

# Network Layer:

## Overview;

## Distance Vector Protocols

Qiao Xiang

<https://qiaoxiang.me/courses/cnns-xmuf21/index.shtml>

11/30/2021

# Outline

---

- Admin and recap
- Network overview
- Network control-plane
  - Routing

# Admin

---

- Lab assignment 3 and 4 due today
- Lab assignment 6 to be posted this week
  - Written assignment

# Recap: BW Allocation Framework

$$\begin{aligned}
 & \max && \sum_{f \in F} U_f(x_f) \\
 & \text{subject to} && \sum_{\substack{f: f \text{ uses link } l}} x_f \leq c_l \text{ for any link } l \\
 & \text{over} && x \geq 0
 \end{aligned}$$

- Forward engineering: systematically design
  - objective function
  - distributed alg to achieve objective
- Science/reverse engineering: what do TCP/Reno, TCP/Vegas achieve?

Objective	Allocation ( $x_1, x_2, x_3$ )		
TCP/Reno	0.26	0.74	0.74
TCP/Vegas	1/3	2/3	2/3
Max throughput	0	1	1
Max-min	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Max sum $\log(x)$	1/3	2/3	2/3
Max sum of $-1/(\text{RTT}^2 x)$	0.26	0.74	0.74

# Recap: Derive Objective Function

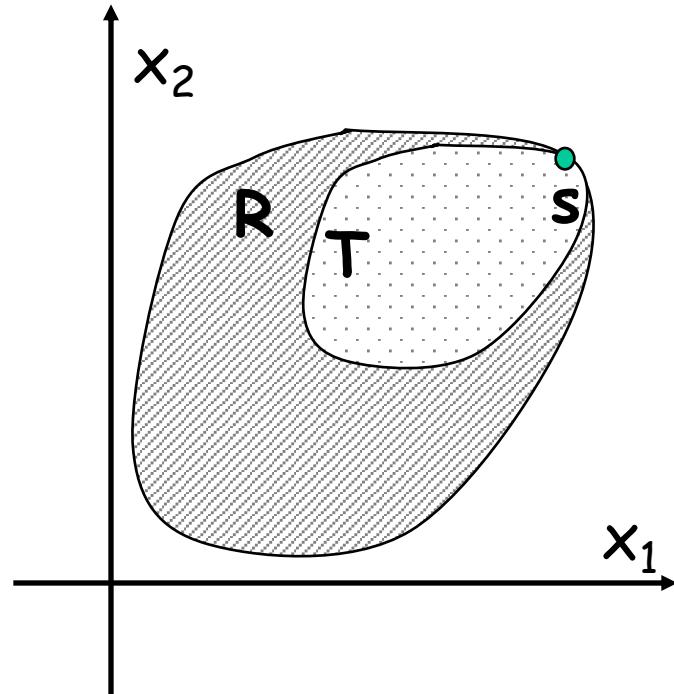
## □ NBS axioms

- Pareto optimality
- symmetry
- invariance of linear transformation
- independence of irrelevant alternatives

## □ NBS solution

- the rate allocation point is the feasible point which maximizes

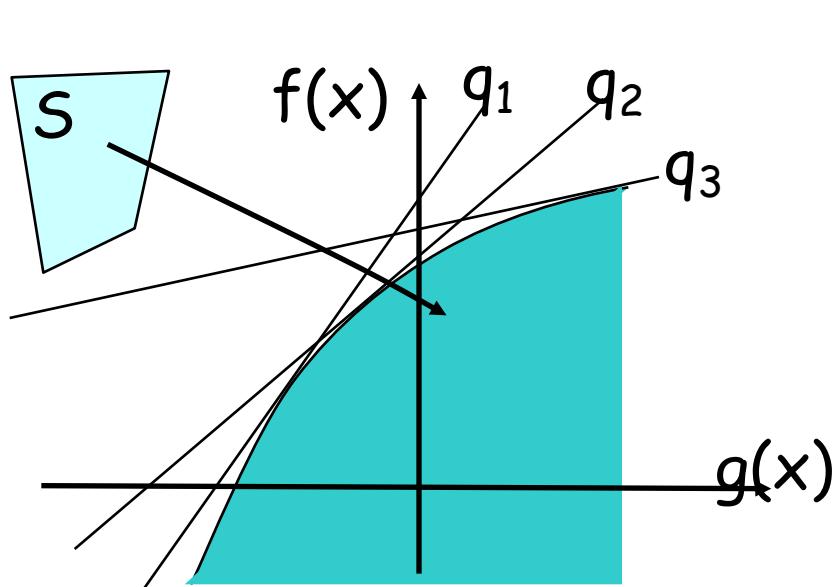
$$x_1 x_2 \cdots x_F$$



# Recap: Derive Alg (Strong Dual Theorem)

$$\begin{array}{ll} \max & f(x) \\ \text{subject to} & g(x) \leq 0 \\ \text{over} & x \in S \end{array}$$

$f(x)$  concave  
 $g(x)$  linear  
 $S$  is a convex set



$$D(q) = \max_{x \in S} (f(x) - qg(x))$$

$-D(q)$  is called the dual;  
 $q$  ( $\geq 0$ ) are called prices in economics

# Recap: Primal-Dual Decomposition of Network-Wide Resource Allocation

## □ SYSTEM(U):

$$\begin{array}{ll}\max & \sum_{f \in F} U_f(x_f) \\ \text{subject to} & \sum_{\substack{f: f \text{ uses link } l}} x_f \leq c_l \text{ for any link } l \\ \text{over} & x \geq 0\end{array}$$

## □ USER<sub>f</sub>:

$$\begin{array}{ll}\max_{x_f} & U_f(x_f) - x_f p_f \\ \text{over} & x_f \geq 0\end{array}$$

## □ NETWORK:

$$\min_{q \geq 0} \tilde{D}(q) = \sum_l q_l (c_l - \sum_{\substack{f: f \text{ uses link } l}} x_f)$$

## TCP/Reno Dynamics

$$\Delta x_f \propto U'_f(x_f) - p_f$$

$$\Delta x = \frac{RTT}{2} x^2 \left( \frac{2}{x^2 RTT^2} - p \right)$$

$$U'_f(x_f) - p_f$$

$$\Rightarrow U'_f(x_f) = \left( \frac{\sqrt{2}}{x_f RTT} \right)^2 \Rightarrow U_f(x_f) = -\frac{2}{RTT^2 x_f}$$

## TCP/Vegas Dynamics

$$\Delta x_f \propto U'_f(x_f) - p_f$$

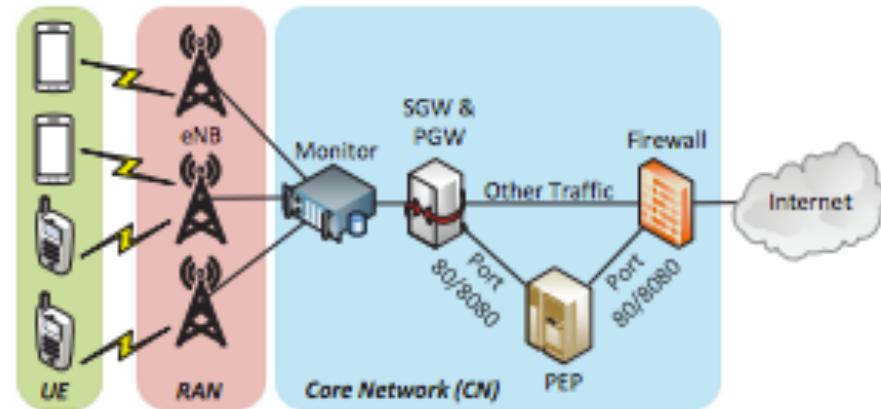
$$\Delta x = \frac{x}{RTT} \left( \frac{\alpha}{x} - (RTT - RTT_{\min}) \right)$$

$$U'_f(x_f) - p_f$$

$$\Rightarrow U'_f(x_f) = \frac{\alpha}{x} \quad \Rightarrow U_f(x_f) = \alpha \log(x_f)$$

# Summary

- ❑ Many aspects of TCP can be studied, for example
  - TCP under wireless (LTE)
  - Multipath TCP
  - TCP BBR
  - ...



# Outline

---

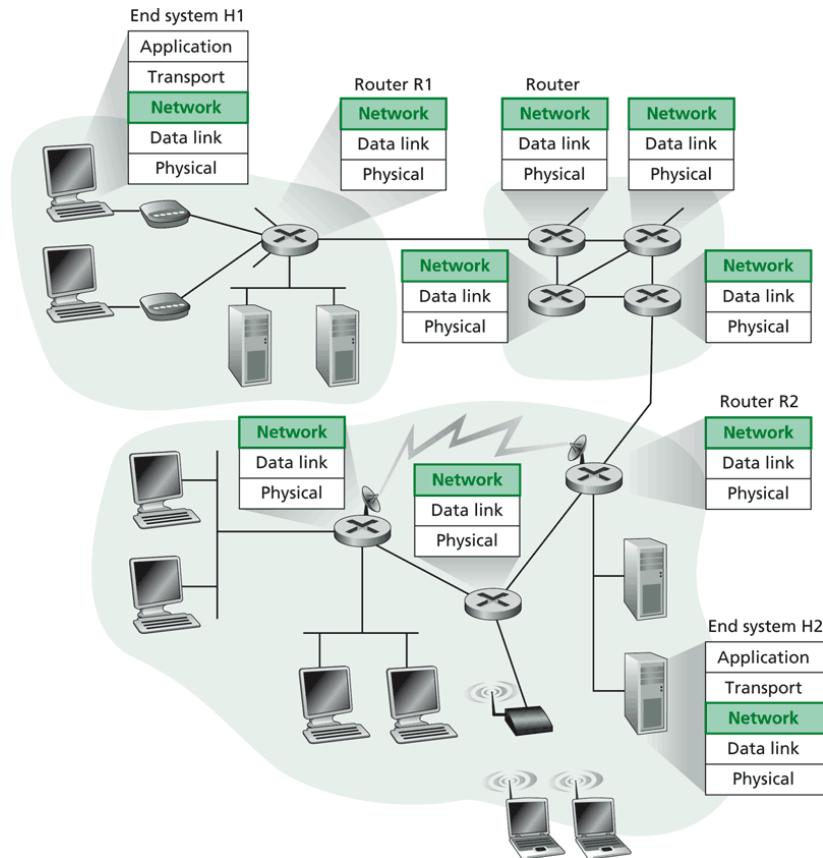
- Admin and recap
- *Network overview*

# Network Layer

- Transport packet from source to dest.
- Network layer in *every* host, router

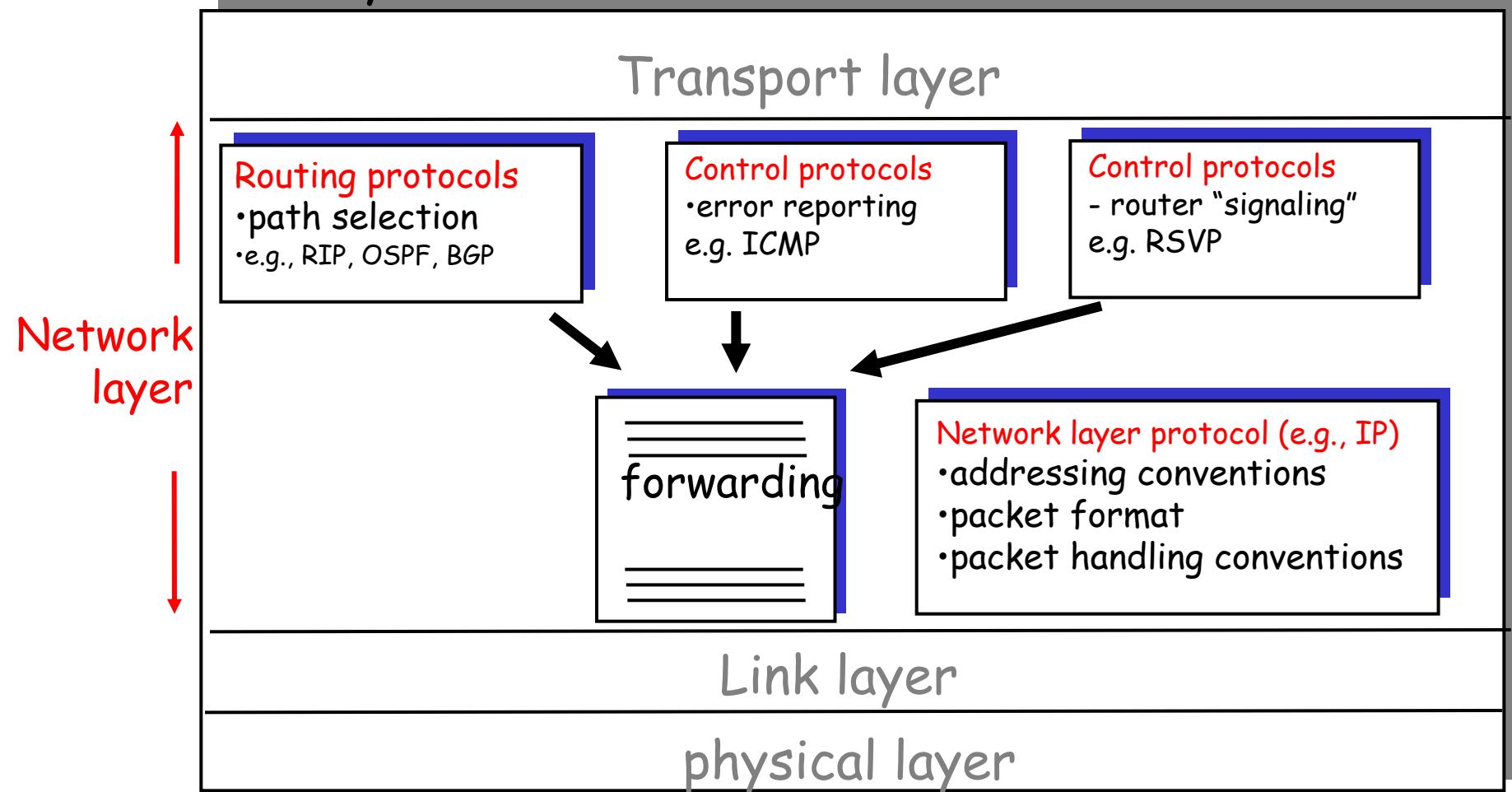
## Basic functions:

- inter-networking (e.g., fragmentation/assembly)
- **routing** (determine route(s) taken by packets of a flow), and **forwarding** (move the packets along the route(s))



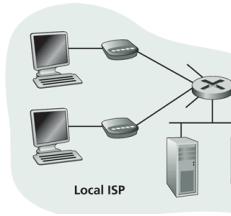
# Current Internet Network Layer

Network layer functions:

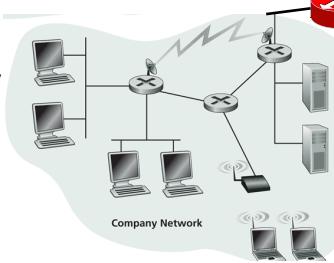


# Our Focus: Global Routing System

Residential networks, e.g., cable, fiber, DSL, wireless, cellular



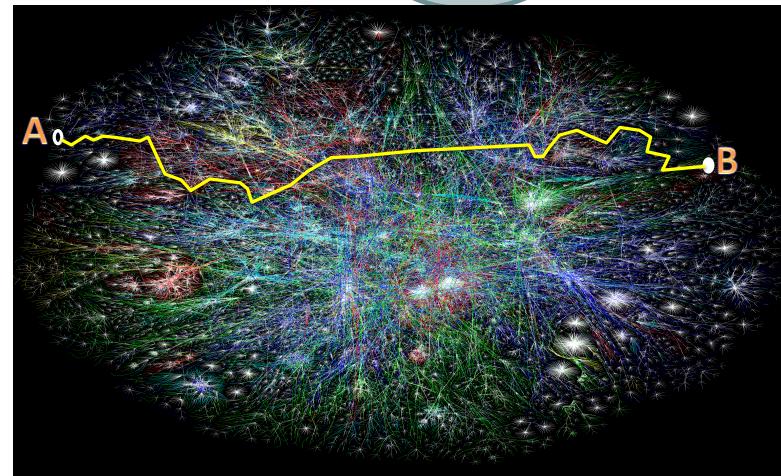
Campus networks, e.g., Ethernet, Wireless



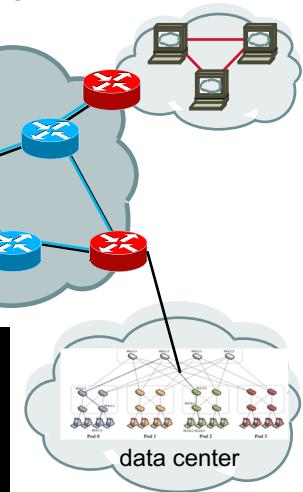
ISP

Backbone ISP

ISP



Discussion: What are the challenges/requirements of designing the global routing system?



# Global Routing Divide and Conquer: Routing with Autonomous Systems

- Global Internet routing is divided into intra-AS routing and inter-AS routing
  - Intra-AS routing (also called intradomain routing)
    - A protocol running inside an AS is called an Interior Gateway Protocol (IGP), each AS can choose its own protocol, such as RIP, E/IGRP, OSPF, IS-IS
  - Inter-AS routing (also called interdomain routing)
    - A protocol runs among autonomous systems is also called an Exterior Gateway Protocol (EGP)
    - The de facto EGP protocol is BGP

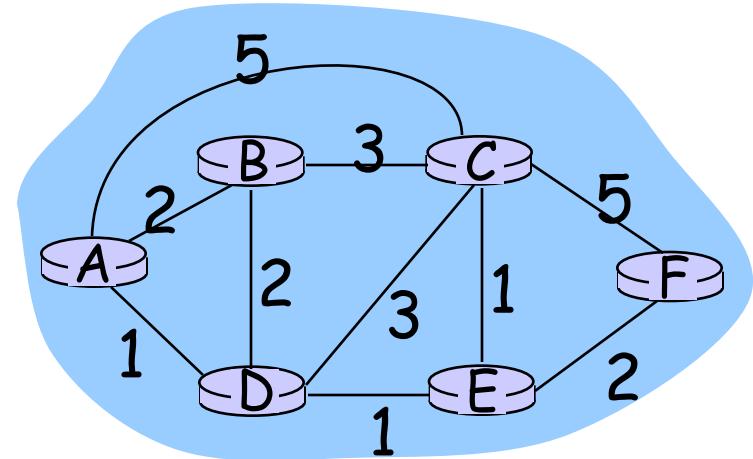
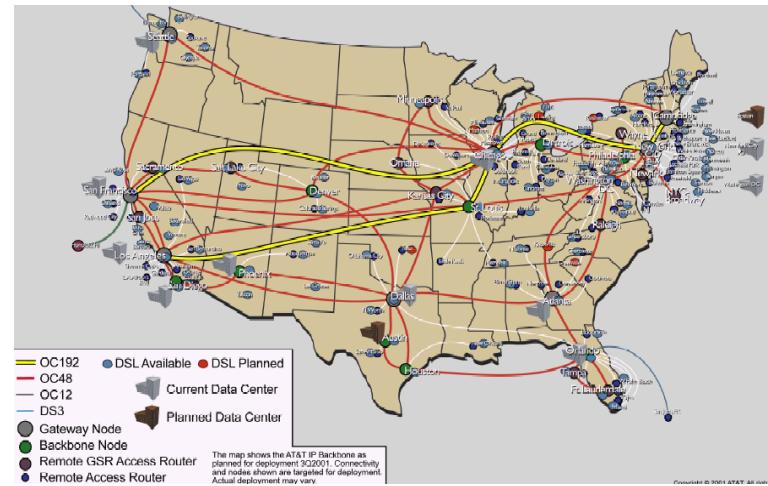
# Routing: Overview

## Routing

**Goal:** determine “good” paths (sequences of routers) thru networks from source to dest.

Graph abstraction for the routing problem:

- graph nodes are routers
- graph edges are physical links
  - links have properties: delay, capacity, \$ cost, **policy**



# Network Layer: Complexity

## Factors/Objectives

- For network providers
  - efficiency of routes
  - policy control on routes
  - scalability
- For users: quality of services
  - guaranteed bandwidth?
  - preservation of inter-packet timing (no jitter)?
  - loss-free delivery?
  - in-order delivery?
- Users and network may interact

# Routing Design Space

- Robustness
- Optimality
- Simplicity

- Routing has a large design space
  - who decides routing?
    - source routing: end hosts make decision
    - network routing: networks make decision
  - how many paths from source s to destination d?
    - multi-path routing
    - single path routing
  - what does routing compute?
    - network cost minimization
    - QoS aware
  - will routing adapt to network traffic demand?
    - adaptive routing
    - static routing
  - ...

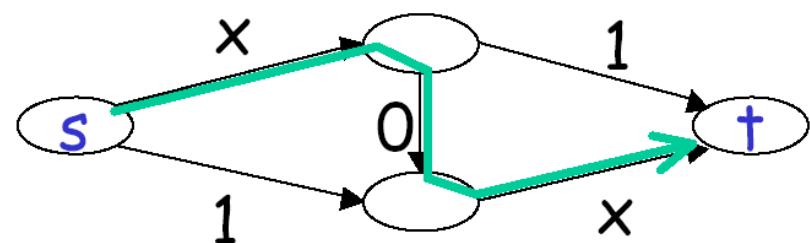
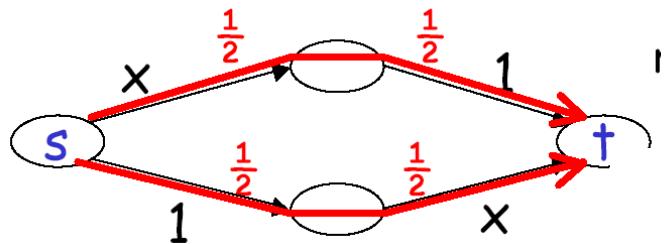
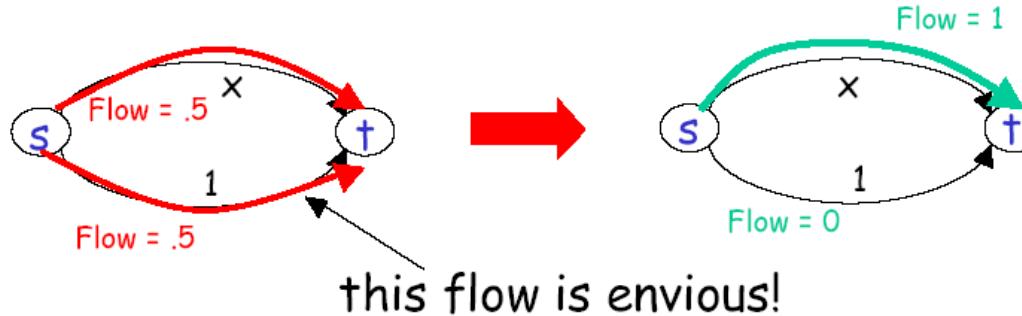
# Routing Design Space: User-based, Multipath, Adaptive

- Robustness
- Optimality
- Simplicity

- Routing has a large design space
  - who decides routing?
    - source routing: end hosts make decision
    - network routing: networks make decision
  - how many paths from source s to destination d?
    - multi-path routing
    - single path routing
  - what does routing compute?
    - network cost minimization
    - QoS aware
  - will routing adapt to network traffic demand?
    - adaptive routing
    - static routing
  - ...

# User Optimal, Multipath, Adaptive

- User optimal: users pick the shortest routes (selfish routing)



Braess's paradox

# Price of Anarchy

For a network with **linear** latency functions

→

total latency of user (selfish) routing for given traffic demand

$\leq 4/3$

total latency of network optimal routing for the traffic demand

# Price of Anarchy

- For any network with continuous, non-decreasing latency functions →

total latency of user (selfish) routing  
for given traffic demand

$\leq$

total latency of network optimal routing  
for **twice** traffic demand

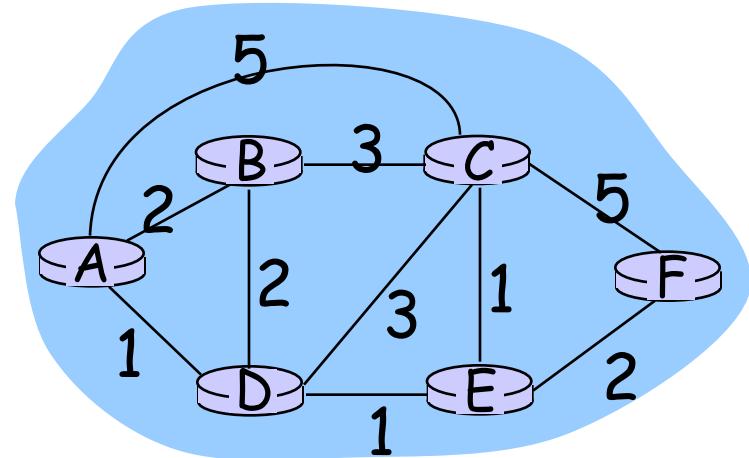
# Routing Design Space: Internet

- Robustness
- Optimality
- Simplicity

- ❑ Routing has a large design space
  - who decides routing?
    - source routing: end hosts make decision
    - network routing: networks make decision
      - (applications such as overlay and p2p are trying to bypass it)
  - what does routing compute?
    - network cost minimization (shortest path)
      - QoS aware
  - how many paths from source s to destination d?
    - multi-path routing
    - single path routing (with small amount of multipath)
  - will routing adapt to network traffic demand?
    - adaptive routing
    - static routing (mostly static; adjust in larger timescale)
  - ...

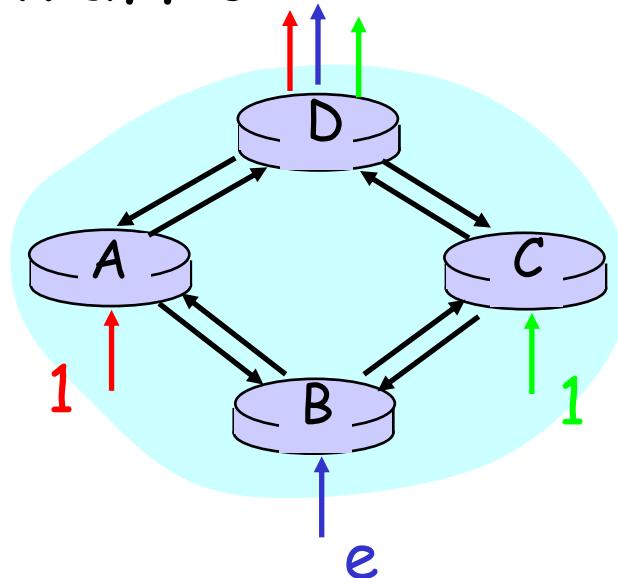
# Basic Formulation

- ❑ Assign link weights
- ❑ Compute shortest path

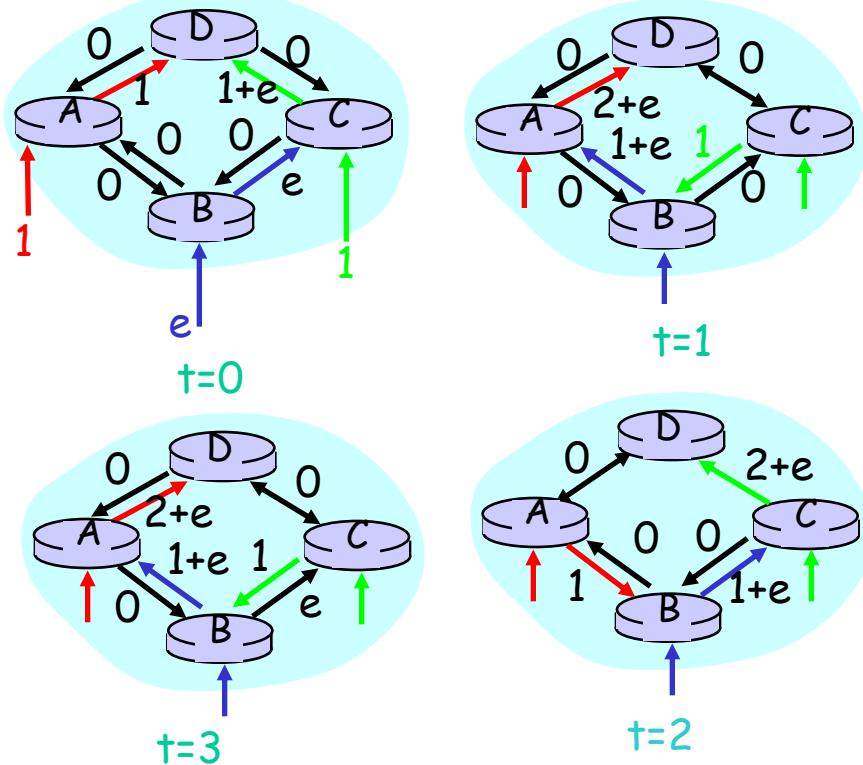


# Assigning Link Weight: Dynamic Link Costs

- Assign link costs to reflect current traffic



Link costs reflect current traffic intensity



Solution: Link costs are a combination of current traffic intensity (dynamic) and topology (static). To improve stability, the static topology part should be large. Thus less sensitive to traffic; thus non-adaptive.

## Example: Cisco Proprietary Recommendation on Link Cost

### □ Link metric:

- metric =  $[K1 * \text{bandwidth}^{-1} + (K2 * \text{bandwidth}^{-1}) / (256 - \text{load}) + K3 * \text{delay}] * [K5 / (\text{reliability} + K4)] * 256$

By default,  $k1=k3=1$  and  $k2=k4=k5=0$ . The default composite metric for EIGRP, adjusted for scaling factors, is as follows:

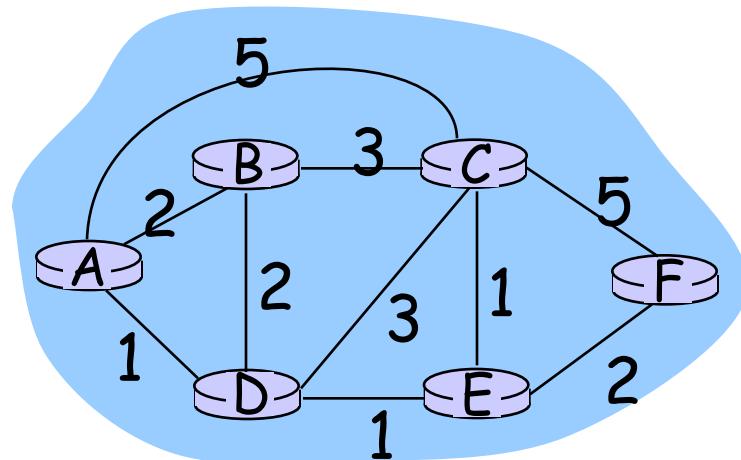
$$\text{EIGRP}_{\text{metric}} = 256 \times \{ [10^7 / \text{BW}_{\min}] + [\text{sum\_of\_delays}] \}$$

$\text{BW}_{\min}$  is in kbps and the sum of delays are in 10s of microseconds.

EIGRP : Enhanced Interior Gateway Routing Protocol

# Example: EIGRP Link Cost

- The bandwidth and delay for an Ethernet interface are 10 Mbps and 1 ms, respectively.
- The calculated EIGRP BW metric is as follows:
  - $256 \times 10^7 / \text{BW} = 256 \times 10^7 / 10,000$   
=  $256 \times 1000$   
= 256,000



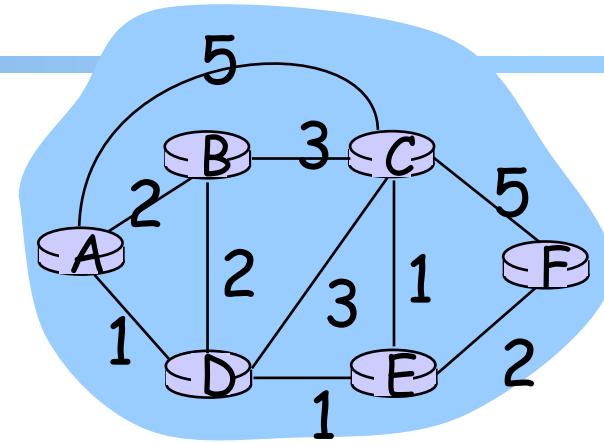
# Outline

---

- Admin and recap
- Network overview
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - *Distributed distance vector protocols*

# Distance Vector Routing

- Setting: static (positive) costs assigned to network links
  - The static link costs may be adjusted in a longer time scale: this is called traffic engineering
- Goal: distributed computing to compute the shortest path from a source to a destination
  - Based on the Bellman-Ford algorithm (BFA)
  - Conceptually, runs for each destination separately
- Look ahead
  - Although few (e.g., RIP) use basic distance vector, it is a foundation for many other protocols
  - We also use the study to acquire another basic set of techniques to understand distributed protocols



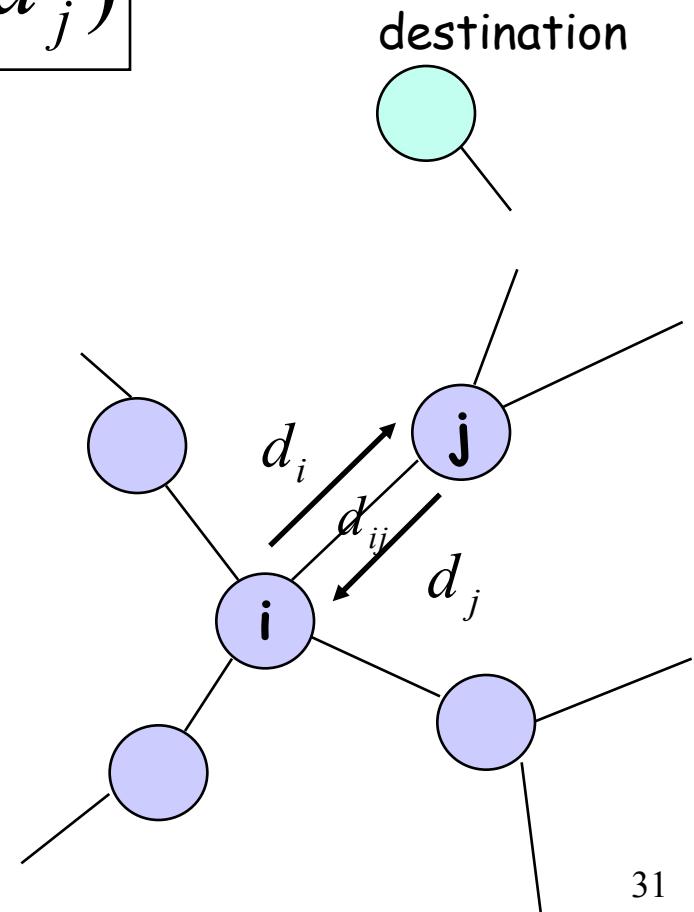
# Distance Vector Routing: Basic Idea

- At node  $i$ , the basic update rule

$$d_i = \min_{j \in N(i)} (d_{ij} + d_j)$$

where

- $d_i$  denotes the distance estimation from  $i$  to the destination,
- $N(i)$  is set of neighbors of node  $i$ , and
- $d_{ij}$  is the distance of the direct link from  $i$  to  $j$



# Outline

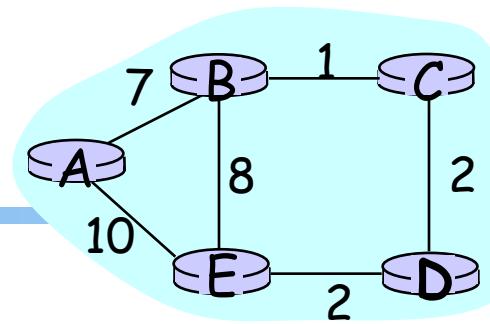
---

- Admin and recap
- Network overview
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - *distributed distance vector protocols*
      - *synchronous Bellman-Ford (SBF)*

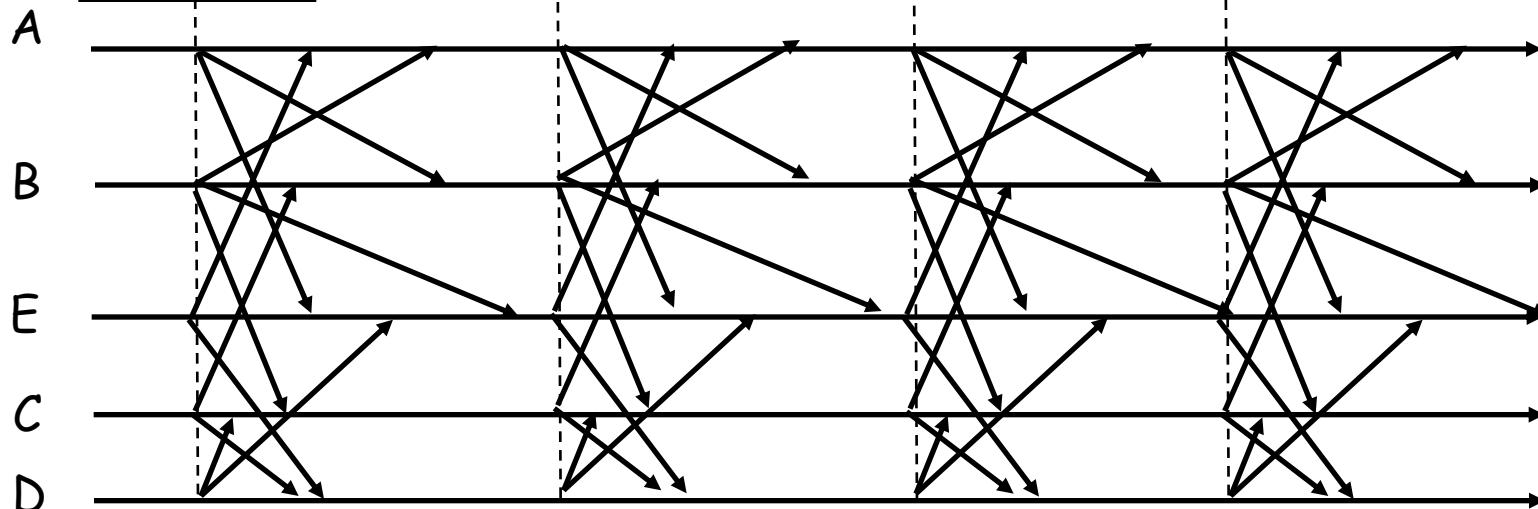
## Synchronous Bellman-Ford (SBF)

### □ Nodes update in rounds:

- there is a global clock;
- at the beginning of each round, each node sends its estimate to all of its neighbors;
- at the end of the round, updates its estimation



$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h))$$

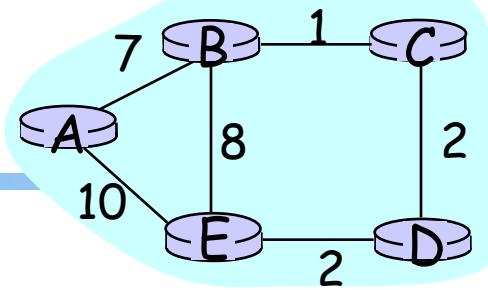


# Outline

---

- Admin and recap
- Network overview
- Network control-plane path
  - Routing
    - Link weights assignment
    - Routing computation
      - *distributed distance vector protocols*
      - *synchronous Bellman-Ford (SBF)*
      - SBF/ $\infty$

# SBF/∞



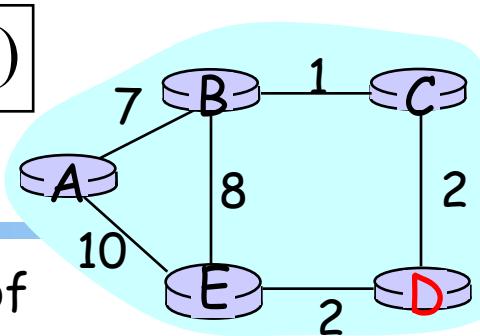
- Initialization (time 0):

$$d_i(0) = \begin{cases} 0 & i = \text{dest} \\ \infty & \text{otherwise} \end{cases}$$

$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h))$$

## Example

Consider D as destination;  $d(t)$  is a vector consisting of estimation of each node at round  $t$



	A	B	C	E	D
$d(0)$	$\infty$	$\infty$	$\infty$	$\infty$	0
$d(1)$	$\infty$	$\infty$	2	2	0
$d(2)$	12	3	2	2	0
$d(3)$	10	3	2	2	0
$d(4)$	10	3	2	2	0

Observation:  $d(0) \geq d(1) \geq d(2) \geq d(3) \geq d(4) = d^*$

$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h))$$

## A Nice Property of SBF: Monotonicity

- Consider two configurations  $d(t)$  and  $d'(t)$
- If  $d(t) \geq d'(t)$ 
  - i.e., each node has a higher estimate in one scenario ( $d$ ) than in another scenario ( $d'$ ),
- then  $d(t+1) \geq d'(t+1)$ 
  - i.e., each node has a higher estimate in  $d$  than in  $d'$  after one round of synchronous update.

$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h))$$

## Correctness of SBF/ $\infty$

- **Claim:**  $d_i(h)$  is the length  $L_i(h)$  of a shortest path from  $i$  to the destination using  $\leq h$  hops
  - base case:  $h = 0$  is trivially true
  - assume true for  $\leq h$ ,  
i.e.,  $L_i(h) = d_i(h)$  ,  $L_i(h-1) = d_i(h-1)$  , ...

$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h))$$

## Correctness of SBF/ $\infty$

□ consider  $\leq h+1$  hops:

$$L_i(h+1) = \min(L_i(h), \min_{j \in N(i)} (d_{ij} + L_j(h)))$$

$$= \min(d_i(h), \min_{j \in N(i)} (d_{ij} + d_j(h)))$$

$$= \min(d_i(h), d_i(h+1))$$

since  $d_i(h) \leq d_i(h-1)$

$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h)) \leq \min_{j \in N(i)} (d_{ij} + d_j(h-1)) = d_i(h)$$

$$L_i(h+1) = d_i(h+1)$$

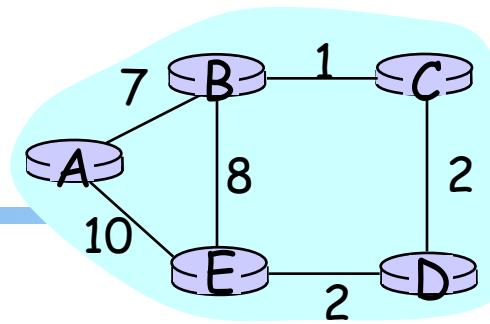
# Outline

---

- Admin and recap
- Network overview
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - *Distributed distance vector protocols*
      - *synchronous Bellman-Ford (SBF)*
        - SBF/ $\infty$
        - SBF/-1 SBF/ $\infty$

## SBF at another

Initial Configuration: SBF/-1



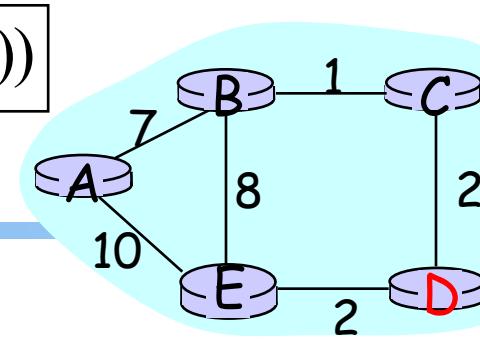
- Initialization (time 0):

$$d_i(0) = \begin{cases} 0 & i = \text{dest} \\ -1 & \text{otherwise} \end{cases}$$

$$d_i(h+1) = \min_{j \in N(i)} (d_{ij} + d_j(h))$$

## Example

Consider D as destination



	A	B	C	E	D
d(0)	-1	-1	-1	-1	0
d(1)	6	0	0	2	0
d(2)	7	1	1	2	0
d(3)	8	2	2	2	0
d(4)	9	3	3	2	0
d(5)	10	3	3	2	0
d(6)	10	3	3	2	0

Observation:  $d(0) \leq d(1) \leq d(2) \leq d(3) \leq d(4) \leq d(5) = d(6) = d^*$

## Correctness of SBF/-1

---

- SBF/-1 converges due to monotonicity
  
- Remaining question:
  - Can we guarantee that SBF/-1 converges to shortest path?