

Vorlesung Telematik (Computer Networks)
WS2004/05

Routing

Telematics group
University of Göttingen, Germany

Telematics group
University of Göttingen, Germany

Routing

- Routing principles
 - Link state routing
 - Distance vector routing
- Hierarchical routing
- Homework
- Self-learning: Internet routing protocols

Credits:

- James Kurose & Keith Ross: Computer Networking(2nd Ed.), Addison-Wesley, 2002

WS 2004/05, fu@informatik.cs.uni-goettingen.de

Telematics group
University of Göttingen, Germany

Routing

Routing protocol
Goal: determine "good" path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
 - link cost: delay, \$ cost, or congestion level

- "good" path:
 - typically means minimum cost path
 - other def's possible

WS 2004/05, fu@informatik.cs.uni-goettingen.de

Telematics group
University of Göttingen, Germany

Routing Algorithm classification

<p>Global or decentralized information?</p> <p>Global:</p> <ul style="list-style-type: none"> • all routers have complete topology, link cost info • "link state" algorithms <p>Decentralized:</p> <ul style="list-style-type: none"> • router knows physically-connected neighbors, link costs to neighbors • iterative process of computation, exchange of info with neighbors • "distance vector" algorithms 	<p>Static or dynamic?</p> <p>Static:</p> <ul style="list-style-type: none"> • routes change slowly over time <p>Dynamic:</p> <ul style="list-style-type: none"> • routes change more quickly <ul style="list-style-type: none"> – periodic update – in response to link cost changes
---	--

WS 2004/05, fu@informatik.cs.uni-goettingen.de

A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
 - accomplished via "link state broadcast"
 - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
 - gives routing table for that node
- iterative: after k iterations, know least cost path to k

Notation:

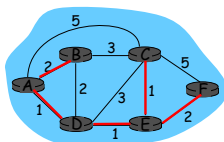
- $c(i,j)$: link cost from node i to j. cost infinite if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. v
- $p(v)$: predecessor node along path from source to v, that is next v
- N : set of nodes whose least cost path definitively known

Dijkstra's Algorithm

- Initialization:**
- $N = \{A\}$
- for all nodes v
- if v adjacent to A
- then $D(v) = c(A,v)$
- else $D(v) = \text{infinity}$
-
- Loop**
- find w not in N such that $D(w)$ is a minimum
- add w to N
- update $D(v)$ for all v adjacent to w and not in N:
- $D(v) = \min(D(v), D(w) + c(w,v))$
- /* new cost to v is either old cost to v or known
- shortest path cost to w plus cost from w to v */
- until all nodes in N**

Dijkstra's algorithm: example

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	infinity	infinity
→ 1	AD	2,A	4,D		2,D	infinity
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



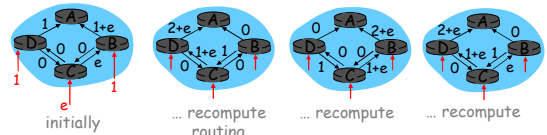
Dijkstra's algorithm, discussion

Algorithm complexity: n nodes

- each iteration: need to check all nodes, w, not in N
- $n*(n+1)/2$ comparisons: $O(n^2)$
- more efficient implementations possible: $O(n*\log n)$

Oscillations possible:

- e.g., link cost = amount of carried traffic



Telematics group
University of Göttingen, Germany

Distance Vector Routing Algorithm

iterative:

- continues until no nodes exchange info.
- self-terminating*: no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

distributed:

- each node communicates *only* with directly-attached neighbors

Distance Table data structure

- each node has its own
 - row for each possible destination
 - column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

$$D^X(Y,Z) = \begin{aligned} & \text{distance from X to} \\ & \text{Y, via Z as next hop} \\ & = c(X,Z) + \min_w \{D^Z(Y,w)\} \end{aligned}$$

WS 2004/05, fu@informatik.cs.uni-goettingen.de 9

Telematics group
University of Göttingen, Germany

Distance Table: example

Distance Table:

$D^E()$	cost to destination via		
	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Calculations:

$$D^E(C,D) = c(E,D) + \min_w \{D^D(C,w)\} = 2+2 = 4$$

$$D^E(A,D) = c(E,D) + \min_w \{D^D(A,w)\} = 2+3 = 5 \text{ loop!}$$

$$D^E(A,B) = c(E,B) + \min_w \{D^B(A,w)\} = 8+6 = 14 \text{ loop!}$$

WS 2004/05, fu@informatik.cs.uni-goettingen.de 10

Telematics group
University of Göttingen, Germany

Distance table gives routing table

$D^E()$	cost to destination via			Outgoing link to use, cost	
	A	B	D		
A	1	14	5	A	A,1
B	7	8	5	B	D,5
C	6	9	4	C	D,4
D	4	11	2	D	D,4

Distance table \longrightarrow Routing table

WS 2004/05, fu@informatik.cs.uni-goettingen.de 11

Telematics group
University of Göttingen, Germany

Distance Vector Routing: overview

Iterative, asynchronous: each local iteration caused by:

- local link cost change
- message from neighbor: its least cost path change from neighbor

Distributed:

- each node notifies neighbors *only* when its least cost path to any destination changes
 - neighbors then notify their neighbors if necessary

wait for (change in local link cost of msg from neighbor)

\downarrow

recompute distance table

\downarrow

if least cost path to any dest has changed, *notify* neighbors

WS 2004/05, fu@informatik.cs.uni-goettingen.de 12

Telematics group
University of Göttingen, Germany

Distance Vector Algorithm:

At all nodes, X:

- 1 Initialization:
- 2 for all adjacent nodes v:
- 3 $D^X(*,v) = \text{infinity}$ /* the * operator means "for all rows" */
- 4 $D^X(v,v) = c(X,v)$
- 5 for all destinations, y
- 6 send $\min_w D^X(y,w)$ to each neighbor /* w over all X's neighbors */

WS 2004/05, fu@informatik.cs.uni-goettingen.de 13

Telematics group
University of Göttingen, Germany

Distance Vector Algorithm (cont.):

- 8 loop
- 9 wait (until I see a link cost change to neighbor V
or until I receive update from neighbor V)
- 11
- 12 if $(c(X,V)$ changes by d)
- 13 /* change cost to all dest's via neighbor v by d */
- 14 /* note: d could be positive or negative */
- 15 for all destinations y: $D^X(y,V) = D^X(y,V) + d$
- 16
- 17 else if (update received from V wrt destination Y)
- 18 /* shortest path from V to some Y has changed */
- 19 /* V has sent a new value for its $\min_w DV(Y,w)$ */
- 20 /* call this received new value is "newval" */
- 21 for the single destination y: $D^X(Y,V) = c(X,V) + \text{newval}$
- 22
- 23 if we have a new $\min_w D^X(Y,w)$ for any destination Y
- 24 send new value of $\min_w D^X(Y,w)$ to all neighbors
- 25
- 26 forever

WS 2004/05, fu@informatik.cs.uni-goettingen.de 14

Telematics group
University of Göttingen, Germany

Distance Vector Algorithm: example

WS 2004/05, fu@informatik.cs.uni-goettingen.de 15

Telematics group
University of Göttingen, Germany

Distance Vector Algorithm: example

$$D^X(Y,Z) = c(X,Z) + \min_w \{D^Z(Y,w)\}$$

$$= 7 + 1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w \{D^Y(Z,w)\}$$

$$= 2 + 1 = 3$$

WS 2004/05, fu@informatik.cs.uni-goettingen.de 16

Telematics group
University of Göttingen, Germany

Distance Vector: link cost changes

Link cost changes:

- node detects local link cost change
- updates distance table (line 15)
- if cost change in least cost path, notify neighbors (lines 23,24)

"good news travels fast"

D^Y	via X Z	D^X	X Z	D^Y	via X Z	D^X	X Z	D^Y	via X Z	D^X	X Z
X	(4) 6	X	(1) 6	X	(1) 6	X	(1) 6	X	(1) 3	X	(1) 3

algorithm terminates

D^Z	via X Y	D^X	X Y	D^Z	via X Y	D^X	X Y	D^Z	via X Y	D^X	X Y
X	50 (5)	X	50 (5)	X	50 (2)	X	50 (2)	X	50 (2)	X	50 (2)

time t_0 t_1 t_2

WS 2004/05, fu@informatik.cs.uni-goettingen.de 17

Telematics group
University of Göttingen, Germany

Distance Vector: link cost changes

Link cost changes:

- good news travels fast
- bad news travels slow - "count to infinity" problem!

D^Y	via X Z	D^X	X Z	D^Y	via X Z	D^X	X Z	D^Y	via X Z	D^X	X Z
X	(4) 6	X	(6) 6	X	(6) 6	X	(6) 6	X	(8) 6	X	(8) 6

algorithm continues on!

D^Z	via X Y	D^X	X Y	D^Z	via X Y	D^X	X Y	D^Z	via X Y	D^X	X Y
X	50 (5)	X	50 (5)	X	50 (7)	X	50 (7)	X	50 (7)	X	50 (9)

time t_0 t_1 t_2 t_3 t_4

WS 2004/05, fu@informatik.cs.uni-goettingen.de 18

Telematics group
University of Göttingen, Germany

Distance Vector: poisoned reverse

If Z routes through Y to get to X :

- Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- will this completely solve count to infinity problem?

D^Y	via X Z	D^X	X Z	D^Y	via X Z	D^X	X Z	D^Y	via X Z	D^X	X Z
X	(4) ∞	X	(60) ∞	X	(60) ∞	X	(60) ∞	X	(60) (5)	X	(60) (5)

algorithm terminates

D^Z	via X Y	D^X	X Y	D^Z	via X Y	D^X	X Y	D^Z	via X Y	D^X	X Y
X	50 (5)	X	50 (5)	X	50 (61)	X	50 (61)	X	50 (61)	X	50 (61)

time t_0 t_1 t_2 t_3 t_4

WS 2004/05, fu@informatik.cs.uni-goettingen.de 19

Telematics group
University of Göttingen, Germany

Comparison of LS and DV algorithms

Message complexity

- LS:** with n nodes, E links, $O(nE)$ msgs sent each
- DV:** exchange between neighbors only
 - convergence time varies

Speed of Convergence

- LS:** $O(n^2)$ algorithm requires $O(nE)$ msgs
 - may have oscillations
- DV:** convergence time varies
 - may be routing loops
 - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect link cost
- each node computes only its own table

DV:

- DV node can advertise incorrect path cost
- each node's table used by others
 - error propagate thru network

WS 2004/05, fu@informatik.cs.uni-goettingen.de 20

Hierarchical Routing

Our routing study thus far - idealization

- all routers identical
- network "flat"
- ... *not* true in practice

scale: with 200 million destinations:

- can't store all dest's in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

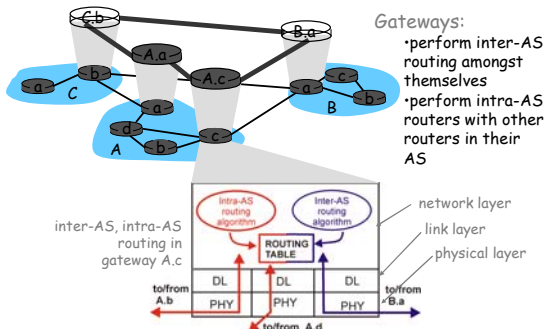
Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
 - "intra-AS" routing protocol
 - routers in different AS can run different intra-AS routing protocol

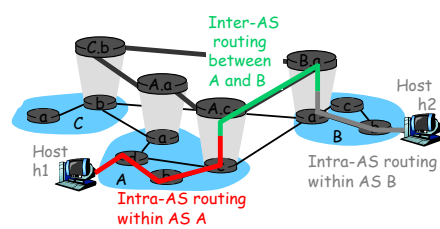
gateway routers

- special routers in AS
- run intra-AS routing protocol with all other routers in AS
- also responsible for routing to destinations outside AS
 - run *inter-AS routing* protocol with other gateway routers

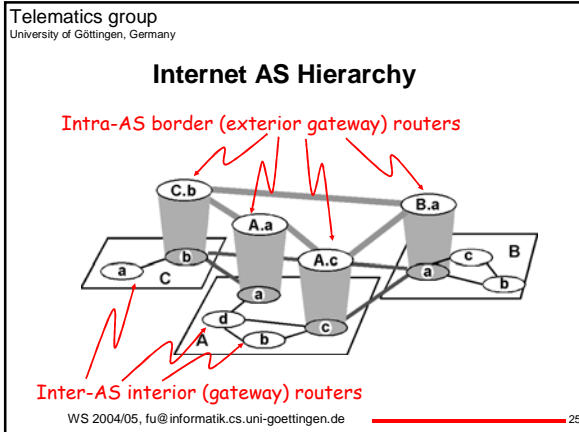
Intra-AS and Inter-AS routing



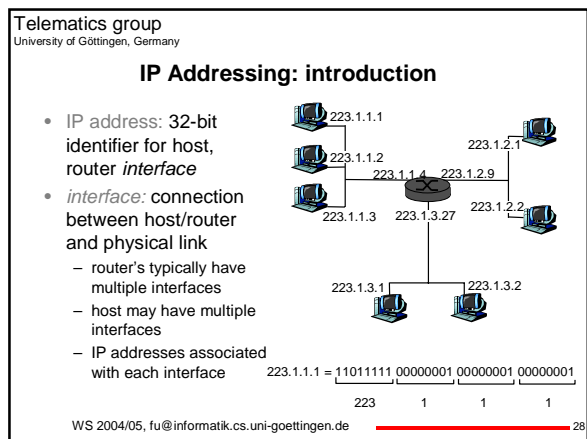
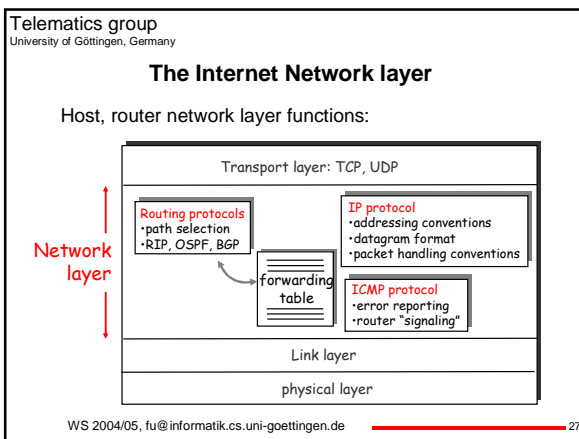
Intra-AS and Inter-AS routing



- Self-learning: typical Internet routing protocols:
 - inter-AS: BGP
 - intra-AS: RIP, OSPF



- Telematics group
University of Göttingen, Germany
- ### Intra-AS Routing
- Also known as **Interior Gateway Protocols (IGP)**
 - Most common Intra-AS routing protocols:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)
- WS 2004/05, fu@informatik.cs.uni-goettingen.de 26



Telematics group
University of Göttingen, Germany

Getting a datagram from source to dest.

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

IP datagram:

misc fields	source IP addr	dest IP addr	data

- datagram remains **unchanged**, as it travels source to destination
- addr fields of interest here

WS 2004/05, fu@informatik.cs.uni-goettingen.de 29

Telematics group
University of Göttingen, Germany

Getting a datagram from source to dest.

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

Starting at A, send IP datagram addressed to B:

misc fields	223.1.1.1	223.1.1.3	data

- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
 - B and A are directly connected

WS 2004/05, fu@informatik.cs.uni-goettingen.de 30

Telematics group
University of Göttingen, Germany

Getting a datagram from source to dest.

forwarding table in router

Dest. Net.	router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27

Arriving at 223.1.4, destined for 223.1.2.2

- look up network address of E in router's forwarding table
- E on *same* network as router's interface 223.1.2.9
 - router, E directly attached
- link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!!

WS 2004/05, fu@informatik.cs.uni-goettingen.de 31

Telematics group
University of Göttingen, Germany

Homework

- In „J. Kurose/K. Ross, Computer Networking“, Chapter 4 Problems (pp.409-410) No. 3-5, 7
- Compare and contrast LS and DV algorithms
- What is the difference between routing and addressing?
- Why are different inter-AS and intra-AS protocols used in the Internet?

WS 2004/05, fu@informatik.cs.uni-goettingen.de 32

Telematics group
University of Göttingen, Germany

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)
 - Can you guess why?
- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called **advertisement**)
- Each advertisement: list of up to 25 destination nets within AS

WS 2004/05, fu@informatik.cs.uni-goettingen.de 33

Telematics group
University of Göttingen, Germany

RIP: Example

Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
...

Routing table in D

WS 2004/05, fu@informatik.cs.uni-goettingen.de 34

Telematics group
University of Göttingen, Germany

RIP: Example

Advertisement from A to D

Dest	Next	hops
w	-	-
x	-	-
z	C	4
...

Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B A	7 5
x	--	1
...

Routing table in D

WS 2004/05, fu@informatik.cs.uni-goettingen.de 35

Telematics group
University of Göttingen, Germany

RIP: Link Failure and Recovery

If no advertisement heard after 180 sec --> neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly propagates to entire net
- poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

WS 2004/05, fu@informatik.cs.uni-goettingen.de 36

Telematics group
University of Göttingen, Germany

RIP Table processing

- RIP routing tables managed by **application-level** process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated

WS 2004/05, fu@informatik.cs.uni-goettingen.de 37

Telematics group
University of Göttingen, Germany

RIP Table example (continued)

Router: *giroulee.eurocom.fr*

Destination	Gateway	Flags	Ref	Use	Interface
127.0.0.1	127.0.0.1	UH	0	26492	lo0
192.168.2.	192.168.2.5	U	2	13	fa0
193.55.114.	193.55.114.6	U	3	58503	le0
192.168.3.	192.168.3.5	U	2	25	qaa0
224.0.0.0	193.55.114.6	U	3	0	le0
default	193.55.114.129	UG	0	143454	

- Three attached class C networks (LANs)
- Router only knows routes to attached LANs
- Default router used to "go up"
- Route multicast address: 224.0.0.0
- Loopback interface (for debugging)

WS 2004/05, fu@informatik.cs.uni-goettingen.de 38

Telematics group
University of Göttingen, Germany

OSPF (Open Shortest Path First)

- "open": publicly available
- Uses Link State algorithm
 - LS packet dissemination
 - Topology map at each node
 - Route computation using Dijkstra's algorithm
- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to **entire** AS (via flooding)
 - Carried in OSPF messages directly over IP (rather than TCP or UDP)

WS 2004/05, fu@informatik.cs.uni-goettingen.de 39

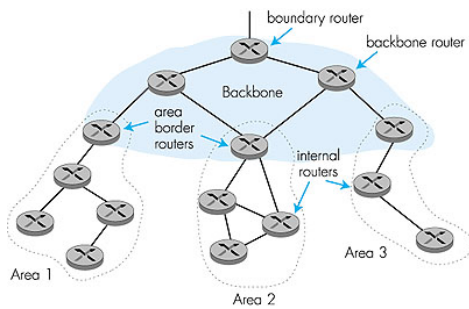
Telematics group
University of Göttingen, Germany

OSPF "advanced" features (not in RIP)

- Security**: all OSPF messages authenticated (to prevent malicious intrusion)
- Multiple same-cost paths** allowed (only one path in RIP)
- For each link, multiple cost metrics for different **TOS** (e.g., satellite link cost set "low" for best effort; high for real time)
- Integrated uni- and **multicast** support:
 - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- Hierarchical** OSPF in large domains.

WS 2004/05, fu@informatik.cs.uni-goettingen.de 40

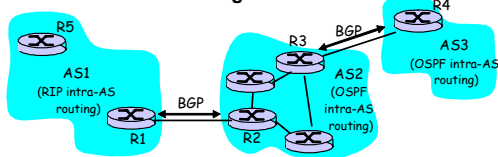
Hierarchical OSPF



Hierarchical OSPF

- **Two-level hierarchy:** local area, backbone.
 - Link-state advertisements only in area
 - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers:** “summarize” distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers:** run OSPF routing limited to backbone.
- **Boundary routers:** connect to other AS's.

Inter-AS routing in the Internet: BGP



Internet inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** *the de facto standard*
- **Path Vector** protocol:
 - similar to Distance Vector protocol
 - each Border Gateway broadcast to neighbors (peers) *entire path* (i.e., sequence of AS's) to destination
 - BGP routes to networks (ASs), not individual hosts
 - E.g., Gateway X may send its path to dest. Z:

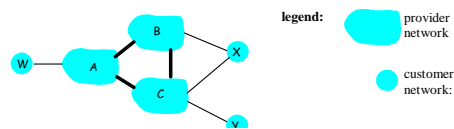
Path (X,Z) = X,Y1,Y2,Y3,...,Z

Internet inter-AS routing: BGP

Suppose: gateway X send its path to peer gateway W

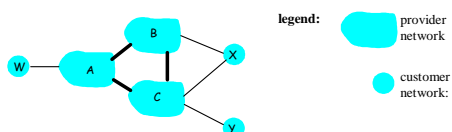
- W may or may not select path offered by X
 - cost, policy (don't route via competitors AS), loop prevention reasons.
- If W selects path advertised by X, then:
 - Path (W,Z) = w, Path (X,Z)
- Note: X can control incoming traffic by controlling it route advertisements to peers:
 - e.g., don't want to route traffic to Z -> don't advertise any routes to Z

BGP: controlling who routes to you



- A,B,C are **provider networks**
- X,W,Y are customer (of provider networks)
- X is **dual-homed**: attached to two networks
 - X does not want to route from B via X to C
 - .. so X will not advertise to B a route to C

BGP: controlling who routes to you



- A advertises to B the path AW
- B advertises to W the path BAW
- Should B advertise to C the path BAW?
 - No way! B gets no "revenue" for routing CBAW since neither W nor C are B's customers
 - B wants to force C to route to w via A
 - B wants to route **only** to/from its customers!

BGP operation

Q: What does a BGP router do?

- Receiving and filtering route advertisements from directly attached neighbor(s).
- Route selection.
 - To route to destination X, which path (of several advertised) will be taken?
- Sending route advertisements to neighbors.

BGP messages

- BGP messages exchanged using TCP.
- BGP messages:
 - **OPEN**: opens TCP connection to peer and authenticates sender
 - **UPDATE**: advertises new path (or withdraws old)
 - **KEEPALIVE** keeps connection alive in absence of UPDATES; also ACKs OPEN request
 - **NOTIFICATION**: reports errors in previous msg; also used to close connection

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, reduced update traffic

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance