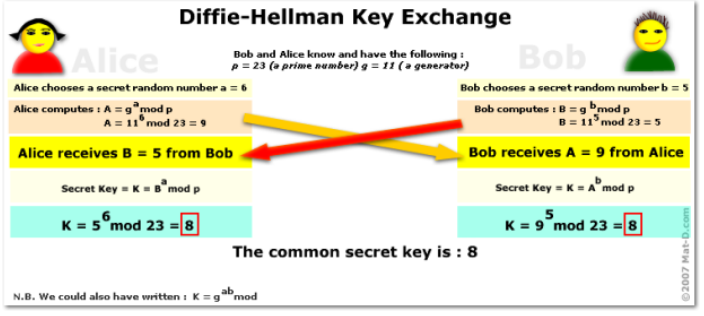
MISC: SECURITY AND QOS

The Diffie-Hellman Key Exchange

- A mechanism to establish secret keys without the need for CAs
- Based on the difficulty of computing discrete logarithms of large numbers
  - Public (or exchanged) information
    - p (prime number) as modulus, g (primitive root modulo p)
  - Private to one person
    - a for Alice, b for Bob (integer from 0 to p-1)
  - Computed and exchanged (therefore public)
    - $A=g^a \bmod p$, $B=g^b \bmod p$
  - Alice computes
    - $S=B^a \bmod p$
  - Bob computes
    - $S=A^b \bmod p$



https://shyamapadabatabyal.files.wordpress.com/2014/07/diffiehellman.png?w=618

Network support for multimedia

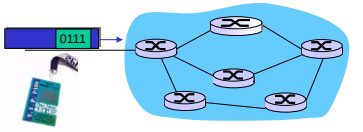
Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None of soft	Packet market, scheduling, policing.	med	some
Per-connection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

Dimensioning best effort networks

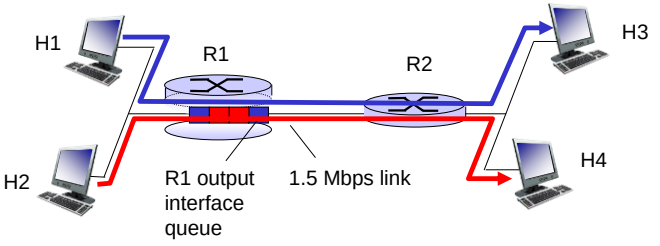
- *approach:* deploy enough link capacity so that congestion doesn't occur, multimedia traffic flows without delay or loss
  - low complexity of network mechanisms (use current "best effort" network)
  - high bandwidth costs
- challenges:
  - *network dimensioning:* how much bandwidth is "enough?"
  - *estimating network traffic demand:* needed to determine how much bandwidth is "enough" (for that much traffic)

Providing multiple classes of service

- thus far: making the best of best effort service
  - one-size fits all service model
- alternative: multiple classes of service
  - partition traffic into classes
  - network treats different classes of traffic differently (analogy: VIP service versus regular service)
- granularity: differential service among multiple classes, not among individual connections
- history: ToS bits

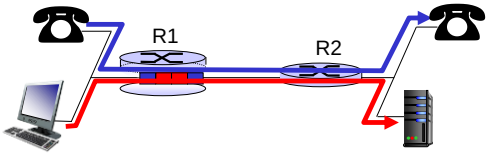


Multiple classes of service: scenario



Scenario 1: mixed HTTP and VoIP

- example: 1Mbps VoIP, HTTP share 1.5 Mbps link.
  - HTTP bursts can congest router, cause audio loss
  - want to give priority to audio over HTTP

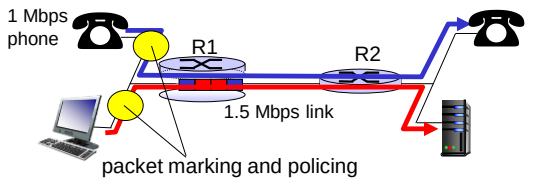


Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

Principles for QOS guarantees (more)

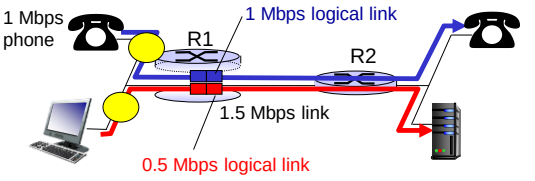
- what if applications misbehave (VoIP sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations
- *marking*, *policing* at network edge



Principle 2
provide protection (isolation) for one class from others

Principles for QOS guarantees (more)

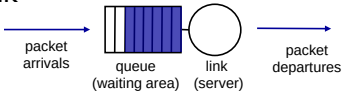
- allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flow doesn't use its allocation



Principle 3
while providing isolation, it is desirable to use resources as efficiently as possible

Scheduling and policing mechanisms

- *packet scheduling:* choose next queued packet to send on outgoing link



- previously covered in Chapter 4:
  - FCFS: first come first served
  - simply multi-class priority
  - round robin
  - weighted fair queueing (WFQ)

Policing mechanisms

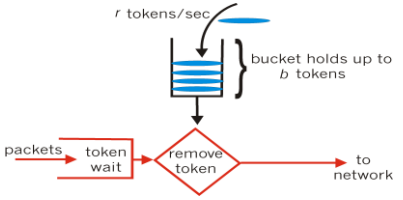
goal: limit traffic to not exceed declared parameters

Three common-used criteria:

- *(long term) average rate:* how many pkts can be sent per unit time (in the long run)
  - crucial question: what is the interval length: 100 packets per sec or 6000 packets per min have same average!
- *peak rate:* e.g., 6000 pkts per min (ppm) avg.; 1500 packets per sec peak rate
- *(max.) burst size:* max number of pkts sent consecutively (with no intervening idle)

Policing mechanisms: implementation

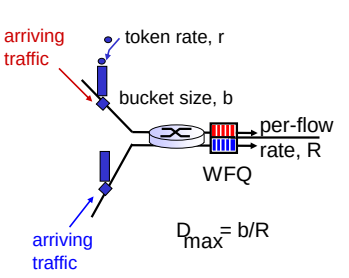
token bucket: limit input to specified *burst size* and *average rate*

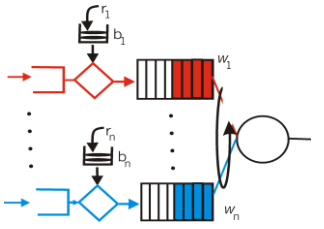


- bucket can hold b tokens
- tokens generated at rate *r token/sec* unless bucket full
- *over interval of length t: number of packets admitted less than or equal to (r t + b)*

Policing and QoS guarantees

- token bucket, WFQ combine to provide guaranteed upper bound on delay, i.e., *QoS guarantee!*



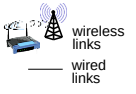


Computer Networks – Wrap Up

Sandhya Dwarkadas
Department of Computer Science
University of Rochester

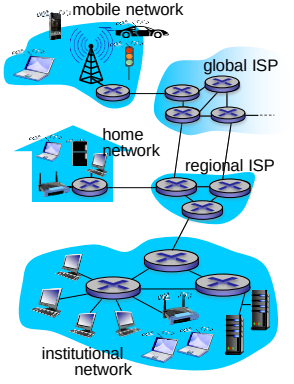
What is the Internet? A "nuts and bolts" view

- billions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*



- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate: *bandwidth*

- *packet switches:* forward packets (chunks of data)
  - *routers* and *switches*



Internet-connected devices

IP picture frame
http://www.ceiva.com/

Web-enabled toaster + weather forecaster

Tweet-a-watt: monitor energy use

Slingbox: watch, control cable TV remotely

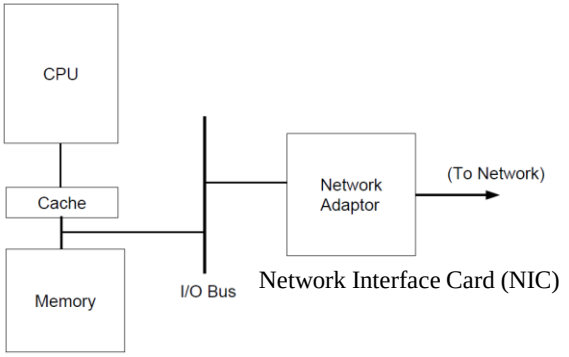
Internet refrigerator

sensorized, bed mattress

Internet phones

NETWORK INTERFACE:
HOW DOES THE HOST CONNECT TO THE COMMUNICATION LINK?

Abstract View of Machine



PROTOCOL/ABSTRACTION LAYERS

Computer Networks: Principles and Architecture

- Network Architecture:
  - Design and Implementation Guide

Principle of Abstraction – layering of protocols

A protocol provides a communication service to the next higher-level layer

- Service interface – e.g., send and receive
- Peer interface – form and meaning of messages exchanged between peers

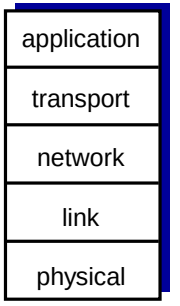
Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

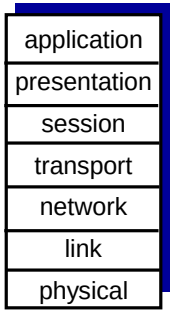
Internet protocol stack

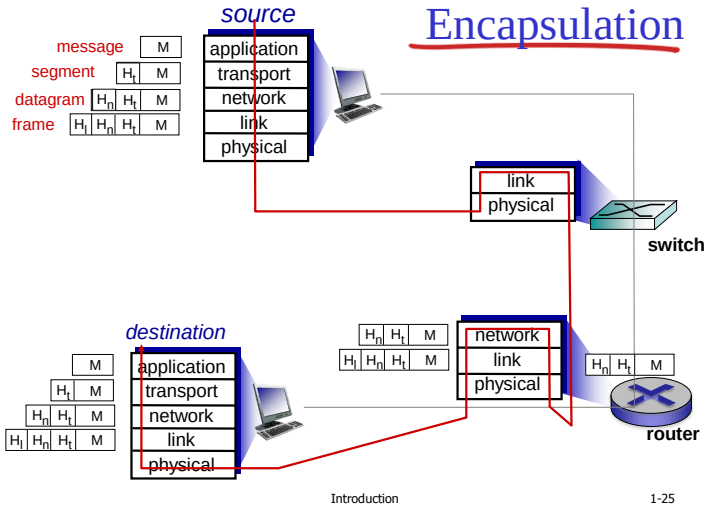
- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* **process-process data transfer**
  - **TCP, UDP**
- ***network:*** **routing** of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet,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



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 "missing" these layers!
  - these services, *if needed*, must be implemented in application
  - needed?





Course Overview

- Network architectures (protocols, layering, interfaces, encapsulation)
- Network technologies (e.g., Ethernet, FDDI, ATM, ISDN, wireless, HIPPI)
- Internetworking (addressing, routing, subnetting, autonomous systems)
- Resource allocation (fair queuing, virtual clocks, congestion avoidance)
- End-to-end issues (data representation, compression, authentication, encryption)
- Interprocess communication (datagrams, virtual circuits, request/reply, multicast, reliable broadcast, mobility)
- Multimedia networking
- Security in computer networks
- Wireless and mobile networks

Disclaimer

- Parts of the lecture slides are adapted from and copyrighted by James Kurose and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).

- What
- Why
- Trends
- architecture
- Issues