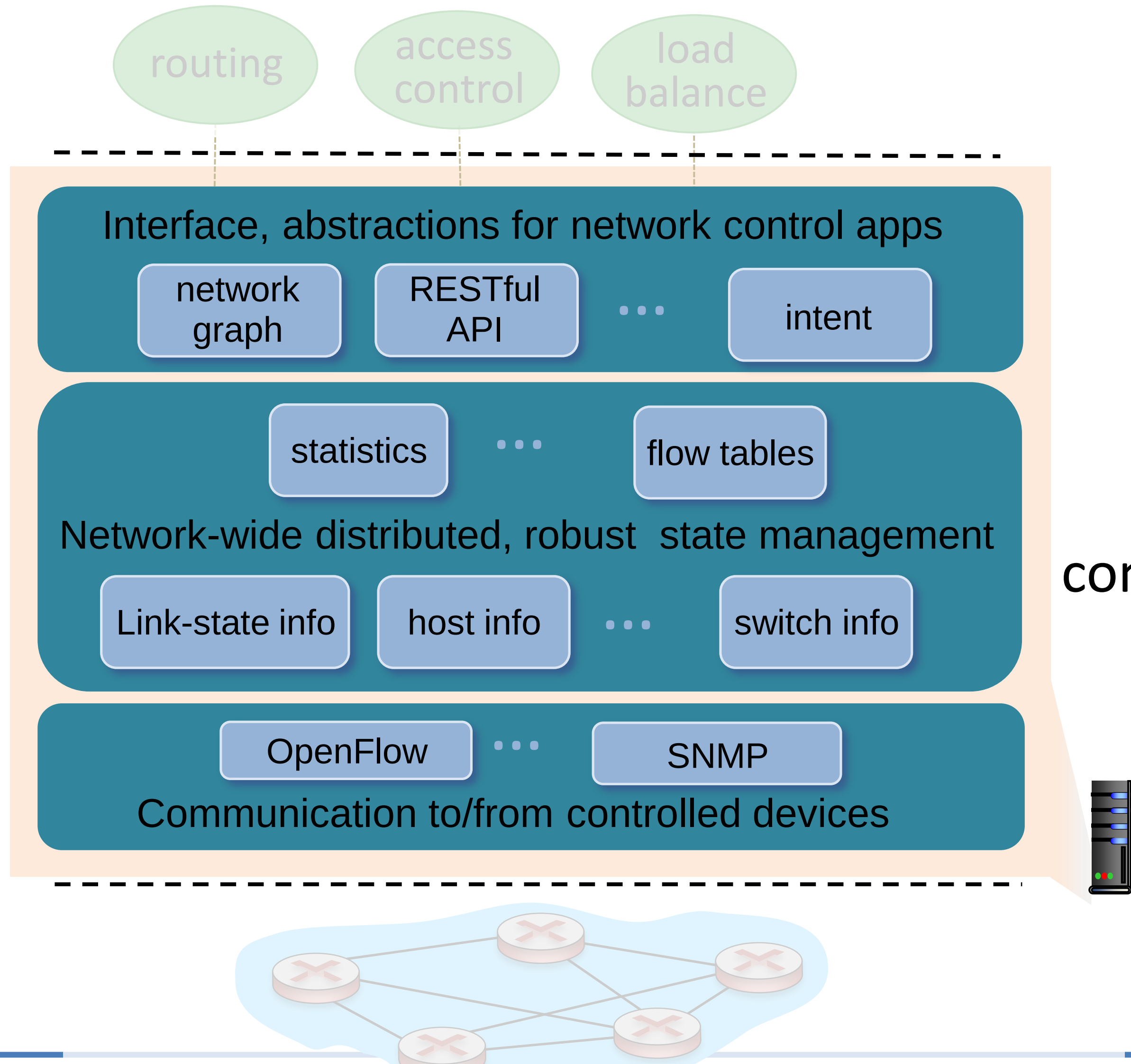


Components of SDN controller

Interface layer to network control apps: abstractions API

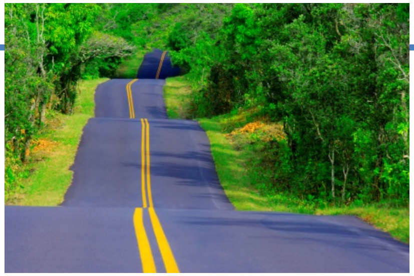
Network-wide state management layer: state of networks links, switches, services: a *distributed database*

communication layer: communicate between SDN controller and controlled switches



SDN controller

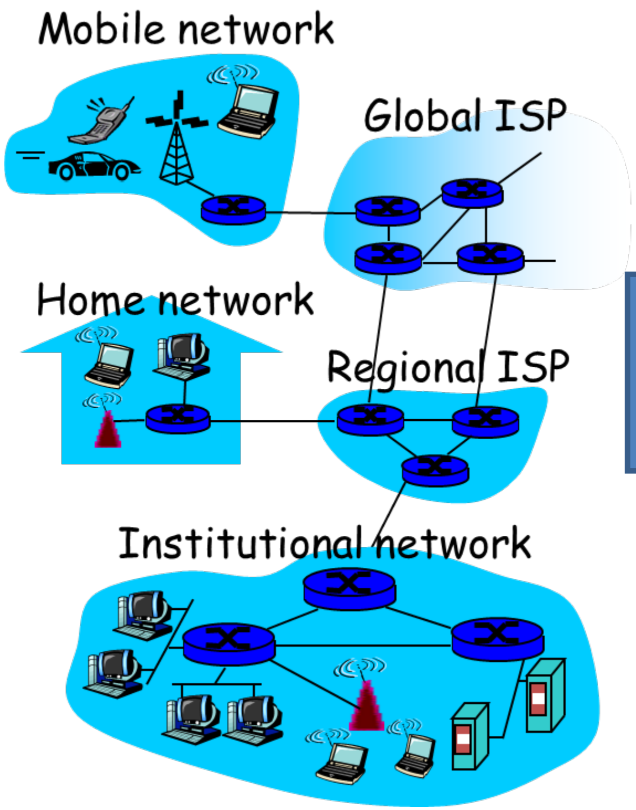
Roadmap

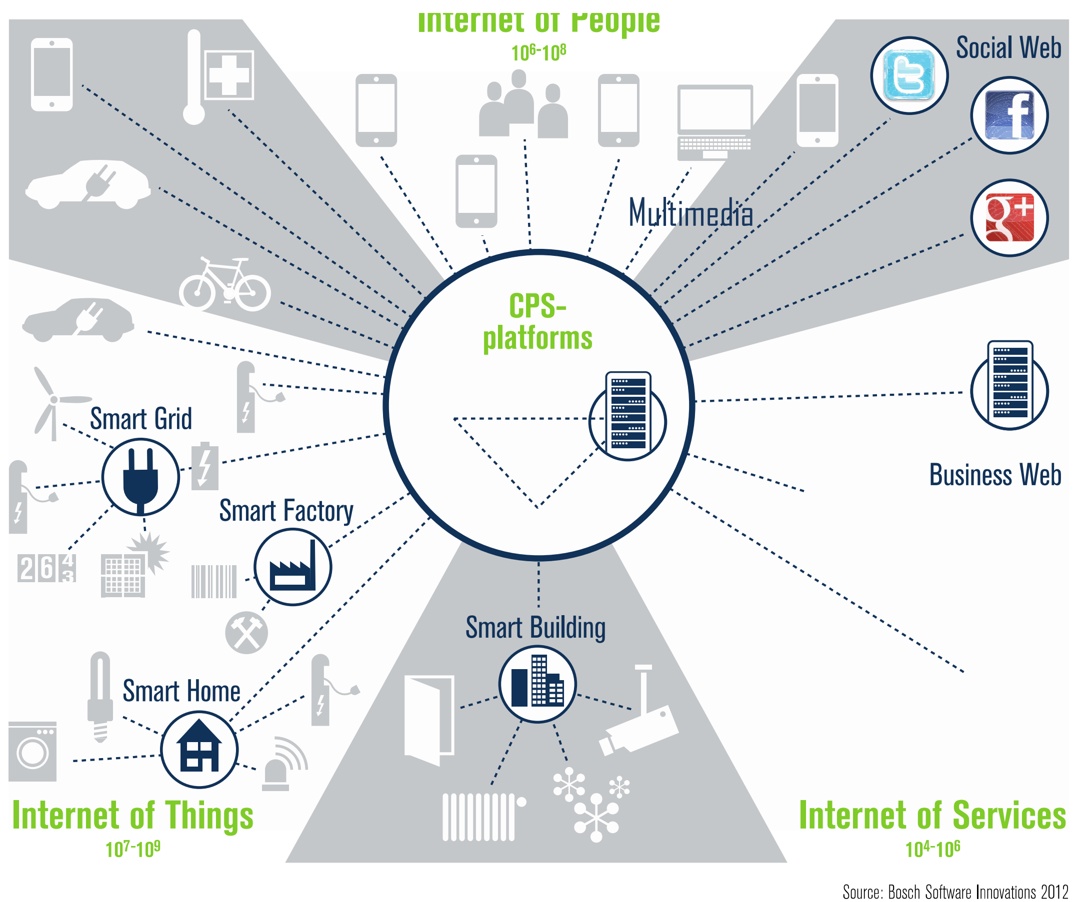


NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*

Recall: Internet & its context….

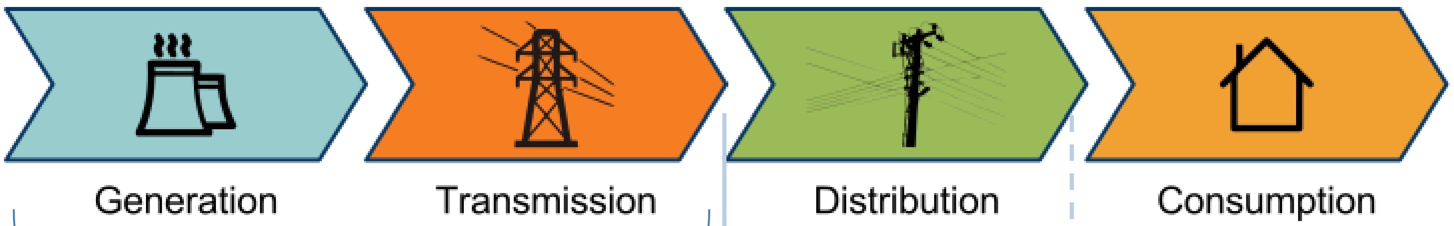




approx 10 yrs ago

continuous evolution ….

Example:
Data networking technologies in Smart Grids

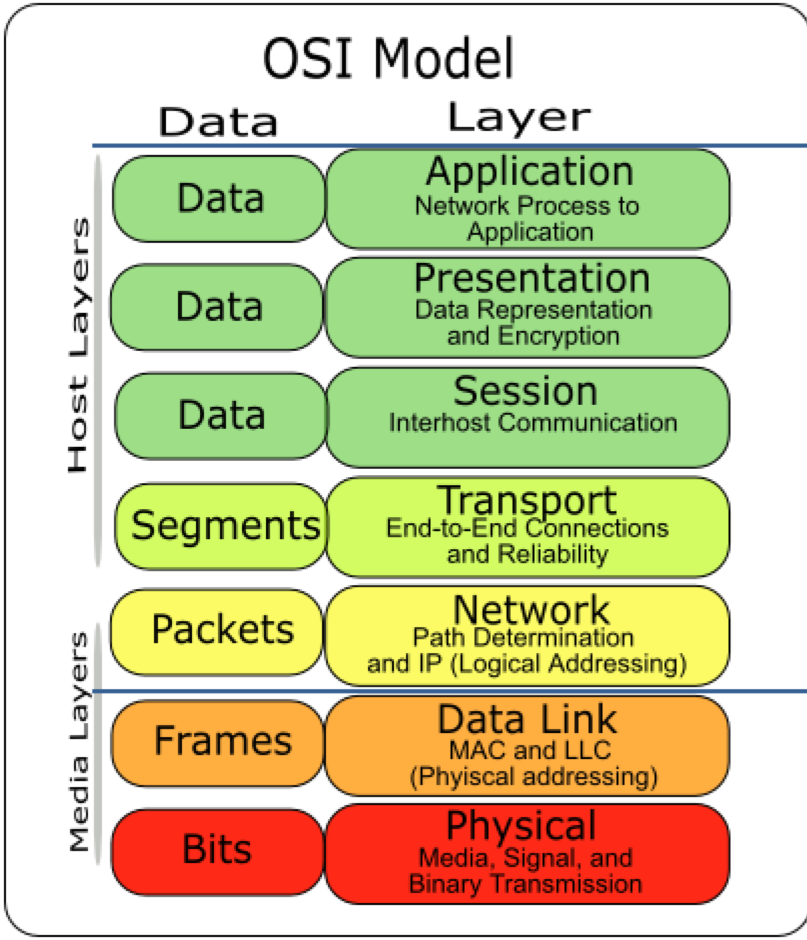


"Here be SCADA"

	Generation / Transmission	Distribution	Consumption
Communication technologies	• **SONET - Synchronous Optical Network** • **WDM - Wavelength Division Multiplexing** • **Satellite/VSAT** • **IP Radio**	• **Twisted pair / Fiber optics** • **BPL - Broadband over Power Lines** • **WiMax - Worldwide Interoperability for Microwave Access** • **GPRS**	
Communication protocols	**Ethernet rules~**	**Ethernet rules!** **(IEC 61850)**	**Ethernet rules?**  **Hint: not always →**

Slides: Giorgos Georgiadis

Approximate overview of shaping new stacks



Protocols @ Distribution's last mile

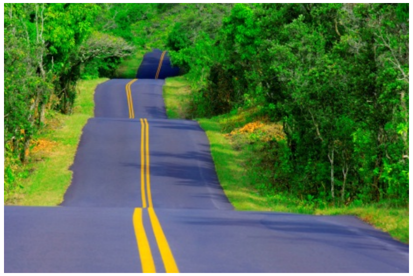
OpenADR REST-based (i.e. CoAP)				
XMPP BACNet LonWorks Modbus	WiFi	ZigBee	6LoW PAN	HomePlug
Proprietary	Ethernet / Gigabit Ethernet	IEEE 802.15.4		Proprietary, part 2: HomePlug

Slides: Giorgos Georgiadis
(see extra slides for more refs&notes)

Summary & Study list

NW support for multimedia / QoS: *[Ch. 7.5 (7e 9.5)]*

- **Improving timing/QoS guarantees in Networks** (also related with congestion-control): Packet scheduling and policing
- **A VC (ATM) approach** *[incl. Ch 3.62-3.6.3 (7e 3.7.2)]*
- **Internet approaches**
  - Diff-serv, Int-serv + RSVP,
  - **Traffic Engineering** MPLS *[incl. ch 5.5. (7/e 6.5)]*
- **SDN** *[cf separate notes @pingpong docs, 7e ch 4.4, 5.5)]*
- **Internet-of-Things in evolution: more types of traffic/devices…** *[optional study, just browse example protocols mentioned]*



- Internet core and transport protocols do not provide guarantees for multimedia streaming traffic
- Applications take matters into own hands
  - interesting evolving methods
- Another type of service at the core (VC-like) would imply a different situation
  - Internet core is re-shaping (Traffic engineering, SDN, Intserv & Diffserv
- Internet-of-Things in evolution
  - even more types of traffic…

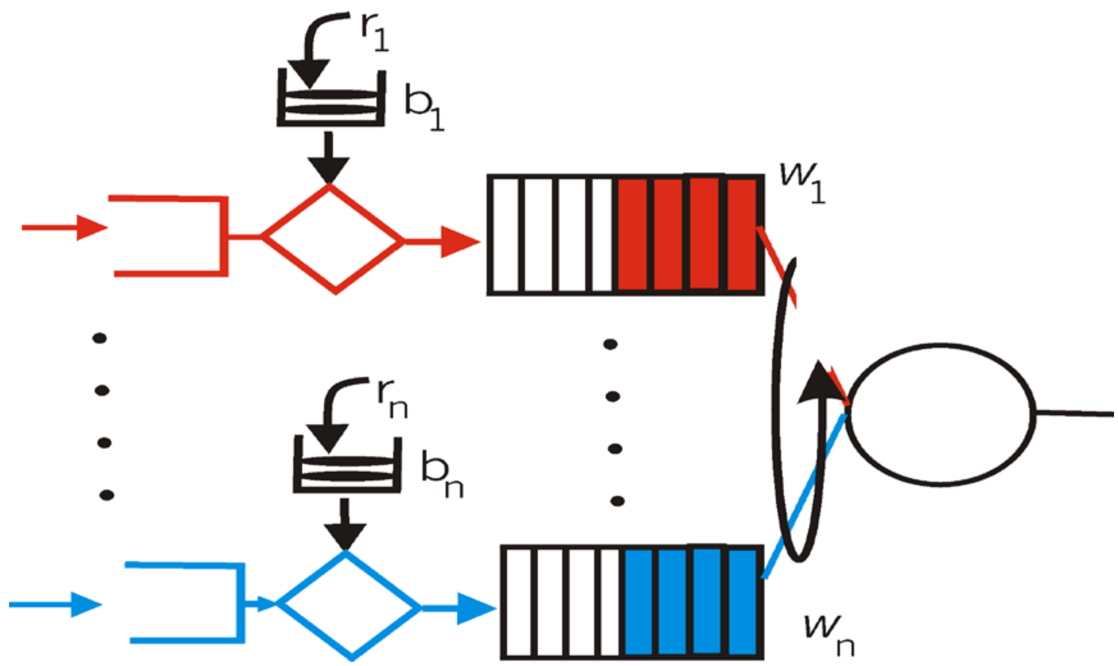
Extra slides/notes for further study

Token bucket + WFQ…

…can be combined to provide upper bound on packet delay in queue:

- b_i packets in queue, packets are serviced at a rate of at least $R \cdot w_i/\sum (wj)$ packets per second, then the time until the last packet is transmitted is at most

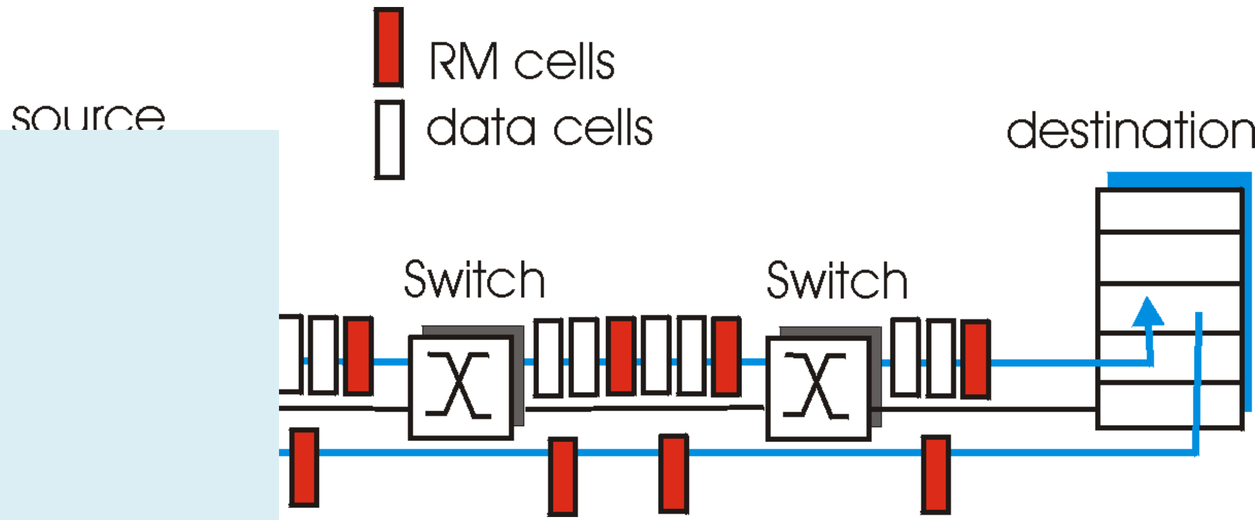
$$b_i /(R \cdot w_i/\sum (wj))$$



ATM ABR congestion control

ABR: available bit rate:

- "elastic service"



rk-assisted")

ngestion)

te ER (explicit rate) field in RM cell

n cell

ortable rate on path

Traffic Shaping and Policing in ATM

Enforce the QoS parameters: check if *Peak Cell Rate (PCR)* and *Cell Delay Variation* (CDVT) are within the negotiated limits:

Generic Cell Rate Algo**:** introduce:

expected next time for a successive cell based on T = 1/PCR

border time L (= CDVT) < T in which next transmission may start (but never before T-L)

A nonconforming cell may be discarded or its *Cell Loss Priority* bit be set, so it may be discarded in case of congestion

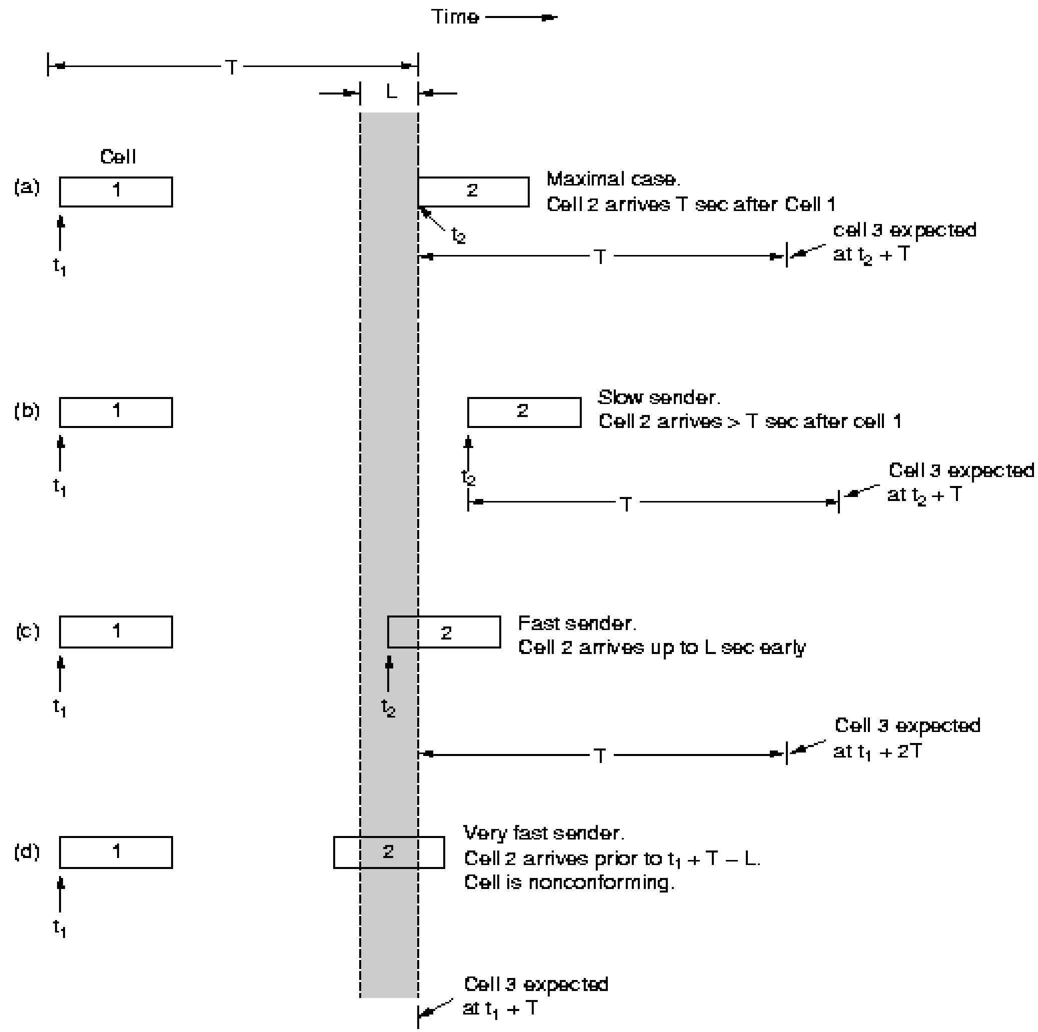


Fig. 5-73. The generic cell rate algorithm.

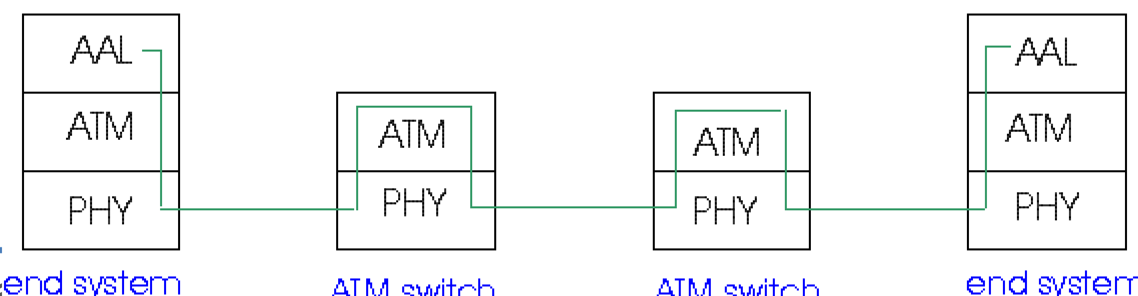
From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

ATM Adaptation (Transport) Layer: AAL

Basic idea: cell-based VCs need to be "complemented "to be supportive for applications.

- Several ATM Adaptation Layer (AALx) protocols defined, suitable for different classes of applications
  - AAL1: for CBR (Constant Bit Rate) services, e.g. circuit emulation
  - AAL2: for VBR (Variable Bit Rate) services, e.g., MPEG video
  - .....

- "suitability" has not been very successful
- computer science community introduced AAL5, (simple, elementary protocol), to make the whole ATM stack usable as switching technology for data communication under IP!



dia+ATM;QoS,

Network support for multimedia

Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None of soft	Packet market, scheduling, policing.	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

Software defined networking (SDN)

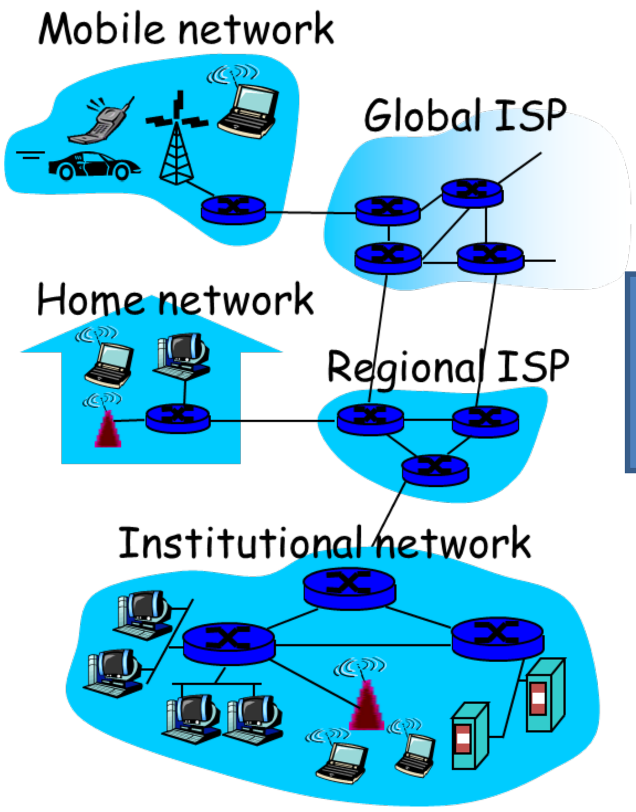
- Internet network layer: historically has been implemented via distributed, per-router approach
  - *monolithic* router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different "middleboxes" for different network layer functions: firewalls, load balancers, NAT boxes, ..

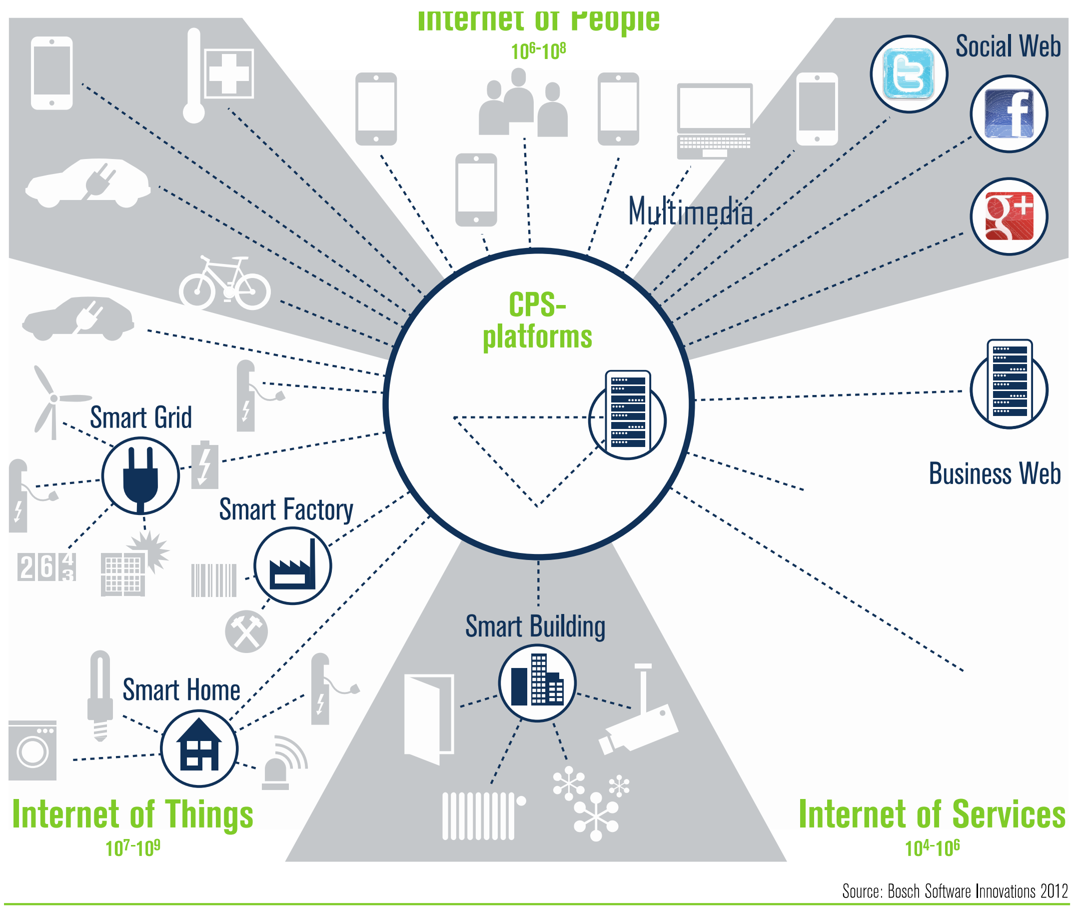
- ~2005: renewed interest in rethinking network control plane

Data networking technologies in Smart Grids

Presentation by
Giorgos Georgiadis
(former CTH / curr. Bosch R&D)

Recall: Internet & its context….





approx 10 yrs ago

continuous evolution ….

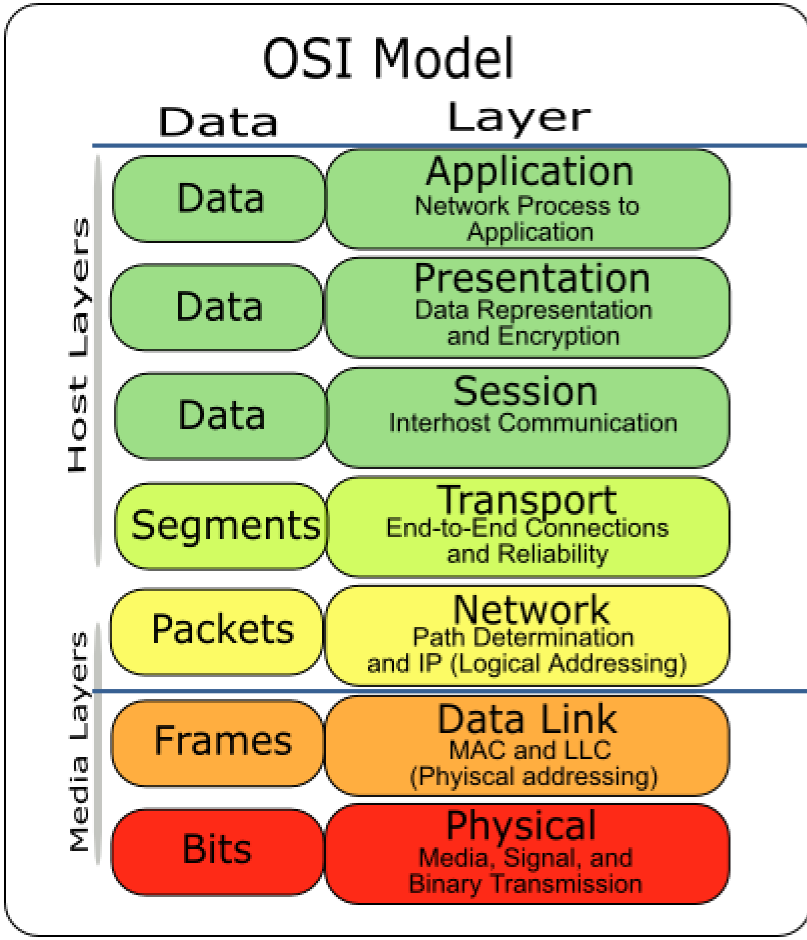
Introduction

“Here be SCADA”

	Generation / Transmission	Distribution	Consumption
Communication technologies	• SONET - Synchronous Optical Network • WDM - Wavelength Division Multiplexing • Satellite/VSAT • IP Radio	• Twisted pair / Fiber optics • BPL - Broadband over Power Lines • WiMax - Worldwide Interoperability for Microwave Access • GPRS	
Communication protocols	Ethernet rules~	Ethernet rules! (IEC 61850)	Ethernet rules?  Hint: not always

Fig. Giorgos Georgiadis

Approximate overview of shaping new stacks



Protocols @ Distribution's last mile

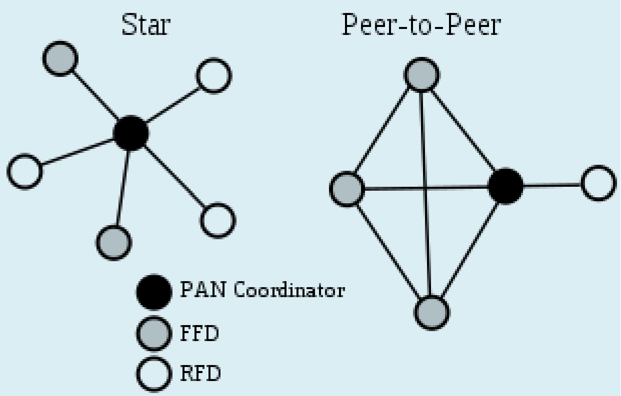
OpenADR REST-based (i.e. CoAP)				
XMPP BACNet LonWorks Modbus	WiFi	ZigBee	6LoW PAN	HomePlug
Proprietary	Ethernet / Gigabit Ethernet	IEEE 802.15.4		Proprietary, part 2: HomePlug

PHY/DataLink protocols

- Ethernet
  - Not much to say
- HomePlug
  - Honorable mention: popular home automation protocol
  - Powerline based
  - Speed: ~200mbps
  - Otherwise, vanilla protocol:
    - i.e. using TDMA,
    - Two kinds of nodes,
    - …

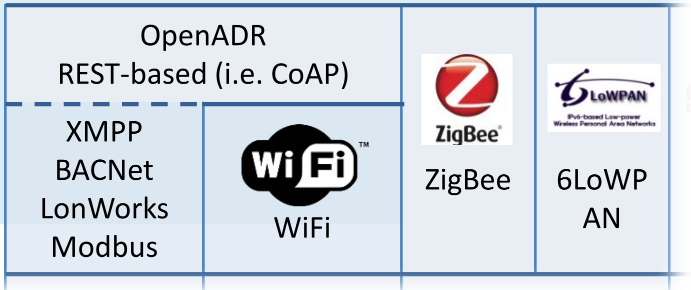
PHY/DataLink protocols

- IEEE 802.15.4
  - Radio based, usually 2.4GHz
  - Small packets (<=127bytes)
  - Medium speed (~250kbps)
  - Originally DSSS
  - Topologies supported:
    - Star
    - Peer-to-peer
  - Roles supported:
    - Full-function device
    - Reduced-function device

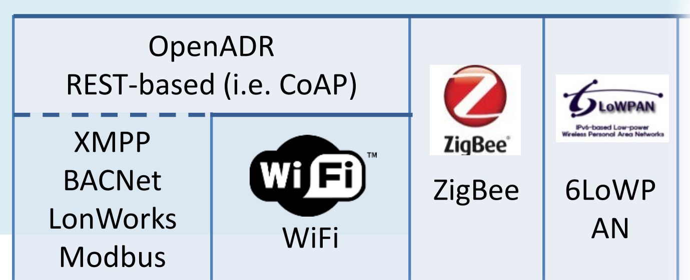


Higher protocols

- 6LoWPAN
  - "IPv6 over LoW Power wireless Area Networks"
  - Builds on 802.15.4, IPv6
  - Aimed at low power devices (sensors, controllers)
  - Topologies
    - Star, peer-to-peer **+ Mesh**
  - Many Challenges:
    - IP packets >=1280bytes (!)
    - 128bit IP addresses
    - …

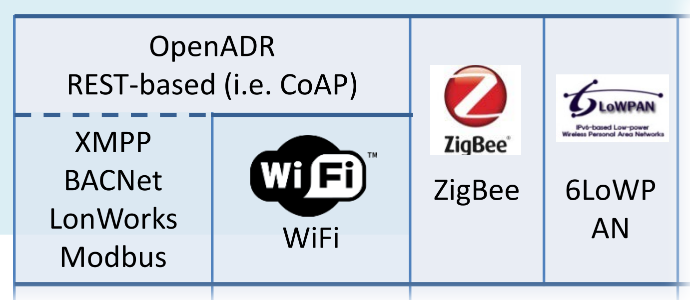


- ZigBee
  - Builds on 802.15.4, **but not IP**
  - Aimed at low power devices too (sensors, controllers)
    - Speed 250kbps
    - Packet 127bytes
    - Battery powered devices (supports sleep)
  - Topologies supported
    + Mesh (jump to: example)
  - Roles supported
    - Coordinator, router, end node
  - Different profiles exist:
    - ZigBee Home Automation
    - Zigbee Smart Energy
    - **Zigbee IP, ...**



Higher protocols

- More protocols, same story:
  - XMPP, BACNet, LonWorks, Modbus, …
  - Wired
  - Proprietary, build around specific companies (BACNet, LonWorks) or legacy protocols (Modbus)
  - Today gateway devices to “break out” to Ethernet are in use
  - Simple topologies (i.e bus), same roles as before
- But what is the connecting thread over all?
  - Open standards!
  - Internet! (of Things?)



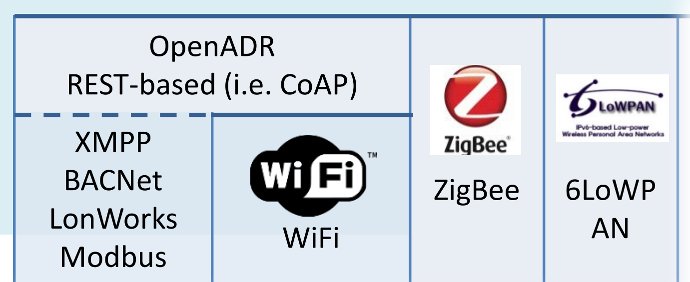
Towards interoperability

- OpenADR
  - ADR: Advanced Metering Response
  - Trying to ‘unify’ different solutions in a high level protocol
  - Formalizing:
    - Roles
    - Messages
    - Device detection
  - Simple topologies (i.e bus), same roles as before

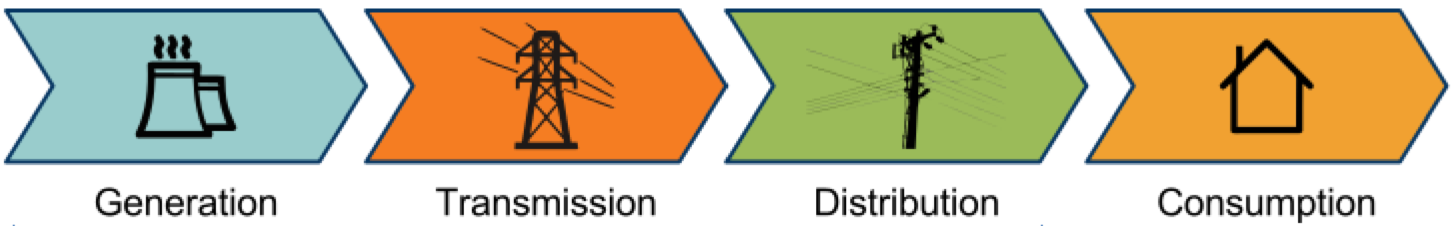
- REST-based APIs
  - I.e. Costrained Application Protocol
  - Ultimately, HTTP-based
  - Verb oriented: GET, PUT, DELETE, …

Towards interoperability

- Ethernet/IP-based integration
  - Remember:
    - Radio band: 2.4GHz (WiFI, ZigBee, 6LoWPAN)
    - Similar topologies, roles
    - Made for low energy devices, but flops/watt/kr increase!
    - Ethernet gateways commonly used
  - Solution: make them (formally) interoperable
    - ZigBee Smart Energy v2.0
    - ZigBee, WiFi, HomePlug on board
    - 6LoWPAN coming soon



Conclusion



- Ethernet + misc communication technologies
- Ethernet vs non-ethernet
  - Why?
    - Design for low energy devices (smaller packets, lower comm speed)
    - Peer to peer, mesh topologies
  - Now + Future?
    - Devices' specs catching up
    - Importance of being connected (to the Internet?)
    - Topologies still important (i.e. reliability)
    - Will probably remain radio-based