Herding Packets

How to get Packets to Go Where You Expect

Judith Samson
UCSC SOE Computer Engineering Dept.
jtsamson@soe.ucsc.edu
Advisor: Brad Smith

Herding Packets…the Rules

- Modern TCP/IP networks break communications down into millions of packets
- All packets are forwarded through a network individually
- No packet knows where it came from, only where it is going
- **Destination-based, hop-by-hop routing**

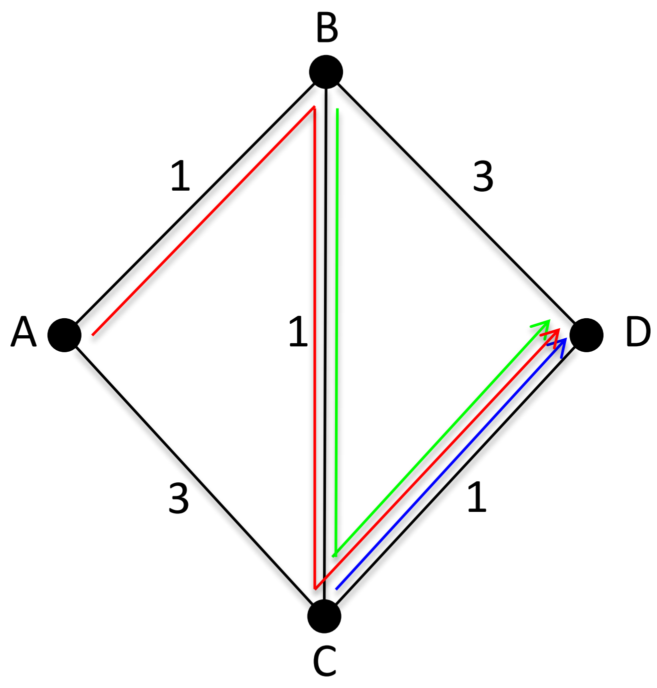
Bike Relay Race Across America

- Thousands of cyclists---need roads that have large capacity
- Shortest routes
- Hop-by-hop—each cyclist only knows best path to destination, not paths earlier teammates took.

Shortest Path is Easy…

…Just add up the distance of each leg of the trip

Shortest



Forwarding paths

- C → D: C – D = 1
- B → D: B – C – D = 2
- A → D: A – B – C – D = 3

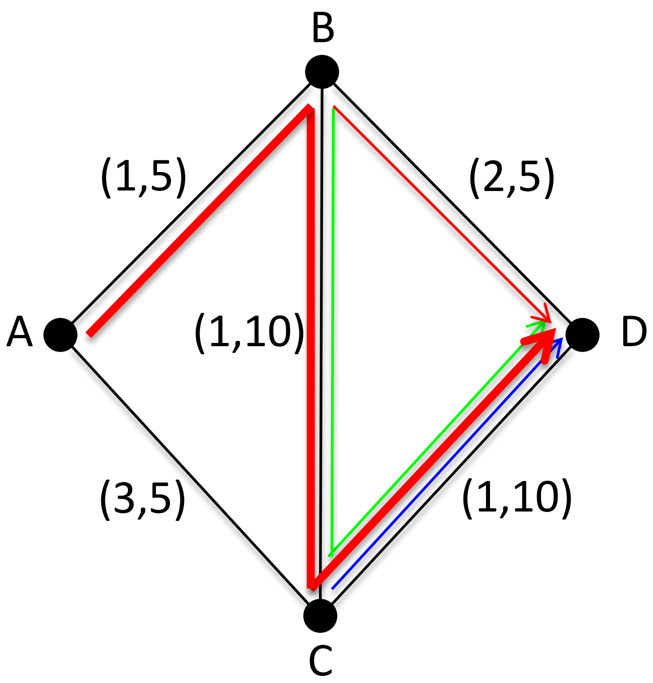
Path Properties

- Best path can be found
- Loop-free
- Each sub-path is also a best path

Shortest with Greatest Capacity, not so bad…

…Constrained by road with least **capacity** (the narrowest or most crowded leg of the trip)

Widest-Shortest



Metrics: (delay, bandwidth)

- "+": $(d_1,b_1) + (d_2,b_2) = (d_1+d_2, Min(b_1,b_2))$
- "≤": $(d_1,b_1) \leq (d_2,b_2) = (d_1 < d_2)$ *or* $((d_1 = d_2)$ *and* $(b_1 \geq b_2))$

Forwarding paths

- C → D: C – D = (1,10)
- B → D: B – C – D = (2,10)
- A → D: **A – B – C – D = (3,5)**
  A – B – D = (3,5)

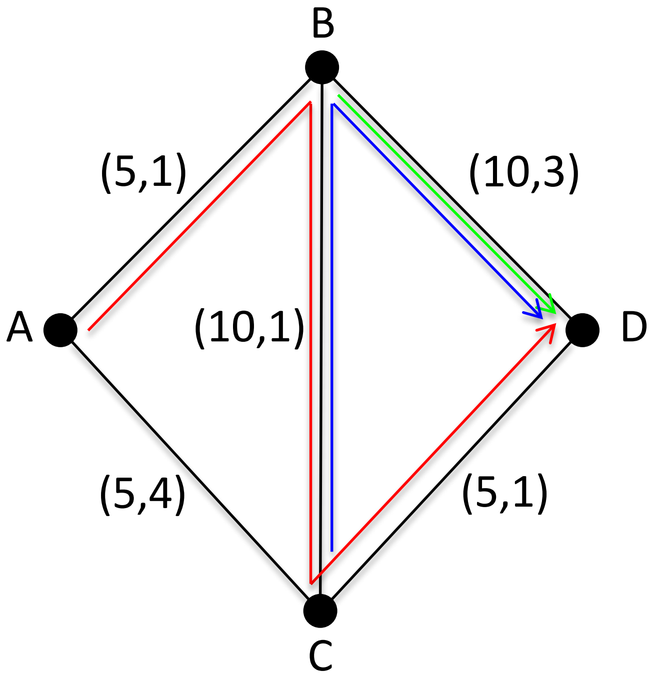
Path Properties

- Best path can be found
- Loop-free

Multiple Metrics Makes Things Harder

- Finding the greatest bandwidth, shortest path, **hard**
- Finding just the greatest bandwidth results in loops
- Finding the least cost per distance--**impossible**

Shortest-Widest



Metrics: (bandwidth, delay)

- "+": $(b_1,d_1) + (b_2,d_2) = (Min(b_1,b_2), d_1+d_2)$
- "≤": $(b_1,d_1) \leq (b_2,d_2) = (b_1 > b_2)$ *or* $((b_1 = b_2)$ *and* $(d_1 \leq d_2))$

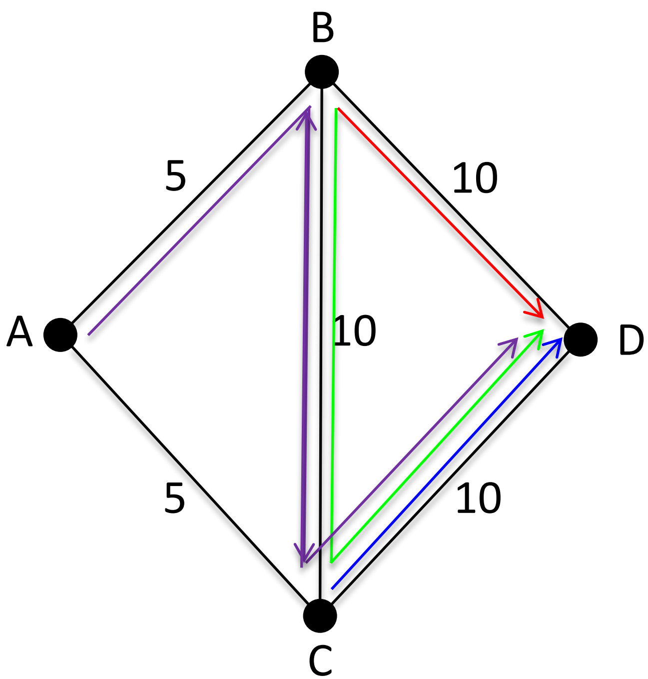
Forwarding paths

- C → D: C – B – D = (10,4)
- B → D: B – D = (10,3)
- A → D: A – B – C – D = (5,3)

Path Properties

- **Best path cannot be found**
- Loop-free

Widest



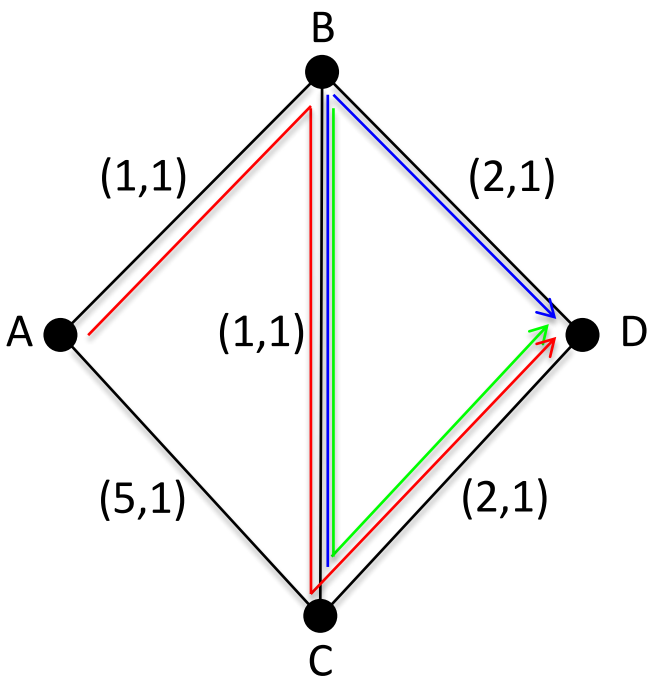
Forwarding paths

- C → D: C – D = 10
- B → D: B – C – D = 10
- B → D: B – D = 10
- A → D: B – C – B – … – D = 10

Path Properties

- Best path can be found
- **Not Loop-free**

Cost per Distance



Metrics: (cost, distance)

- "+": $(c_1,d_1) + (c_2,d_2) = (c_1+c_2, d_1+d_2)$
- "≤": $(c_1,d_1) \leq (c_2,d_2) = (c_1/d_1) \leq (c_2/d_2)$

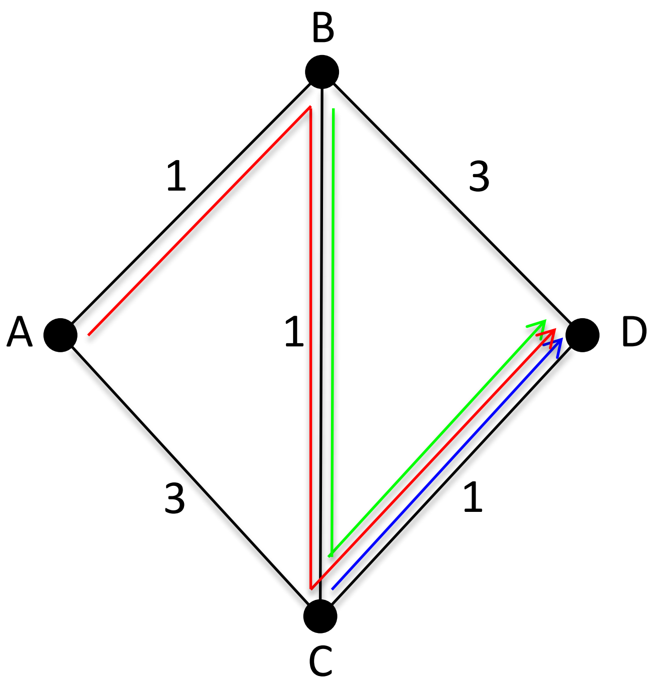
Forwarding paths

- C → D: C – B – D = 3/2
- B → D: B – C – D = 3/2
- A → D: A – B – C – D = 4/3

Path Properties

- $\frac{4}{3} < \frac{3}{2}$ → adding a hop produces a shorter path!
- **Best path cannot be found**
- **Not Loop-free**

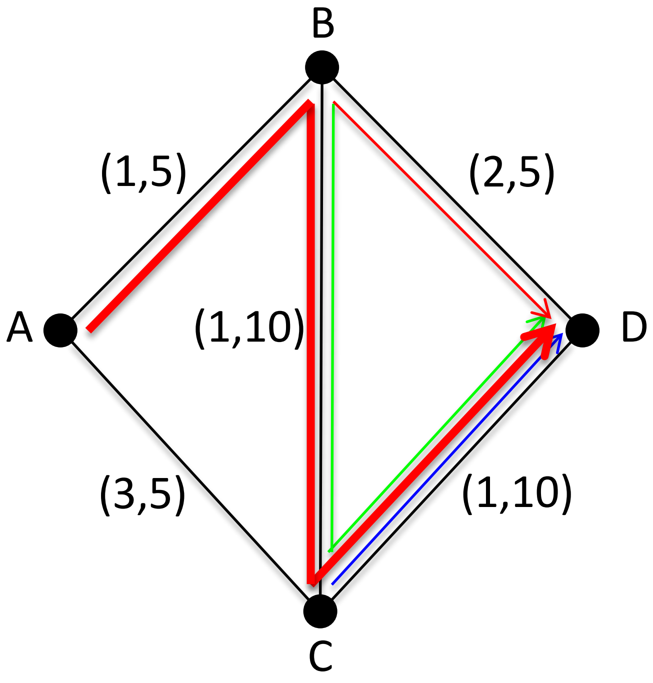
Properties: Shortest



- Strictly Monotonic → Best path can be found
- Strictly Bounded → Loop-free
- Optimal

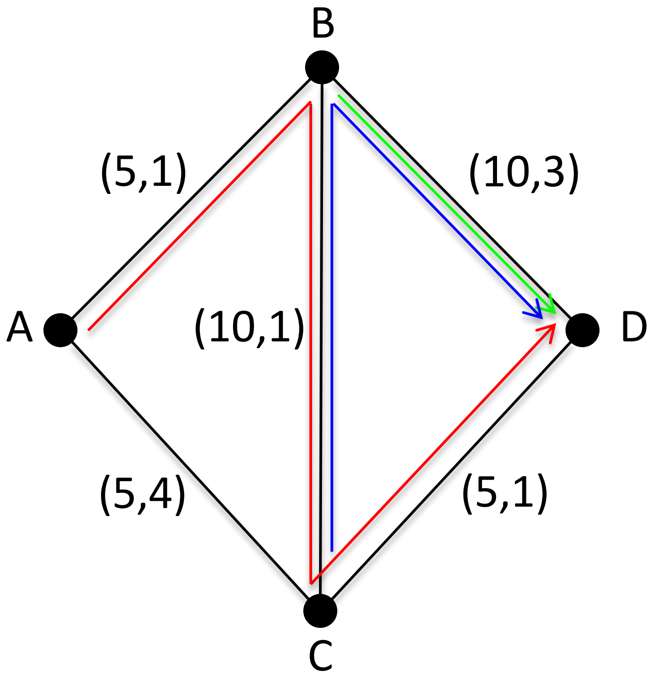
Optimal paths are paths whose sub-paths are also best paths

Properties: Widest-Shortest



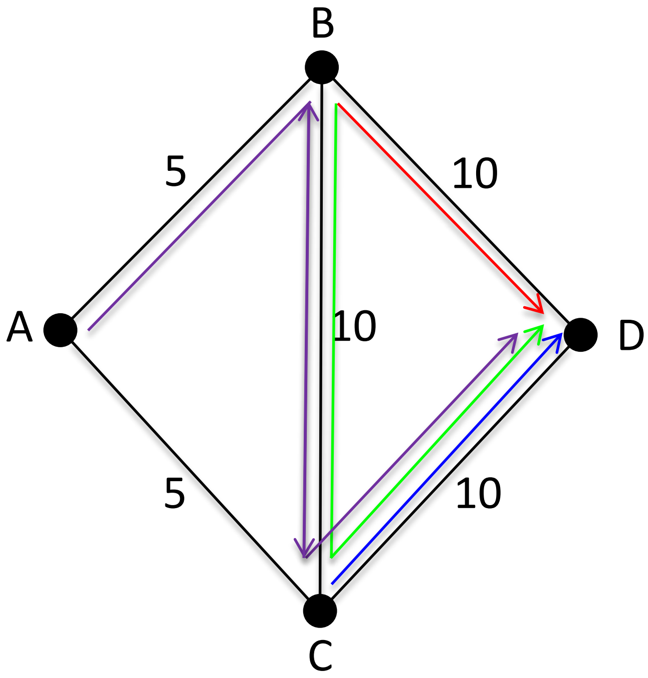
- Monotonic → Best path can be found
- Strictly Bounded → Loop-free

Properties: Shortest-Widest



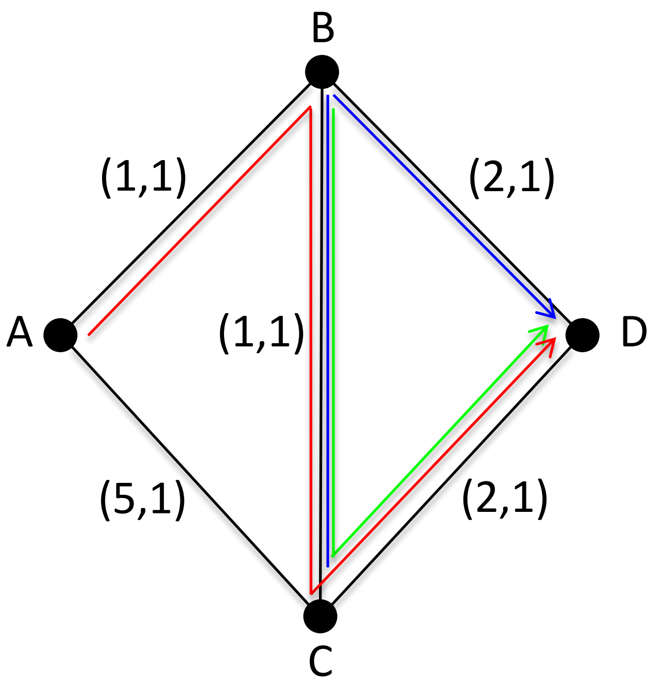
- Not Monotonic→ Best path **cannot** be found
- Strictly Bounded → Loop-free

Properties: Widest



- Monotonic → Best path can be found
- Bounded (but not strictly) → **Not** Loop-free

Properties: "Slope"



- Not Monotonic → Best path **cannot** be found
- Not Bounded → **Not** Loop-free

Properties of Path Algebras

- Goal
  - Given a set of metrics, determine if best, loop-free paths can be found
- Forwarding Paths
  - Each node has routing table listing best path from it to all other nodes
  - Packets have no information about source or previous paths—only their destination

UC SANTA CRUZ

Santa Cruz Baskin School of Engineering
Engineering

Results

For paths a, b and c:

- Bounded: $a \leq a + b$
- Monotonic $a \leq b \rightarrow a + c \leq b + c$
- If a path algebra is *monotonic*, then best forwarding paths will be found.
- If a path algebra is *strictly bounded*, then it is loop-free.
- If a path algebra is *strictly monotonic* and *strictly bounded*, then optimal forwarding paths will be found

Next Step

Extend to multipath environment:

Find set of best paths, rather than the single best path through a network

Questions??