

Chapter 4

Network Layer: The Data Plane

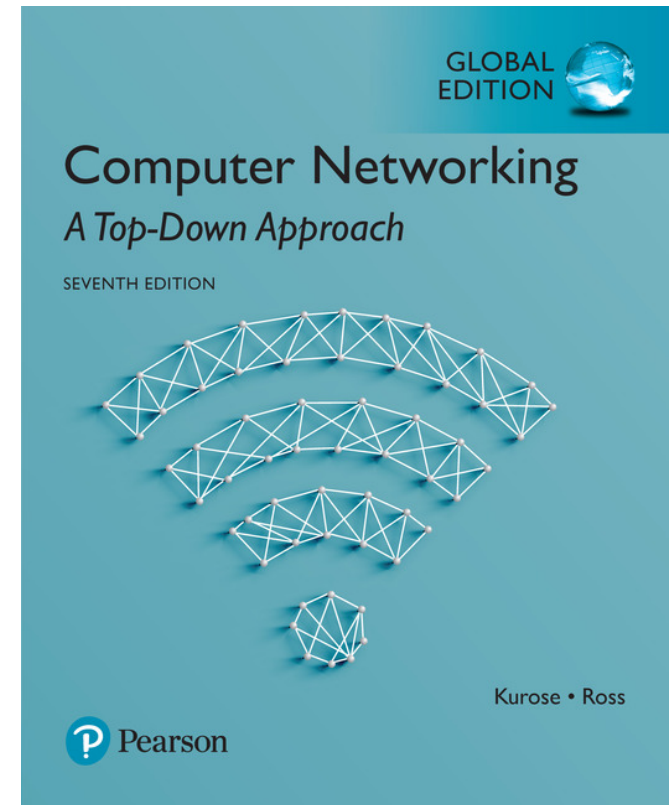
A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

© All material copyright 1996-2016
J.F Kurose and K.W. Ross, All Rights Reserved



Computer Networking: A Top Down Approach

7th Edition, Global Edition
Jim Kurose, Keith Ross
Pearson
April 2016

Chapter 4: outline

4.1 Sare geruza, gainbegirada bat

- Informazio planoa
- Kontrol planoa

4.2 Zer dago Router batean?

4.3 IP: Internet Protocol

- datagramen formatua
- zatikaketa
- IPv4 helbideraketa
- network address translation (NAT)
- IPv6

4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

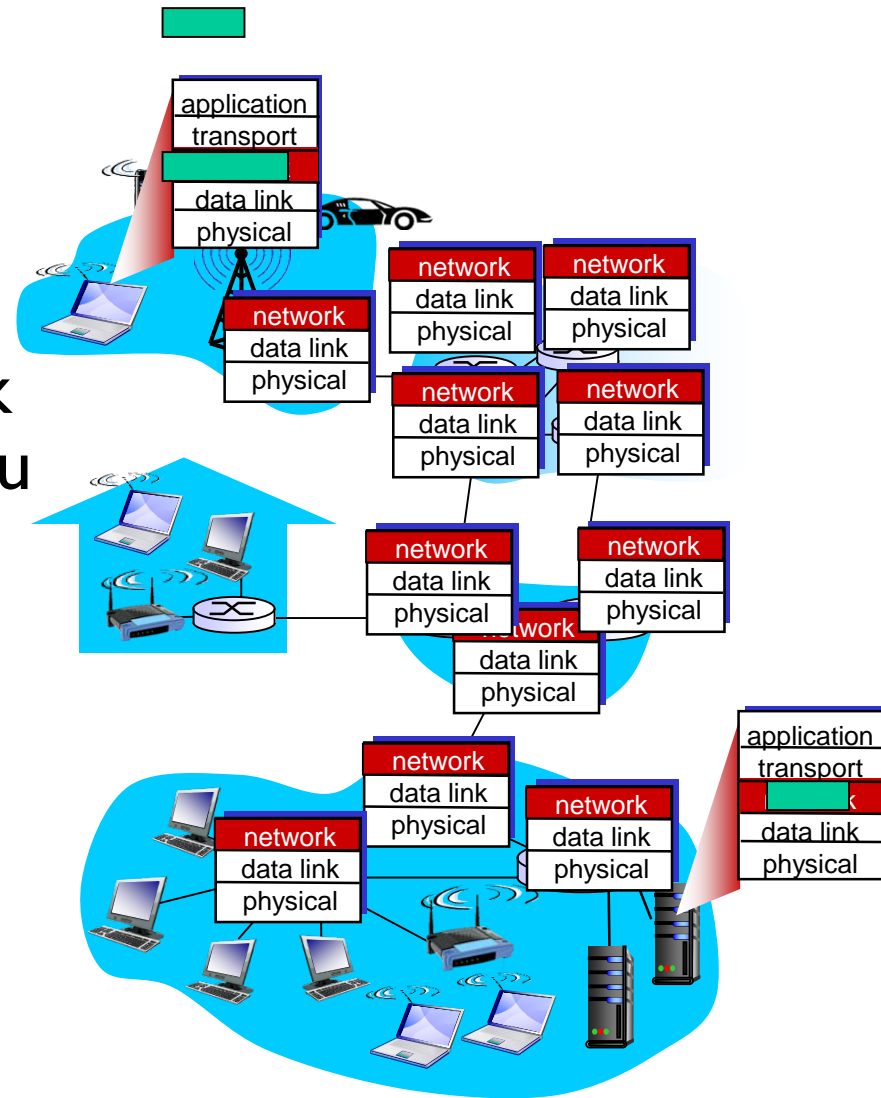
Chapter 4: sare geruza

Helburuak:

- Sare geruzan dauden zerbitzuen ulermena, informazio planoaz aztertuz:
 - Sare geruzaren zerbitzu ereduak
 - forwarding versus routing
 - Nola funtzionatzen du Router bat
 - generalized forwarding
- Inplementazioa interneten

Sare geruza

- Segmentuak garraiatzen ditu **igorle** eta **jasotzaile**aren artean
- Igorlearen aldean, segmentuak datagrametan kapsulatzen ditu
- Jasotzailearen aldean, segmentuak garraio geruzara pasatzen ditu
- Sare geruzako protokoloak host *guztietan* daude, eta router-etan
- Router-ak aztertzen ditu bera zeharkatzen duten IP datagrama guztien goiburua



Sare geruzaren oinarrizko bi funtzioak

Sare geruzaren funtzioak:

- *forwarding*: paketeak mugitzen ditu routerraren sarreratik irteera egokira
- *routing*: paketeek igorletik jasotzailera hartzen duten bidearen aukeraketa
 - *Bideraketa algoritmoak*

analogia: bidai bat

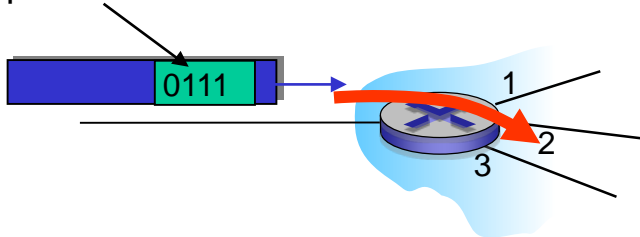
- *forwarding*: bidaiari egiten dioten bide aukeraketak
- *routing*: bidaiaren planifikazioa irteeratik helmugara

Sare geruza: informazio planoa, kontrol planoa

Informazio planoa

- lokala, per-router funtzioa
- Routerren sarrerara heltzen den datagrama routerraren irteerako portura nola bireratzen den
- **forwarding** funtzioa betetzen du

values in arriving packet header

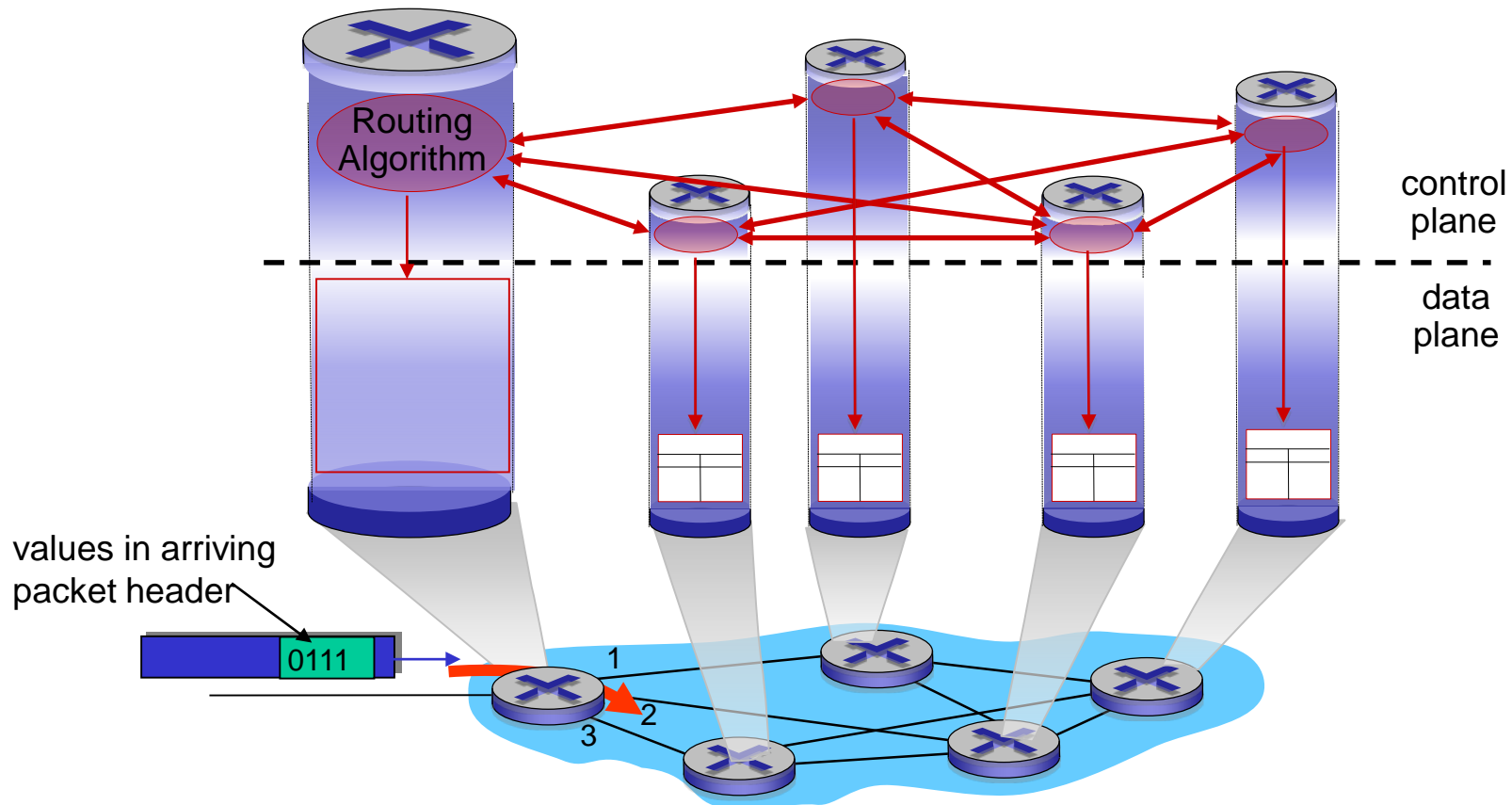


Kontrol planoa

- network-wide logic
- datagram nola bideratzen den routerren artean horren end-end bidean, iturritik helmugara
- Bi urbilketa kontrol planora:
 - *traditional routing algorithms*: routerretan implementatzen dira
 - *software-defined networking (SDN)*: (Hurruneko) serbitzarietan implementatuta

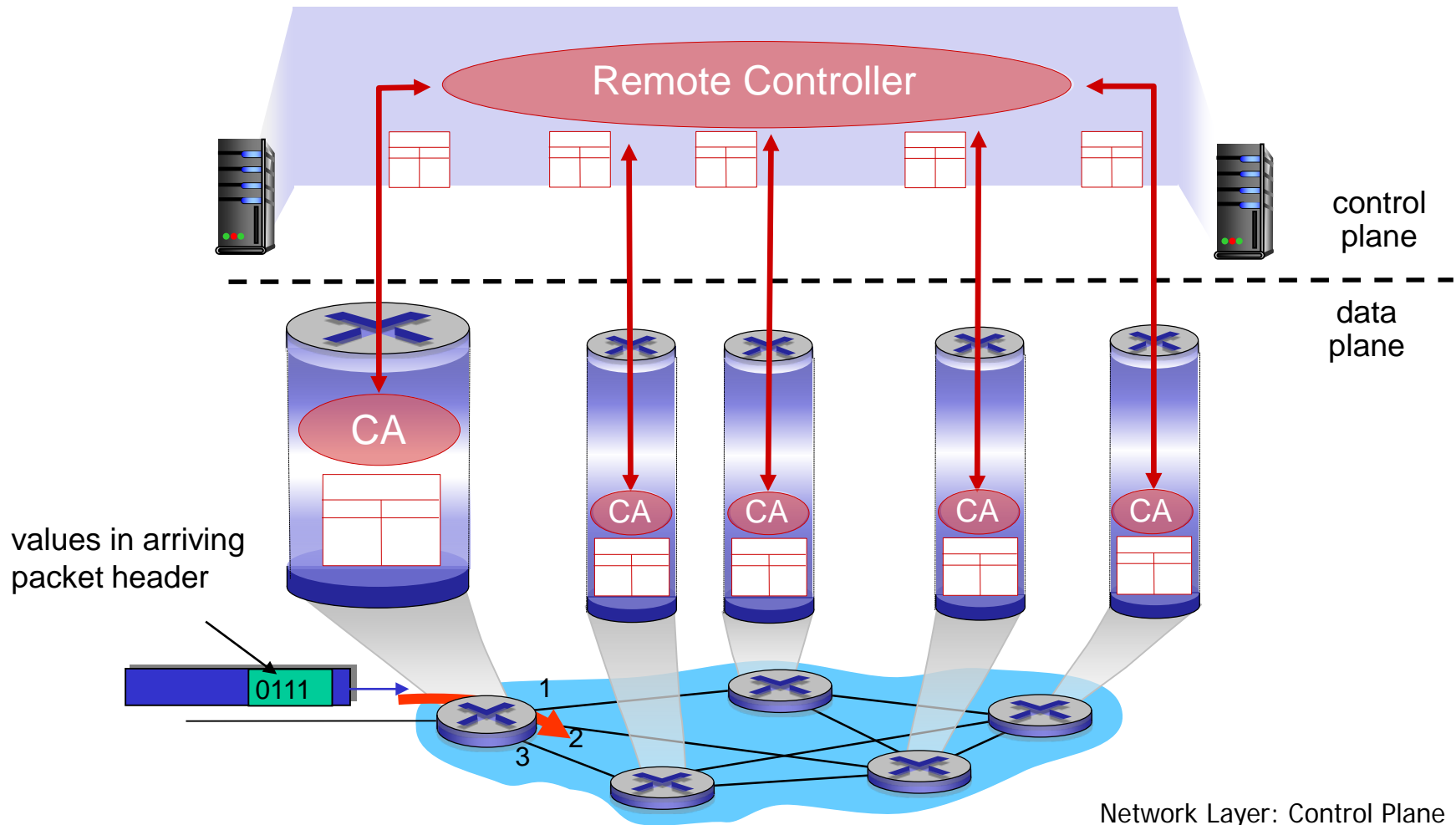
Per-router control plane

Router *bakoitzean eta guztietan* dauden bideraketa algoritmo konponenteek kontrol planoan interaktuatzeko dute



Logikoki zentralizatutako kontrol plano (SDN)

(Normalean remotoa den) kontrola, kontrol agente (CA) lokalekin elkarreragiten dute



Network service model

Q: Zer *zerbitzu eredu*a igorle-jasotzaile
“kanalizatutako” datagramentzat?

Banakako datagramentzako zerbitzu adibideak:

- Bidalketa zihurtatuta
- Bidalketa zihurtatuta,
atzerapena 40 msec baino
txikiago izanik

Datagrama fluxuentzako zerbitzu adibideak:

- Ordenean bidalitako
datagramak
- Banda zabalera minimoa
zihurtatzea
- Murrizketak paketeen
arteko tarteen aldeketen
artean

Eredu zerbitzuak sare geruzarentzako:

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

INTERNET EZ DA SARE PROTOKOLO BAKARRA!!

Chapter 4: outline

4.1 Sare geruza, gainbegirada bat

- Informazio planoa
- Kontrol planoa

4.2 Zer dago Router batean?

4.3 IP: Internet Protocol

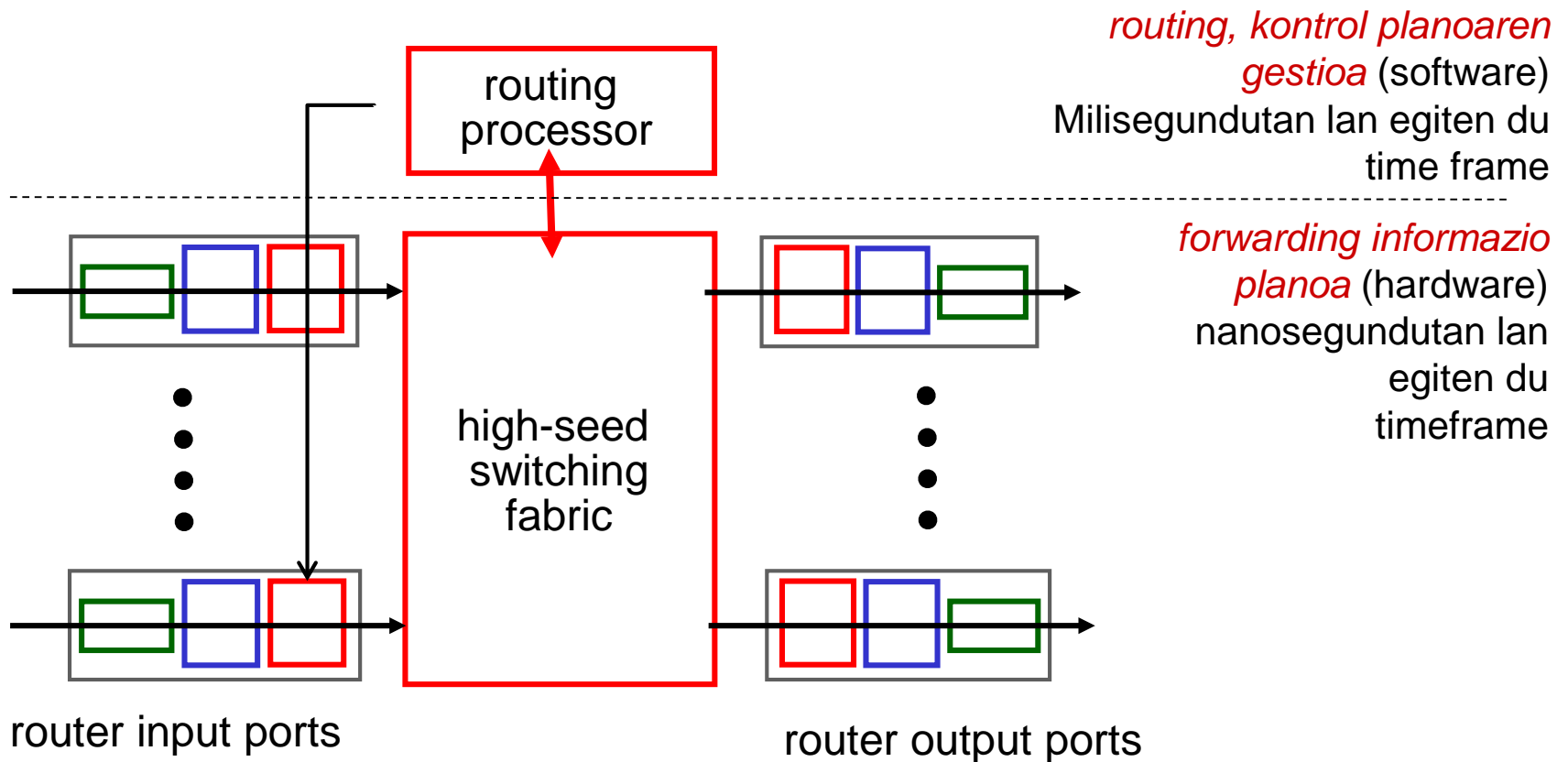
- datagramen formatua
- zatikaketa
- IPv4 helbideraketa
- network address translation (NAT)
- IPv6

4.4 Generalized Forward and SDN

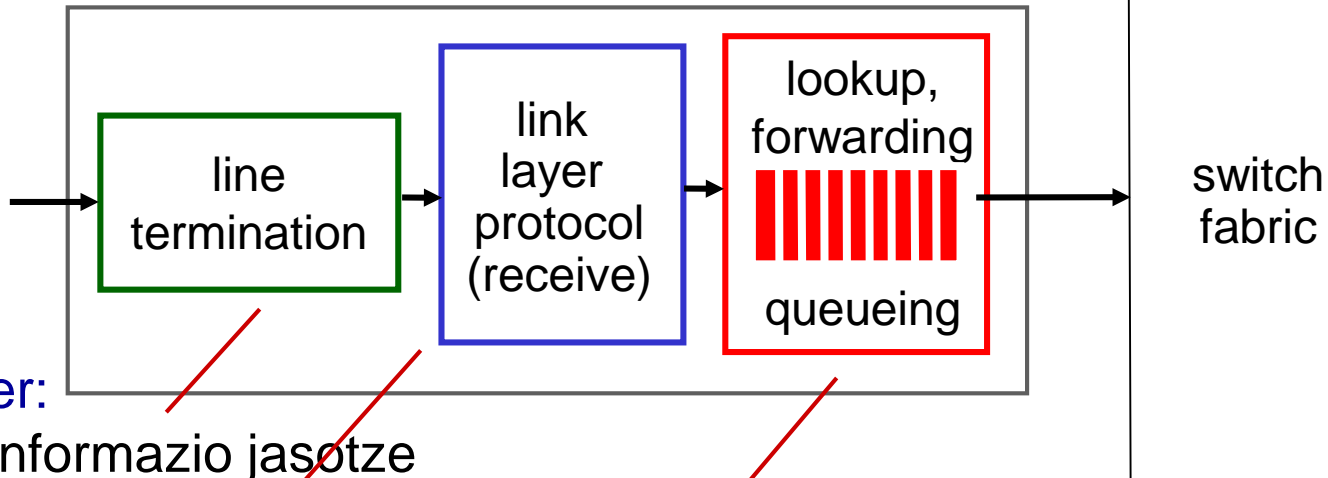
- match
- action
- OpenFlow examples of match-plus-action in action

Gainbegirada bat Routerren arkitekturari

- Router generiko baten arkitekturaren goi-mailako ikuspuntua:



Sarrerako portuen funtzioak



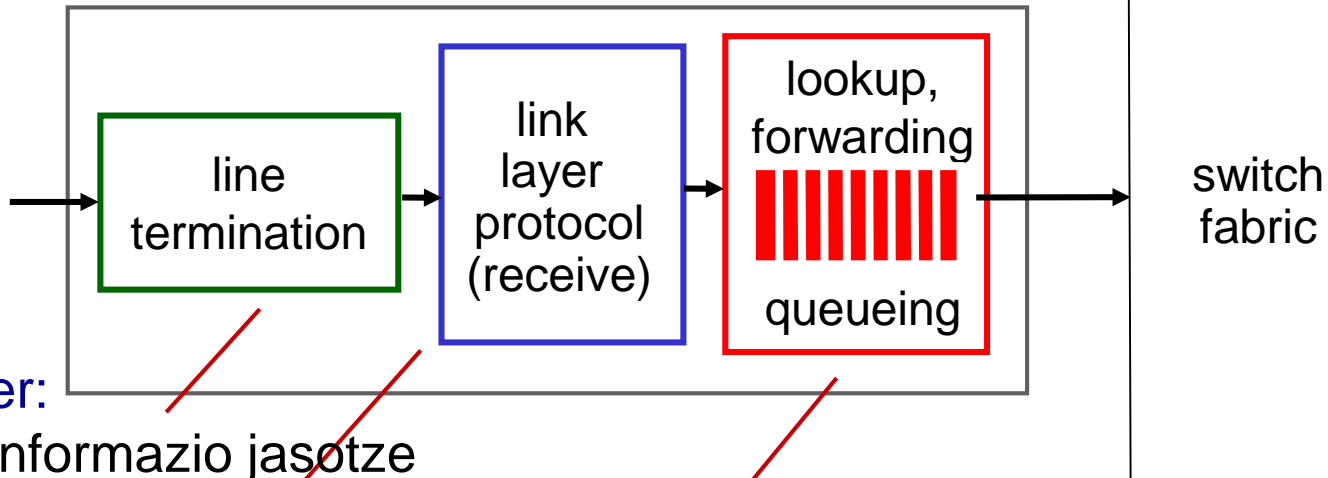
physical layer:
Bit mailako informazio jasotze

data link layer:
e.g., Ethernet

decentralized switching:

- Goiburuen balioak erabiltuta, irteerako portua bilatzen da sarrerako portuaren bideratze taula erabiliz (“*match plus action*”)
- helburua: bideraketaren prozesatua ‘line speed’abiaduran
- Queuing - pilaketak: datagramak heltzen dira bidera daitezkeen baino arinago

Sarrerako portuen funtzioak



physical layer:

Bit mailako informazio jasotze

data link layer:

e.g., Ethernet

decentralized switching:

- Goiburuen balioak erabilita, irteerako portua bilatzen da sarrerako portuaren bideratze taula erabiliz (“*match plus action*”)
- **destination-based forwarding:** bideraketa helburuko IP helbidean oinarrituta (traditional)
- **generalized forwarding:** bideraketa goiburuaren eremuetan oinarrituta (ez bakarrik IPTan)

Helburuan oinarritutako bideraketa

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010 000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011 001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Bat egiten duen aurrizki luzeena

Bat egiten duen aurrizki luzeena

Bideraketako sarreraren taulan begiratzen denean emandako helburu helbidearentzako, bat egiten duen aurrezki *luzeena* erabiltzen du.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

adibideak:

DA: 11001000 00010111 00010110 10100001

which interface?

DA: 11001000 00010111 00011000 10101010

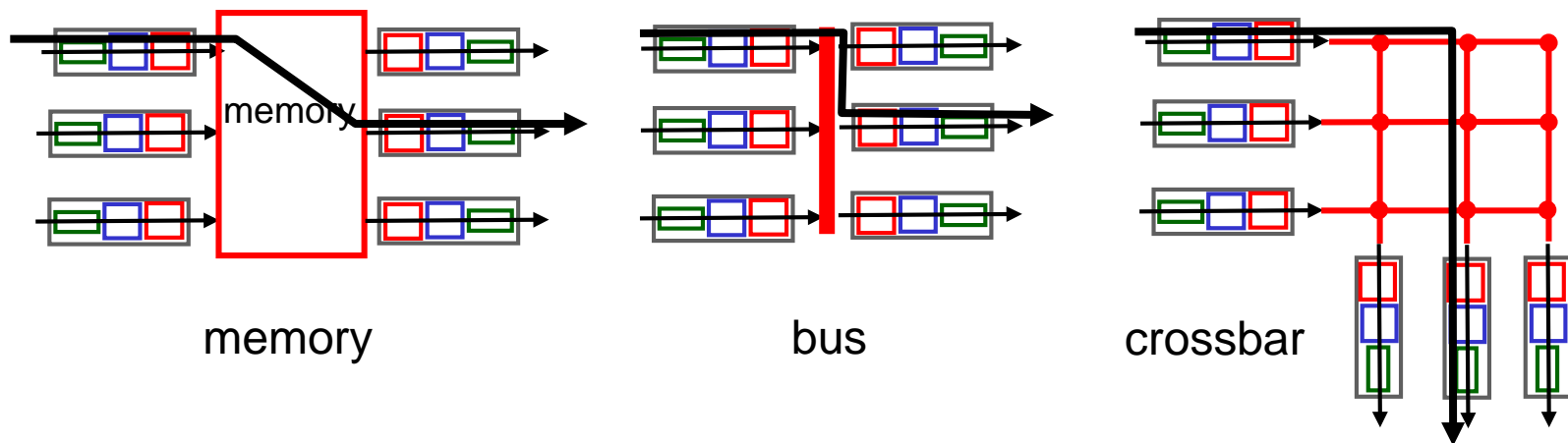
which interface?

Bat egiten duen aurrizki luzeena

- we'll see *why* longest prefix matching is used shortly, when we study addressing
- longest prefix matching: often performed using ternary content addressable memories (TCAMs)
 - *content addressable*: present address to TCAM: retrieve address in one clock cycle, regardless of table size
 - Cisco Catalyst: can up ~1M routing table entries in TCAM

Switching fabrics

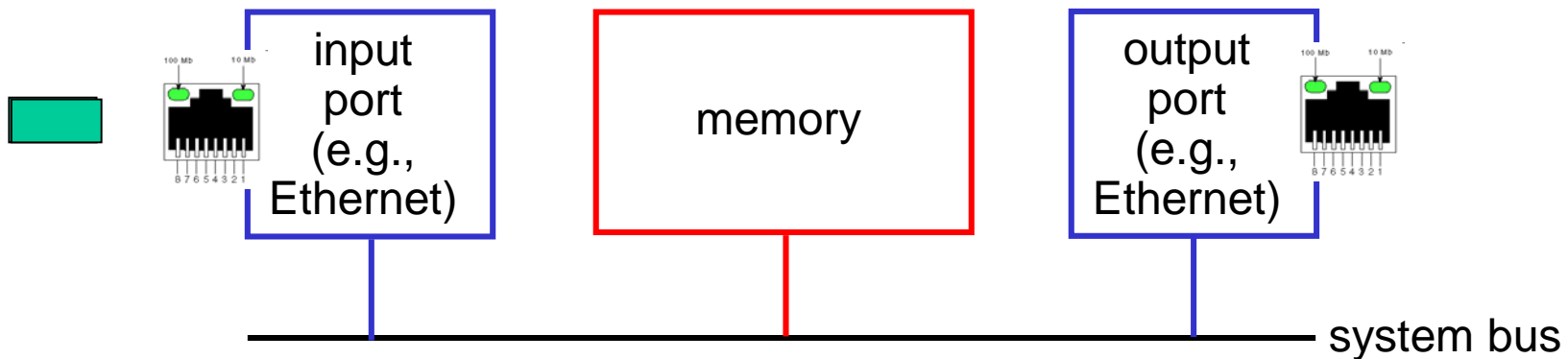
- Paketeen bideratzea sarrerako buffer-etik irteerako buffer egokira
- switching rate: paketeak sarreratik irteerara bidaltzeko abiadura
 - Batzutan sarrera/irtera anitzen abiadura neurtzen da
 - N inputs: switching rate N times line rate desirable
- Hiru mota



Switching via memory

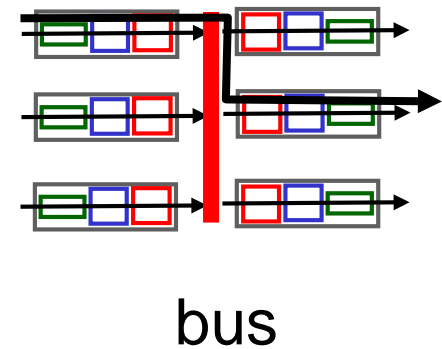
Lehen belaunaldiko routerrak:

- Ohiko konputagailuak, CPU-k zuzenean kontrolatzen du switching-a
- Paketea sistemaren memorian kopiatzen da
- Memoriaren banda zabalera abiadura mugatzen du (2 bus crossings per datagram)



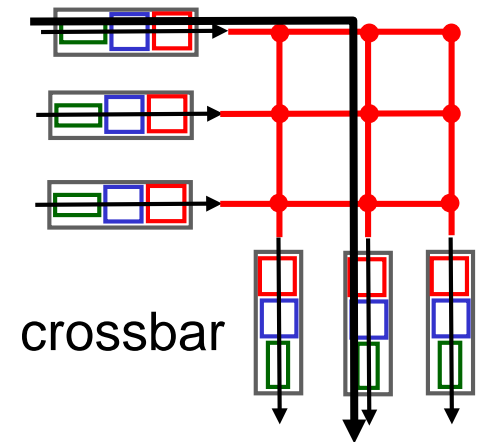
Switching via a bus

- Datagrama memoriaren sarrerako portutik irterakora doa partekatutako buffer baten bidez
- *bus contention*: switching speed (trakaketa abiadura) busaren banda zabaleraz mugatuta
- 32 Gbps bus, Cisco 5600: behar bezain azkarra atzipen eta enpresen routerrentzat



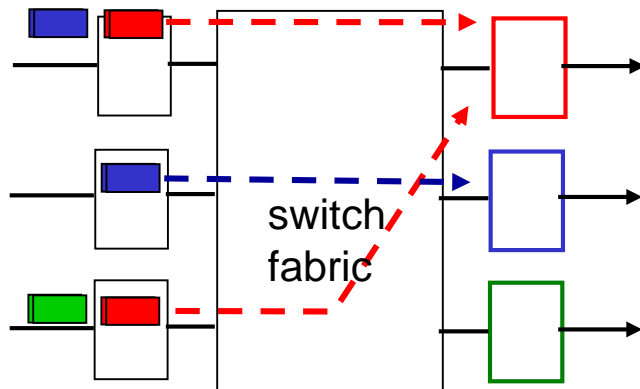
Switching via interconnection network

- Bus-en banda zabalen mugak gainditzen ditu
- banyan networks, crossbar, mikroak multiprozesadore ingurunetan komunikatzeko metodoak
- Diseinu aurreratua: datagramak zatitzen ditu cells luzera finkoko tarteetan banatzen ditu, eta ehunetik bidaltzen ditu.
- Cisco I2000: switches 60 Gbps through the interconnection network



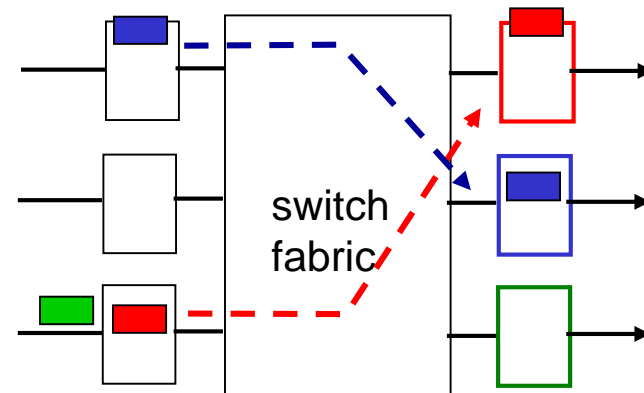
Lerrokaketa sarrerako portuetan

- Ehuna sarrerako portuak baino motelago demean, pilaketak ager daitezke sarreran
 - *Atzerapenak eta galerak ager daitezke sarrerako bufferrak gainezka egiten duenean!*
- **Head-of-the-Line (HOL) blocking:** lerrokatuta dauden datagramak, heltzen diren berriak aurrera egiteko ekiditen dute



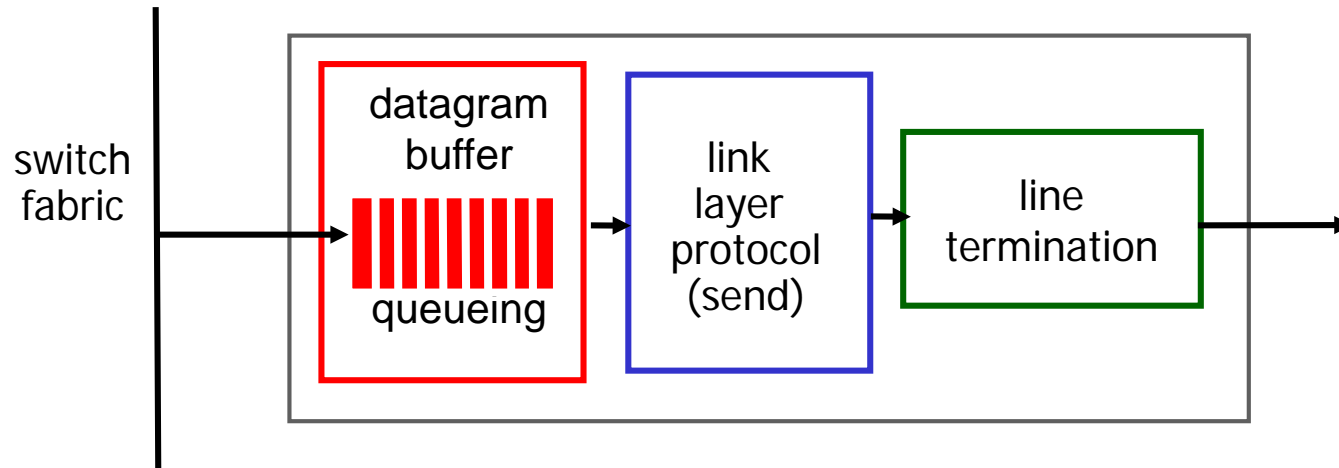
output port contention:
Bakarrik transmitituko da
datagrama gorri bat.

Beheko pakete gorria blokeatzen da



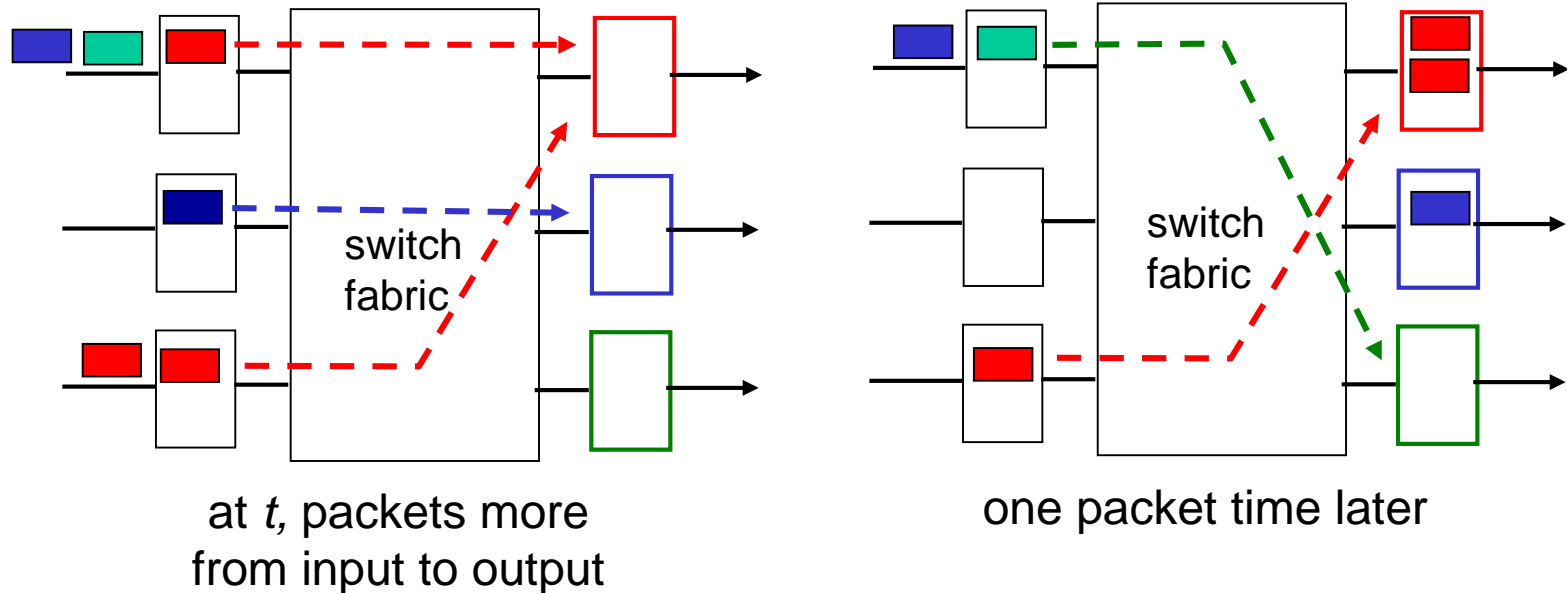
Pakete baten atzerapena:
Datagrama berdeak
HOL pairatzen du

Irteerako portuak *This slide is HUGEY important!*




- Ehunetik heltzen abiadura biderake bada, *buffer* batea
Datagramak (paketeak) gal daitezke kongestioagatik, bufferren falta
- *scheduling* datagrame
Priority scheduling – nork lortzen ditu errekurtsok, sarearen neutraltasuna

Lerrokaketa irteerako portuetan



- Buffer-ean sartu heltzen diren paketeen abiadura irteerakoa baino handiago demean
- *Lerrokaketa, pilaketa (atzerapena) eta paketen galera irteerako portuan overflow dagoenean!*

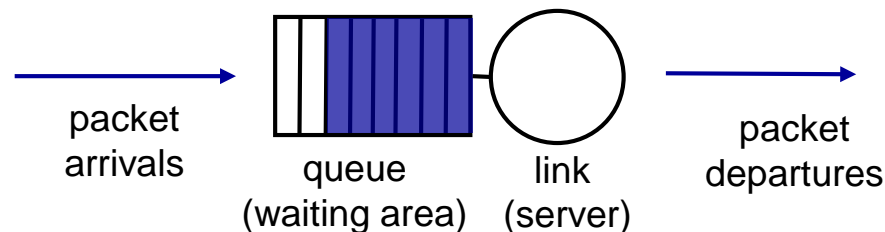
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C  Round Trip Time
 - e.g., $C = 10$ Gpbs link: 2.5 Gbit buffer
- recent recommendation: with N flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Scheduling mechanisms

- *scheduling*: bidali behar den hurrengo paketearen aukeraketa
- *FIFO (first in first out) scheduling*: heltzen den ordenean bidalisend in order of arrival to queue
 - real-world example?
 - *discard policy*: paketeak betetako hilara batera heltzen badira, zein galduko da?
 - *tail drop*: drop arriving packet
 - *priority*: drop/remove on priority basis
 - *random*: drop/remove randomly



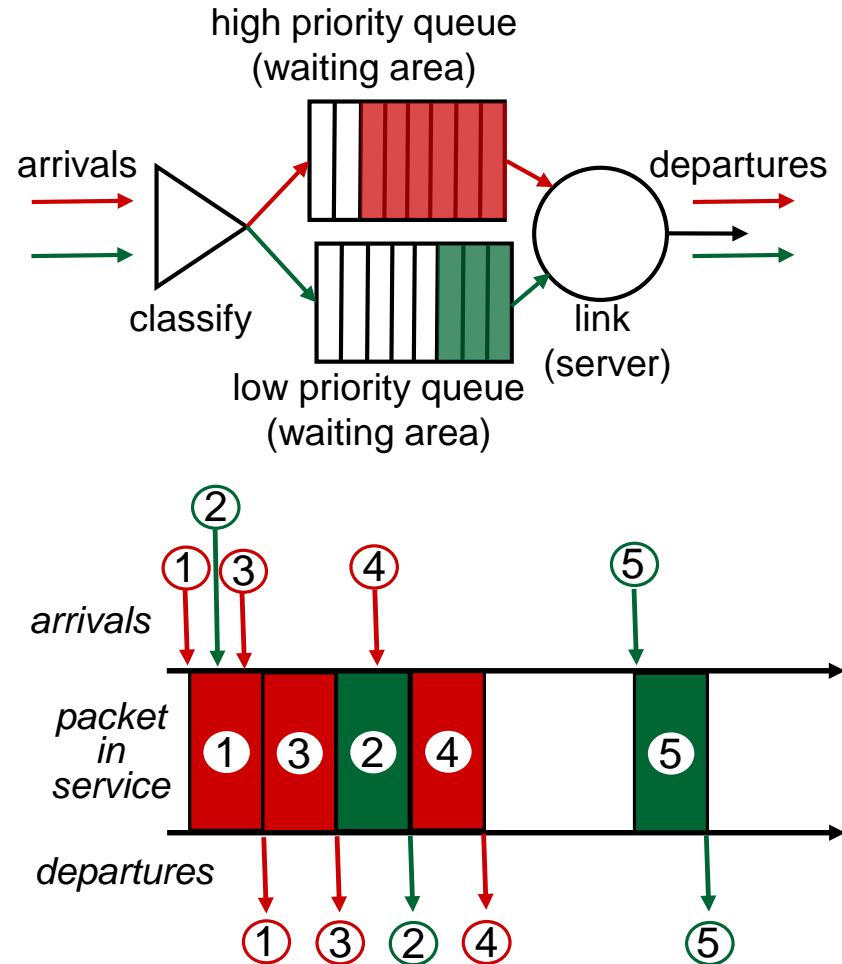
Scheduling politikak: lehentasuna

priority scheduling:

lehentasuna handien
duen paketea bidali

- *pakete mota*
desberdinak,
lehentasun
desberdinekin

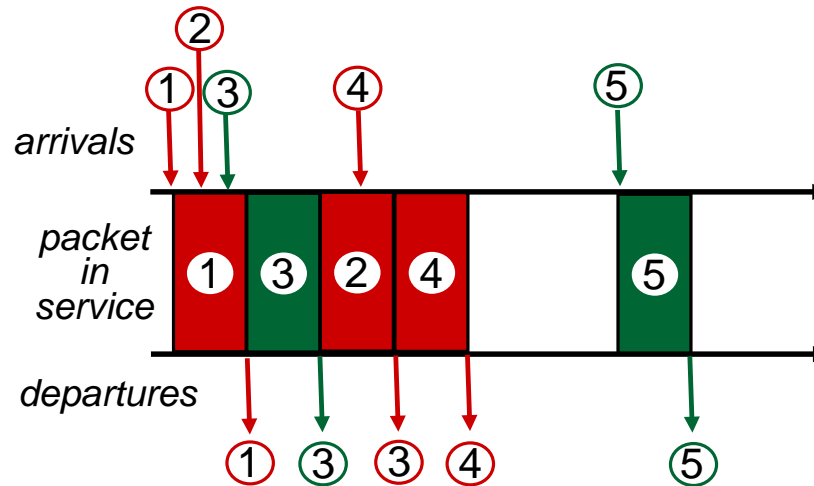
- Pakete mota era
ezberdinez sailka
daiteke: markak edo
goiburuaren
informazioa (IP
helbideak, portu
zenbakiak...)
- real world example?



Scheduling policies: gehiago

Round Robin (RR) scheduling:

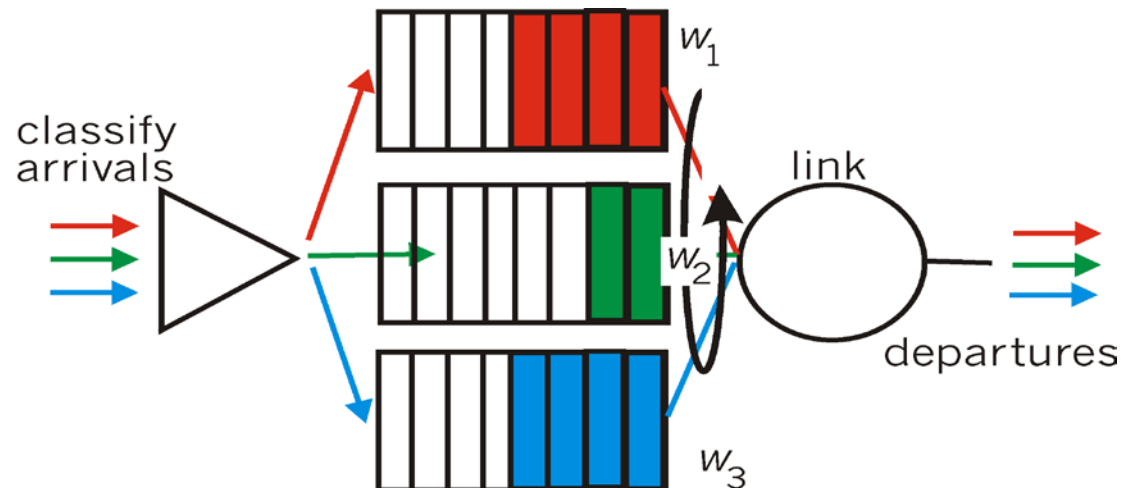
- multiple classes
- cyclically scan class queues, sending one complete packet from each class (if available)
- real world example?



Scheduling policies: gehiago

Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class gets weighted amount of service in each cycle
- real-world example?



Chapter 4: outline

4.1 Sare geruza, gainbegirada bat

- Informazio planoa
- Kontrol planoa

4.2 Zer dago Router batean?

4.3 IP: Internet Protocol

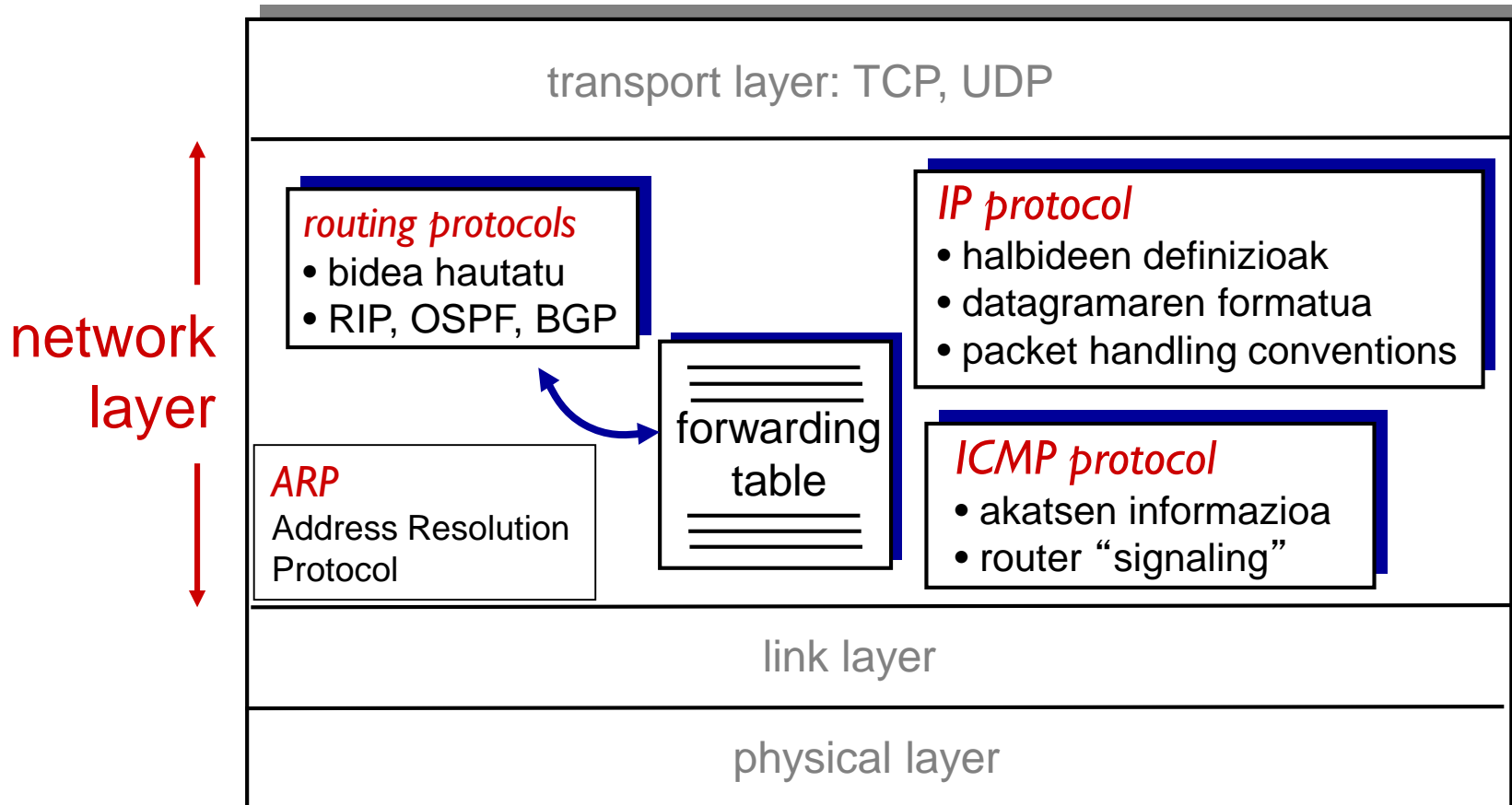
- datagramen formatua
- zatikaketa
- IPv4 helbideraketa
- network address translation (NAT)
- IPv6

4.4 Generalized Forward and SDN

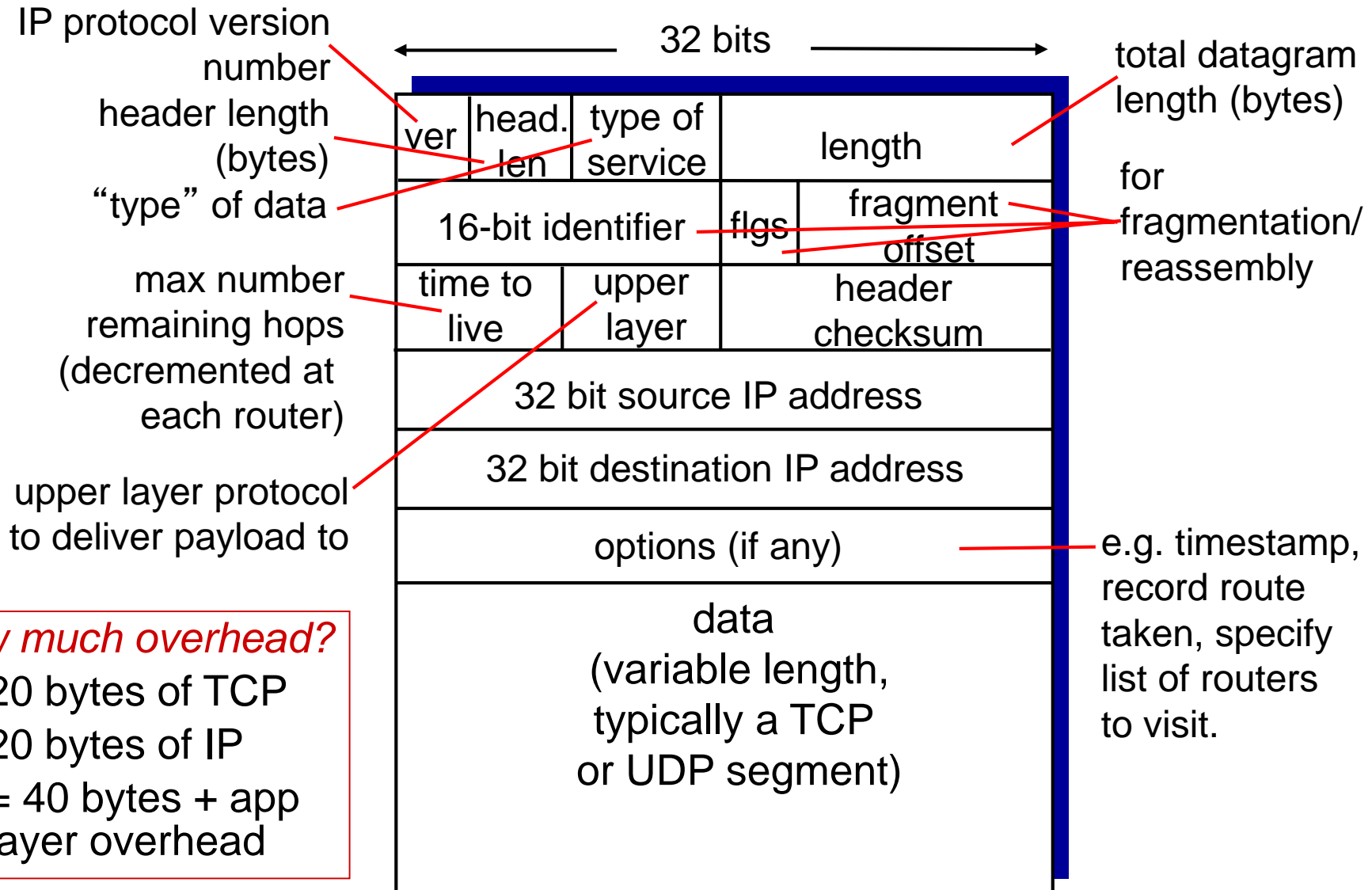
- match
- action
- OpenFlow examples of match-plus-action in action

Internetaren sare geruza

host, router network layer functions:



IP datagram format



IP datagram format

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	13	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bertsioa				goiburuko luzera				Zerbitzu mota								Datagramaren luzera															
Datagramaren identifikazioa (16 bit)Identifikazioa																D	M	Desplazamendua (13 bit)													
																F	F														
TTL (iraupena)								Goiko protokoloa								Erroreak atzementeko funtzioa															
Iturburuko IP helbidea																															
Helburuko IP helbidea																															
Aukerak (egotekotan)																															
Datuak																															

IP datagram format (Wikipedia)

- **Bertsioa** (4 bit)

Honek datagramak nolako goiburukoa duen adierazten du, bertsio ezberdinek datagrama ezberdinak erabiltzen baitituzte. Hau da, eremu honek goiburukoa irakurtzeko gakoa ematen die datagrama aztertu behar duten programei. Horregatik da goiburukoaren lehenengo eremua.

- **Goiburukoaren luzera**

Datagrama prozesatu behar duten IP entitateek (bideko [bideratzaileenak](#) eta helburuko konputagailuarena) jakin behar dute goiburukoa non bukatzen den eta garraiatutako datuak non hasten diren. Goiburukoan aukerazko eremu batzuk daudenez, haren luzera ez da finkoa, eta eremu hau beharrezkoa suertatzen da.

- **Zerbitzu mota**

Esparru honek erabiltzen diren zerbitzu ezberdinak (DiffServ) identifikatzen ditu. Zerbitzuen sailkapena egiten da eta horren arabera Quality of Service (QoS) egokitzen da. Honen adibidea, streaming aplikazioek behar duten latentzia txikirako sarearen egokitzapena litzateke

- **Datagramaren luzera**

Goiburukoaren luzera ez ezik, datu-eremuarena ere ez dago finkatuta. Hasiera batean, datagrama osoaren luzera jakitea arkitekturako mailen arteko interfazearen kontua da, eta ez luke goiburukoan agertu behar. Hau da, sarbide-mailako entitateak kontrolatzen du zenbat [byte](#) erauzten duen [tramatik](#), eta datu hori ematen dio IP mailako entitateari datagramarekin batera (nolabait «tori datagrama hau, hainbat bytetakoa» esango dio). Hala ere, sarbideko protokolo batzuek zaborra gehitzen diote transmititzeko ematen dieten datagramari. Hori da jatorrizko difusioko Ethernet sareen kasua, non talkak atzemango direla bermatzearen, tramek luzera minimo bat izan behar duten. Traman sartu behar den datagramak luzera minimo hori ez badu, zabor betegarria sartzen da tramaren informazio-eremuan. Gero, helburuan, datagramari itsatsita datorren betegarria bereizteko, datagramaren goiburukoan dugun luzera izeneko eremua erabiliko du IP entitateak. Luzera eremuan 16 [bit](#) daudenez, eta [bytetan](#) neurtzen denez, datagramarik handiena 65.536 [bytekoa](#) izan daiteke (datuak gehi goiburukoa). Dena dela, oso arraroa da 1.500 [byte](#) baina handiagoa den datagrama bat aurkitzea (hori da [Ethernet](#) sare batean sartzen den datagramarik handiena), eta sistema askok 576 bytera mugatzen dute datagramaren tamaina (eremu zabaleko sare askok onartzen duten tamaina maximoa).

- **Jatorrizko helbidea** (32 bit)

Irudian ikusten denez, helburuko helbidea ez ezik, jatorrizkoa ere datagramaren goiburukoan dago. [Bideratzaileek](#) ez dute jatorrizko helbide hori bideratzeko behar, nahikoa baitute helburukoarekin. Baina datagrama jasoko duenak, normalki, erantzuna eman beharko dio datagramaren igorleari. Horretarako behar da jatorrizko makinaren helbidea igorritako datagrametan.

- **Helburuko helbidea**

Eremu hau agertzea ezinbestekoa da edozein sarearte-mailako protokolotan. Ematen duten informazioa nahitaezkoa da sarearte-mailako zerbitzua betetzeko, hau da, datagramak sareartearen mutur batetik beste bateraino helarazteko. Jatorrizko makinan eta bideko bideratzaileetan egindako datagramaren prozesamendua helbide honetan datza. Haren balioa aztertuta ebatziko dute konputagailu horiek nondik bideratu behar duten datagrama.

IP datagram format (Wikipedia)

- **Datagramaren identifikazioa**

IP datagrama bat sareko pakete batean sartzeko handiegia baldin bada, zatitu egin behar da. Zati guztiek jatorrizko datagramaren identifikazioa eramango dute. Horrela helburuko konputagailuak zati guztiak bil ditzake.

- **Desplazamendua**

Eremu honek zati honen kokapena jatorrizko datagraman adierazten du.

- **Iraupena edo TTL (ingelesez: Time To Live)**

Eremu honi balio bat ematen zaio jatorrizko [konputagailuan](#), eta bideko [bideratzaile](#) bakoitzak 1 kentzen dio, gutxienez; eremuaren balioa 0-raino heltzen bada, bideratzaileak datagrama ezabatuko du, inora birbidali gabe. Mekanismo honen helburua da datagrama galduak edo oso atzeratuak saretik kentzea (adibidez, bideratze-errore bat badago eta datagramak begizta batean harrapatuta gelditzen badira). Beraz, sareko garbiketarako behar da eremu hau.

- **Goiko protokoloa**

Eremu hau helmugako konputagailuak behar du, eta ez bideko [bideratzaileek](#). Helmugako IP entitateak datagrama nori eman behar dion jakiteko ezinbestekoa da. Hasiera batean, IP mailaren erabiltzailea garraio-maila denez, badirudi argi dagoela nori eman behar zaion: helburuko konputagailuko garraio-mailako entitateari. Baina [TCP/IP eredu](#)ko garraio-mailako entitate bat baino gehiago aurkituko ditugu helburuko konputagailuan. (6 TCP, 17 UDP...)

- **Bit-markak (edo flagak)**

Hiru dira, baina aurrenekoa ez da erabiltzen. Besteak Ez zatitu bita eta Zati gehiago bita dira —ingelesez, Don't Fragment (DF) eta More Fragments (MF)—. Batak bideratzaileei datagrama hori ezin dela zatitu jakinarazteko balio du (aplikazio batzuek horrela beharko dute). Besteak hori ez dela jatorrizko datagramari dagokion azkeneko zatia adierazten dio helburuko IP entitateari.

- **Erroreak atzemateko funtzioa**

Goiburukoari bakarrik ezartzen zaion funtzio matematiko sinplea da. Datagramak bere bidean bisitatuko dituen bideratzaile guztiek [TTL](#) eremuaren balioa aldatuko dutenez, birkalkulatu beharko dute eremu hau. Praktikan, bideratzaileek ez liokete inongo kasurik egin behar eremu honi, zeren gaur egungo sare gehienek IPrena baino askoz indartsuagoak diren erroreak atzemateko funtzioak erabiltzen baitituzte ([CRC](#) funtzioak gehienetan) beren [tramaetan](#), eta, gainera, datagramaren eremu guztiei aplikatzen zaizkie funtzio horiek (ez bakarrik goiburukoari). Beraz, eremu honi kasu egitea denbora galtzea da: txartelak ez lioke IP mailari matxuratuta dagoen datagrama bat pasatuko. Horregatik eremu hau ez da beharrezkoa datagrama batean.

- **Aukerak**

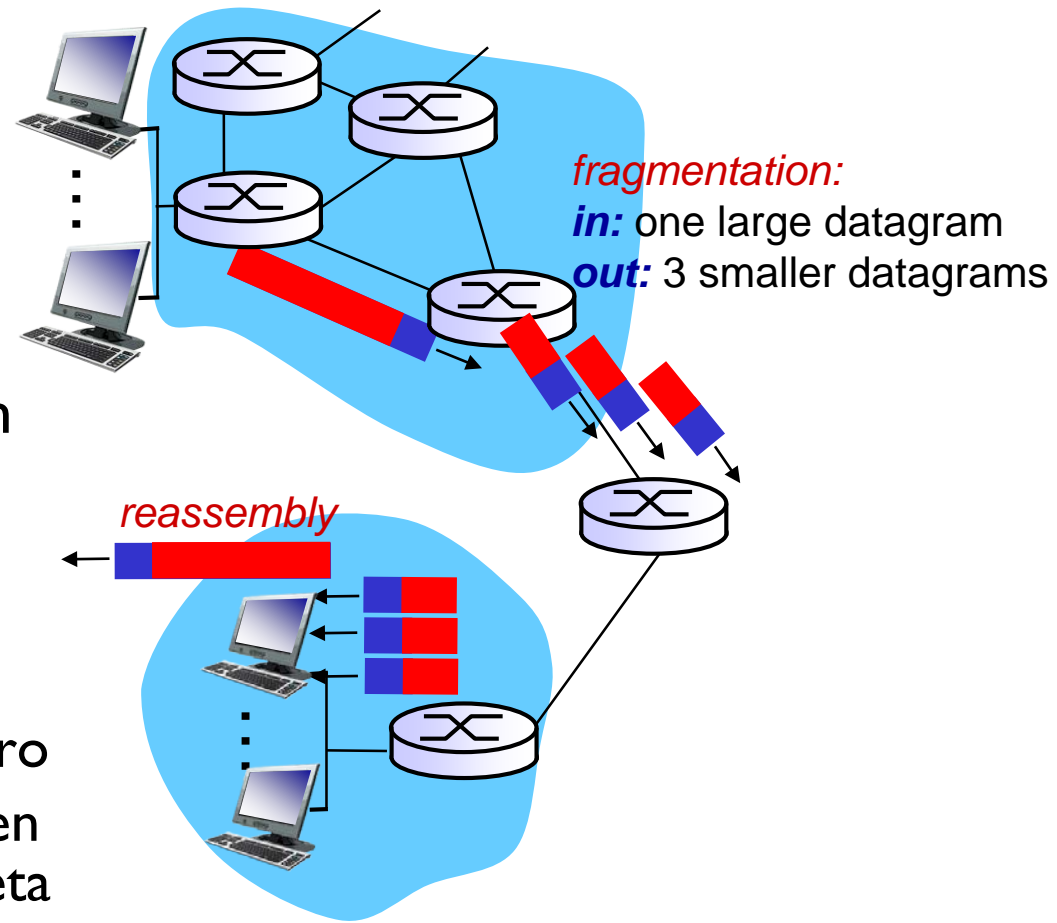
[Bideratzaile](#) askok ez diote kasurik egiten. Protokoloaren ezaugarri berriak frogatzeko sartu zen eremu hau goiburukoan. Gaur egun aukera batzuk daude definituta. Adibidez, eremu honetan datagramak jarraitutako bidea adieraz daiteke (bideko [bideratzaileak](#) hori grabatzeko prest baldin badaude, noski). Horregatik eremu hau ez da beharrezkoa datagrama batean.

Datuen eremua:

Eremu honek datagramaren gehiena okupatzen du. Eremu honetan [konputagailu](#) batek besteari eman nahi dion informazioa dago. Helburuko konputagailuak hau irakurriko du, eta honen arabera dagokion ekintza burutuko du.

IP zatiketa, berreraiketa

- Sareen loturek MTU (max.transfer size) dute
 - Datagramen gehienezko tamaina
 - Lotura desberdinak, MTU desberdinak
- IP datagrama luzeak zatitzen dira (“fragmented”) sarean:
 - Datagrama bat, hainbat datagrama bihurtzen da
 - “reassembled” zatiak helmugan batzen dira berriro
 - IP goiburuko bitak erabiltzen dira zatiak identifikatzeko eta ordenatzeko



IP zatiketa, berreraiketa

adibidea:

- ❖ 4000 byte datagrama
 - ❖ MTU = 1500 bytes
- 20 bytes in header

1480 bytes in
data field

offset =
 $1480/8$

8 byteko multzoetan

	length	ID	fragflag	offset	
	=4000	=x	=0	=0	

*Datagrama luze bat datagrama txikiagoetan
banatzen da*

	length	ID	fragflag	offset	
	=1500	=x	=1	=0	

	length	ID	fragflag	offset	
	=1500	=x	=1	=185	

	length	ID	fragflag	offset	
	=1040	=x	=0	=370	

$$185 * 4 = 1480$$

IP zatiketa, berreraiketa

4820 byteko TCP pakete bat bidali Etherneten (MUT =1500 byte)

IP goiburua

TCP goiburua

TCP datuak



- ID= 12345
- Flag MF =1
- Desplazamiento Fragmento= **0**
- Longitud =1500



- ID= 12345
- Flag MF =1
- Desplazamiento Fragmento= $1480/8 = \mathbf{185}$
- Longitud =1500



- ID= 12345
- Flag MF = **0**
- Desplazamiento Fragmento= $185*2 = \mathbf{370}$
- Longitud = 1080

$$80 = 3*20 \text{ (IP)} + 1*20 \text{ (TCP)}$$

Chapter 4: outline

4.1 Sare geruza, gainbegirada bat

- Informazio planoa
- Kontrol planoa

4.2 Zer dago Router batean?

4.3 IP: Internet Protocol

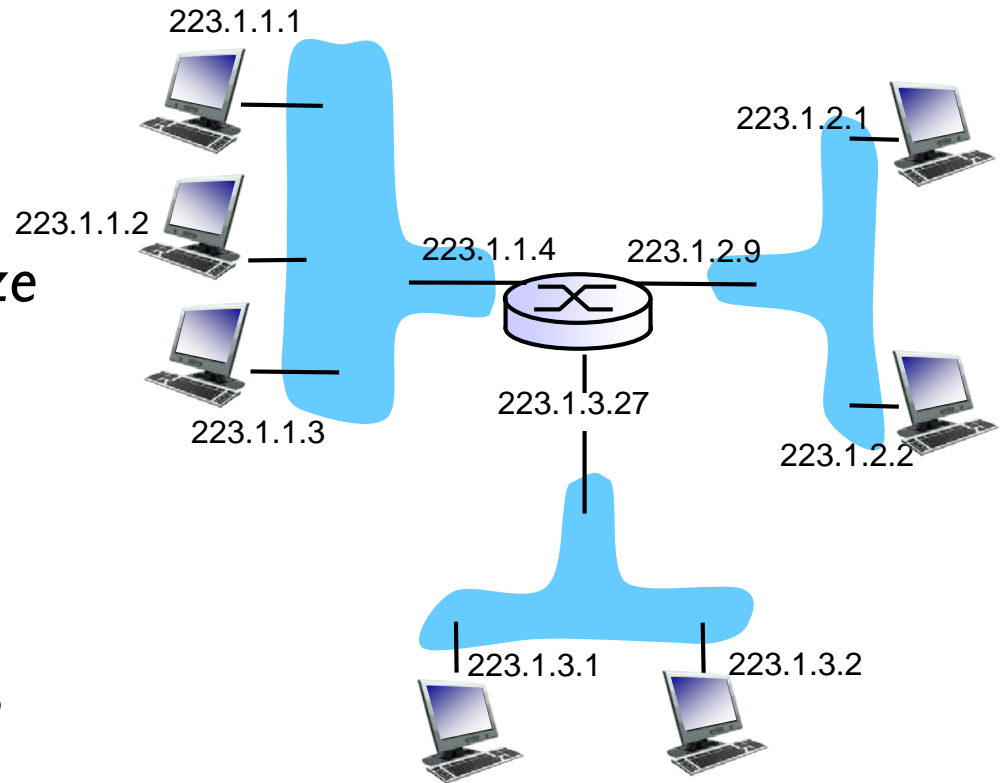
- datagramen formatua
- zatikaketa
- **IPv4 helbideraketa**
- network address translation (NAT)
- IPv6

4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

IP addressing: sarrera

- **IP helbideak:** 32-bit identifikatzailea host-entzat, bideratze *interfaze*
- **interfaze:** Host/router konexioa lotura fisikoarekin
 - Roterrek interfaze bat baino gehiago dute
 - Host-ek interfaze bat edo bi dute (e.g., wired Ethernet, wireless 802.11)
- **IP helbideak interfaze bakoitzarekin lotuta daude**

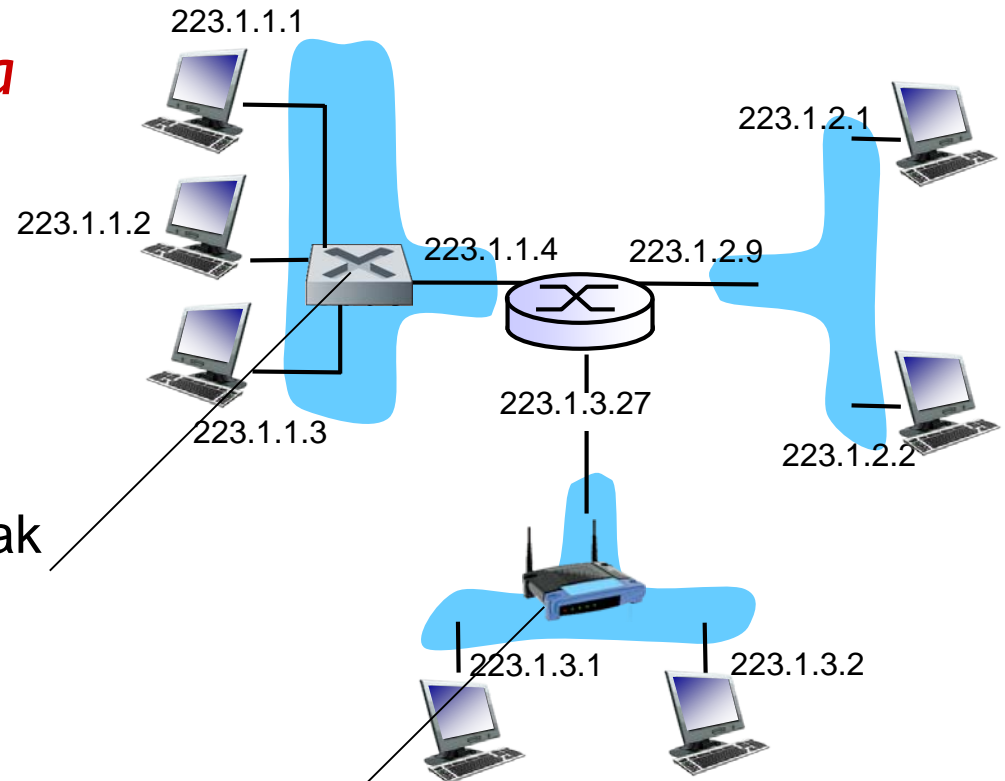


$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_1$$

IP addressing: sarrera

Q: nola konektatzen dira interfazeak?

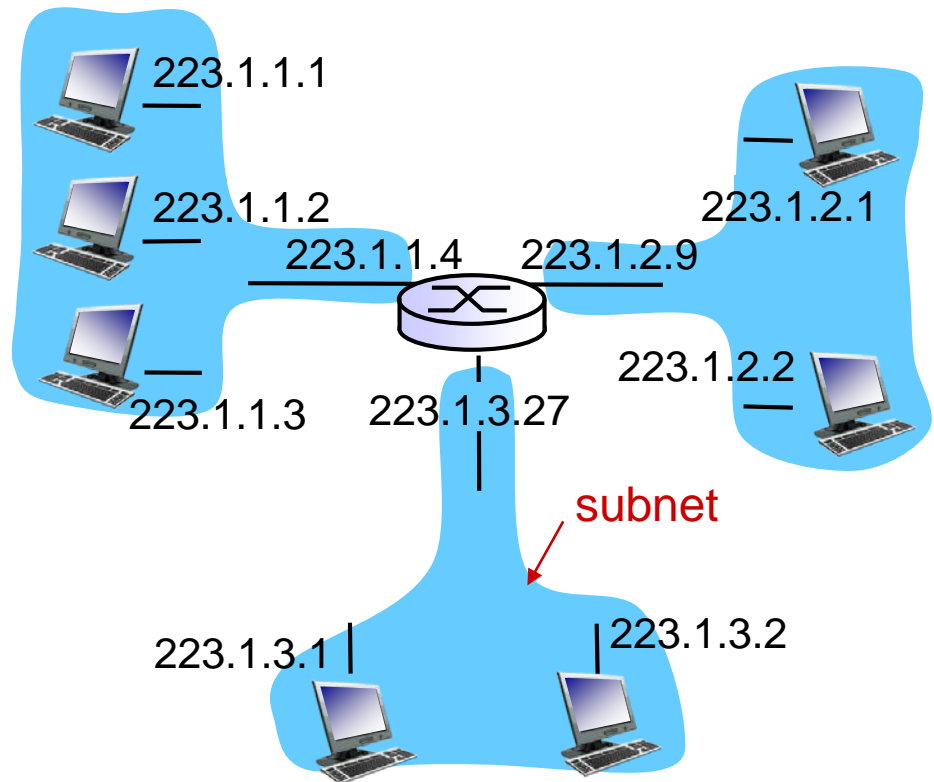
A: Harizko Ethernet interfazeak
Ethernet switch-en bidez
konektatuta



A: Haririk gabeko WiFi interfazeak
WiFi puntura konektatuta

Azpisareak

- IP helbideak:
 - azpisarea - lehen bitak
 - Host-ak - bukaera
- *Zer da azpisare bat?*
 - Sareko IP helbide zati bera duten elementuak (elementuen interfazeak)
 - Elkarren artean konekta daitezke *router bat erabiligabe*

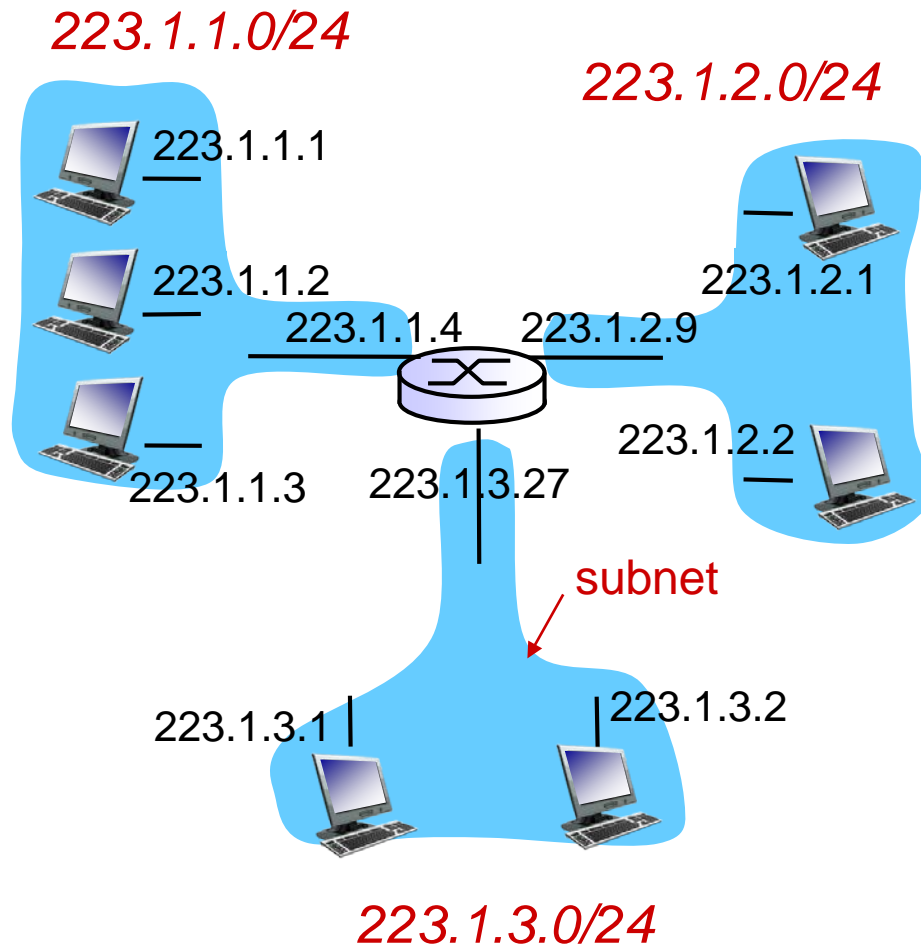


Hiru azpisareko sarea

Azpisareak

errezeta

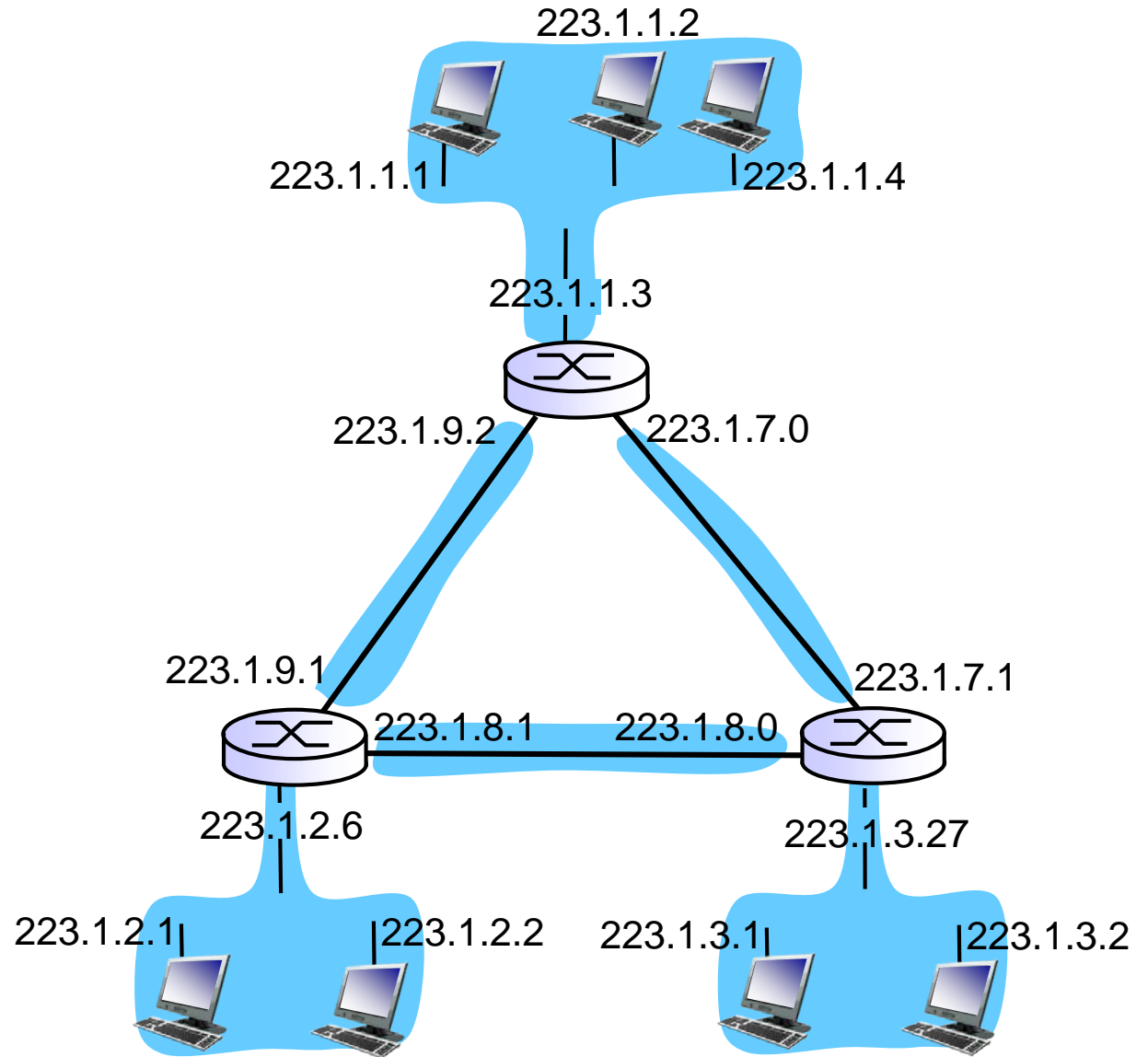
- Azpisare bakoitza mugatzeko, interfazeak, interfaceak eta host/router-ak banandu behar dira, isolatutako azpisareen irlak sortuz
- Isolatutako sare bakoitza, *azpisare bat* da



subnet mask: /24

Subnets

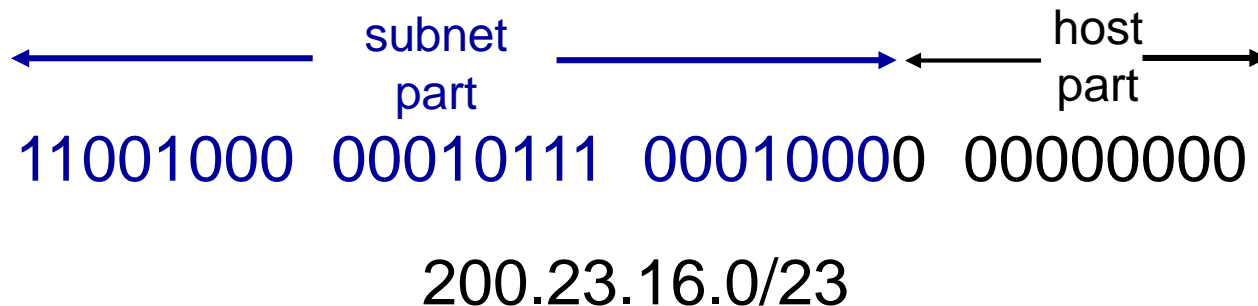
Zenbat?



IP addressing: CIDR

CIDR: Classless InterDomain Routing

- Helbidearen azpisarearen zatia (definitu gabekoa)
- Helbideen formatua: **a.b.c.d/x**, non x helbidearen azpisarearen zatiaren bit kopurua den



IP addressing

Zortzikotea(0-3)	0					1	2	3	Formatua
Bytak (0-31)	0	1	2	3	4-7	8-15	16-23	24-31	
A-mota	0	Sarea					Gailua		N.H.H.H
B-mota	1	0	Sarea				Gailua		N.N.H.H
C-mota	1	1	0	Sarea			Gailua		N.N.N.H
D-mota	1	1	1	0	Sarea		Gailua		erreserbatuta
E-mota	1	1	1	1	1	Sarea		Gailua	esperimentalak

Helbide bereziak:

- 10.0.0.0/8
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255
- 169.254.0.0/24

A-motakoa, helbide pribatuak
 B motakoak, helbide pribatuak
 C-motakoak, helbide pribatuak
 B-motakoa, helbide pribatua

- 224.0.0.0 – 224.255.255.255
- 224.0.0.0 – 224.255.255.254

D-mota, multicast helbideak
E-mota, experimentalak

- 0.0.0.0
- 127.0.0.1

Bidaltzen duen gailua (RARP, DHCP)
 loopback, localhost

IP addressing

Maskara bitarra	Dezimala	CIDR	Host #
11111111.00000000.00000000.00000000	255.0.0.0	/8	16777216
11111111.10000000.00000000.00000000	255.128.0.0	/9	8388608
11111111.11000000.00000000.00000000		/10	4194304
11111111.11100000.00000000.00000000	255.224.0.0	/11	
11111111.11110000.00000000.00000000		/12	1048576
11111111.11111000.00000000.00000000	255.248.0.0	/13	524288
	255.252.0.0	/14	262144
11111111.11111110.00000000.00000000	255.254.0.0	/15	131072
11111111.11111111.00000000.00000000	255.255.0.0	/16	
11111111.11111111.11000000.00000000			32768
11111111.11111111.11100000.00000000	255.255.192.0	/18	16384
	255.255.224.0	/19	8192
11111111.11111111.11111000.00000000		/20	4096
	255.255.248.0	/21	2048
11111111.11111111.11111110.00000000	255.255.252.0	/22	
11111111.11111111.11111111.00000000		/23	512
11111111.11111111.11111111.10000000	255.255.255.0	/24	256
11111111.11111111.11111111.11000000	255.255.255.128	/25	128
11111111.11111111.11111111.11110000	255.255.255.224	/27	32
	255.255.255.240	/28	16
11111111.11111111.11111111.11111100	255.255.255.248		8
11111111.11111111.11111111.11111110		/30	

IP helbideak: nola lortu?

Q: Nola lortzen du *host* batek beraren IP helbidea?

- Systemaren administratzaileak fitxategi batean definituta
 - Windows: control-panel->network->configuration->tcp/ip->properties
 - UNIX: /etc/rc.config
- **DHCP: D**ynamic **H**ost **C**onfiguration **P**rotocol:
Era dinamikoan lortzen da
 - “plug-and-play”

DHCP: Dynamic Host Configuration Protocol

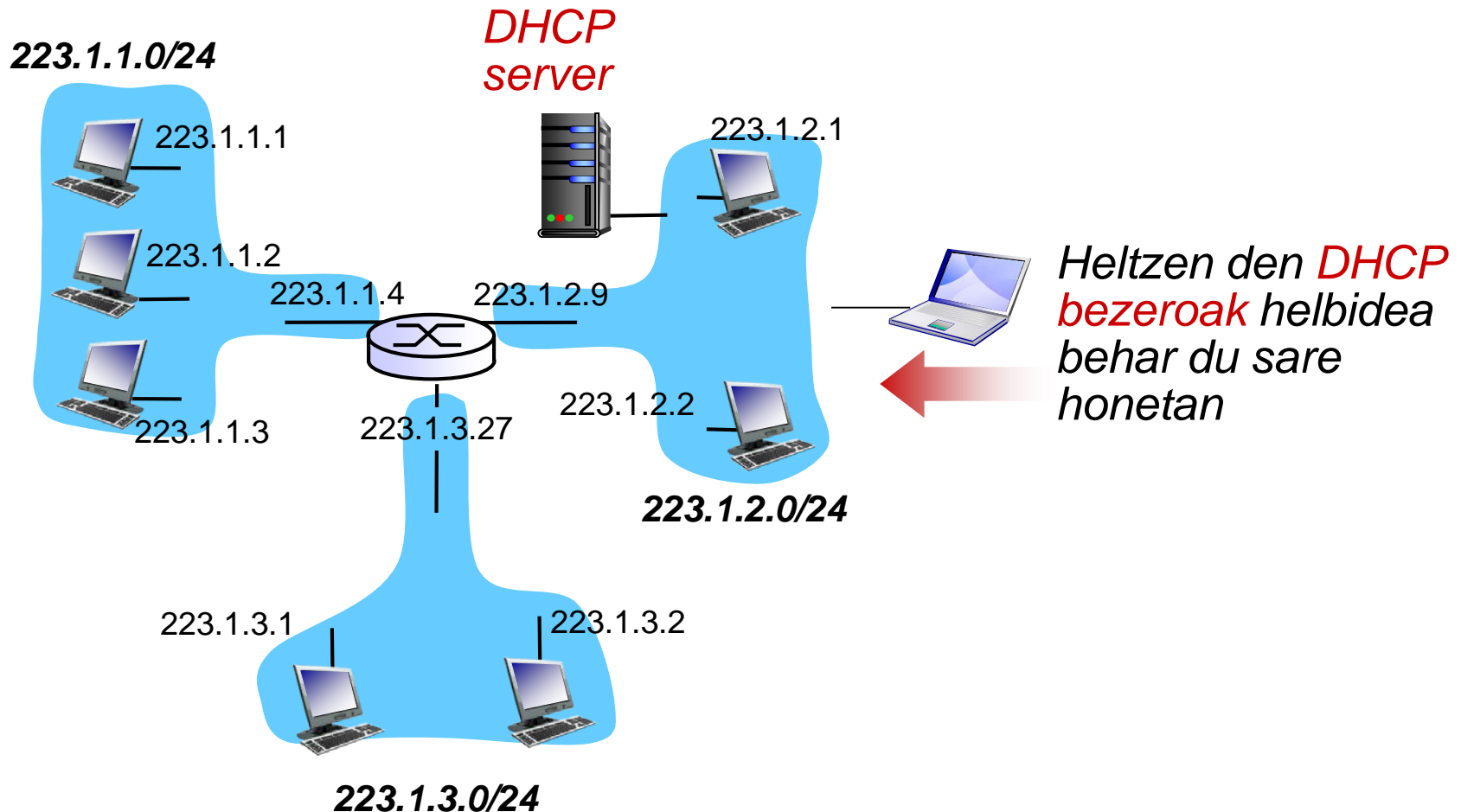
Helburua: Host-ek era dinamikoan lortzen dute haien IP helbidea sare zerbitzari batetik

- Unean erabiltzen diren helbideak mantentzen dira
- Helbideak bererabil daitezke (helbideak mantentzen dira host-ak konektatuta dauden bitartean /“on”)
- Aldizkako erabiltzaileentzako sarera konektatzeko era egokia (ez da IP finko bat behar)

DHCP, gainbegirada:

- host “**DHCP discover**” mezua broadcast-en bidaltzen du [optional]
- DHCP zerbitzariak “**DHCP offer**” mezuarekin erantzuten du [optional]
- host IP helbidea eskatzen du: “**DHCP request**” mezua
- DHCP zerbitzariak IP helbidea bidaltzen du: “**DHCP ack**” mezua

DHCP client-server scenario



DHCP client-server scenario

DHCP server: 223.1.2.5

DHCP discover

arriving
client



Broadcast: is there a
DHCP server out there?

DHCP offer

Broadcast: I'm a DHCP
server! Here's an IP
address you can use

DHCP request

Broadcast: OK. I'll take
that IP address!

DHCP ACK

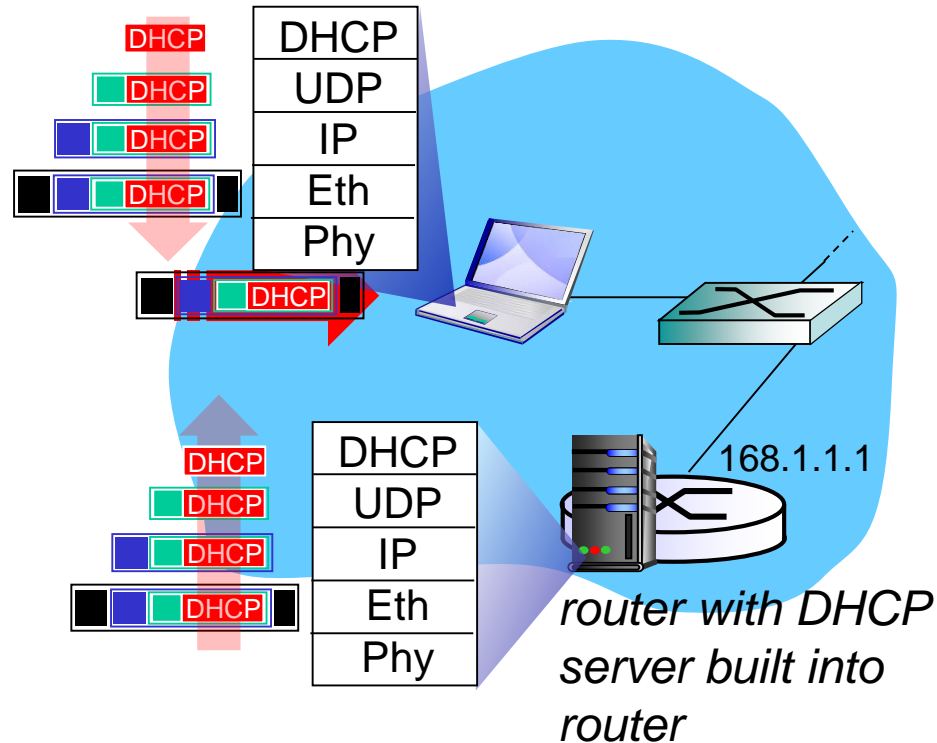
Broadcast: OK. You've
got that IP address!

DHCP: IP-a baino gehiago

DHCP-ek azpisarearen helbideaz gain, ondoko informazioa ere eman dezake:

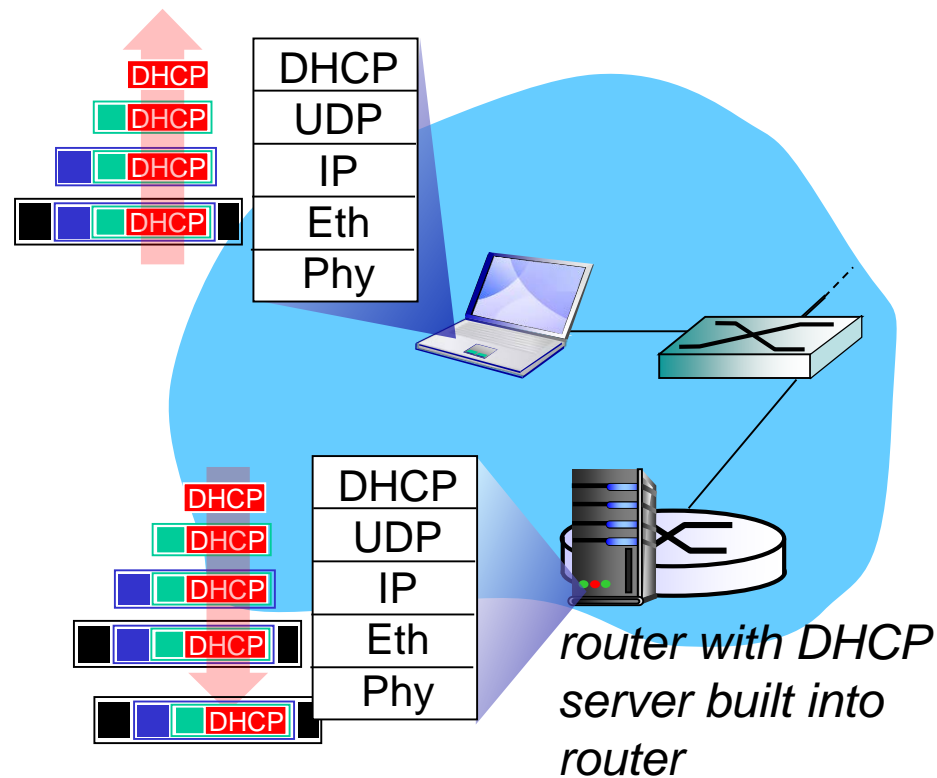
- Gateway
- DNS zerbitzariaren izen eta IP helbidea
- Sarearen maskara (network versus host portion of address)

DHCP: adibidea



- Konektatzen den portatila IP helbidea, gateway-a eta DNS zerbitzaria: DHCP erabiltzen du
- DHCP eskaera UDP-n kapsulatuta, IP-n kapsulatuta, 802.1 Ethernet-n kapsulatuta
- Ethernet broadcast (dest: FFFFFFFF) eskaria LAN-ean, DHCP zerbitzariak jasota
- KAPsulaketa desegiten da: Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

DHCP: adibidea



- DCP zerbitzariak DHCP ACK prestatzen du bezeroaren IP helbidea, DNSren IP-a (eta izena) eta Gateway-aren IP-arekin
- DHCP zerbitzariak informazioa kapsulatzen du. Informazioa bezeroari bidaltzen zaio, bezeroan DHCP informazioa demultiplexatzen da
- Bezeroak beraren IP helbidea, DNSren IP-a eta Gateway-aren IP-a ezagutzen ditu

DHCP: Wireshark output (home LAN)

Message type: **Boot Request (1)**

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

Transaction ID: 0x6b3a11b7

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

Client IP address: 0.0.0.0 (0.0.0.0)

Your (client) IP address: 0.0.0.0 (0.0.0.0)

Next server IP address: 0.0.0.0 (0.0.0.0)

Relay agent IP address: 0.0.0.0 (0.0.0.0)

Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)

Server host name not given

Boot file name not given

Magic cookie: (OK)

Option: (t=53,l=1) **DHCP Message Type = DHCP Request**

Option: (61) Client identifier

Length: 7; Value: 010016D323688A;

Hardware type: Ethernet

Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)

Option: (t=50,l=4) Requested IP Address = 192.168.1.101

Option: (t=12,l=5) Host Name = "nomad"

Option: (55) Parameter Request List

Length: 11; Value: 010F03062C2E2F1F21F92B

1 = Subnet Mask; 15 = Domain Name

3 = Router; 6 = Domain Name Server

44 = NetBIOS over TCP/IP Name Server

.....

request

Message type: **Boot Reply (2)**

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

Transaction ID: 0x6b3a11b7

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

Client IP address: 192.168.1.101 (192.168.1.101)

Your (client) IP address: 0.0.0.0 (0.0.0.0)

Next server IP address: 192.168.1.1 (192.168.1.1)

Relay agent IP address: 0.0.0.0 (0.0.0.0)

Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)

Server host name not given

Boot file name not given

Magic cookie: (OK)

Option: (t=53,l=1) DHCP Message Type = DHCP ACK

Option: (t=54,l=4) Server Identifier = 192.168.1.1

Option: (t=1,l=4) Subnet Mask = 255.255.255.0

Option: (t=3,l=4) Router = 192.168.1.1

Option: (6) Domain Name Server

Length: 12; Value: 445747E2445749F244574092;

IP Address: 68.87.71.226;

IP Address: 68.87.73.242;

IP Address: 68.87.64.146

Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

reply

IP addresses: how to get one?

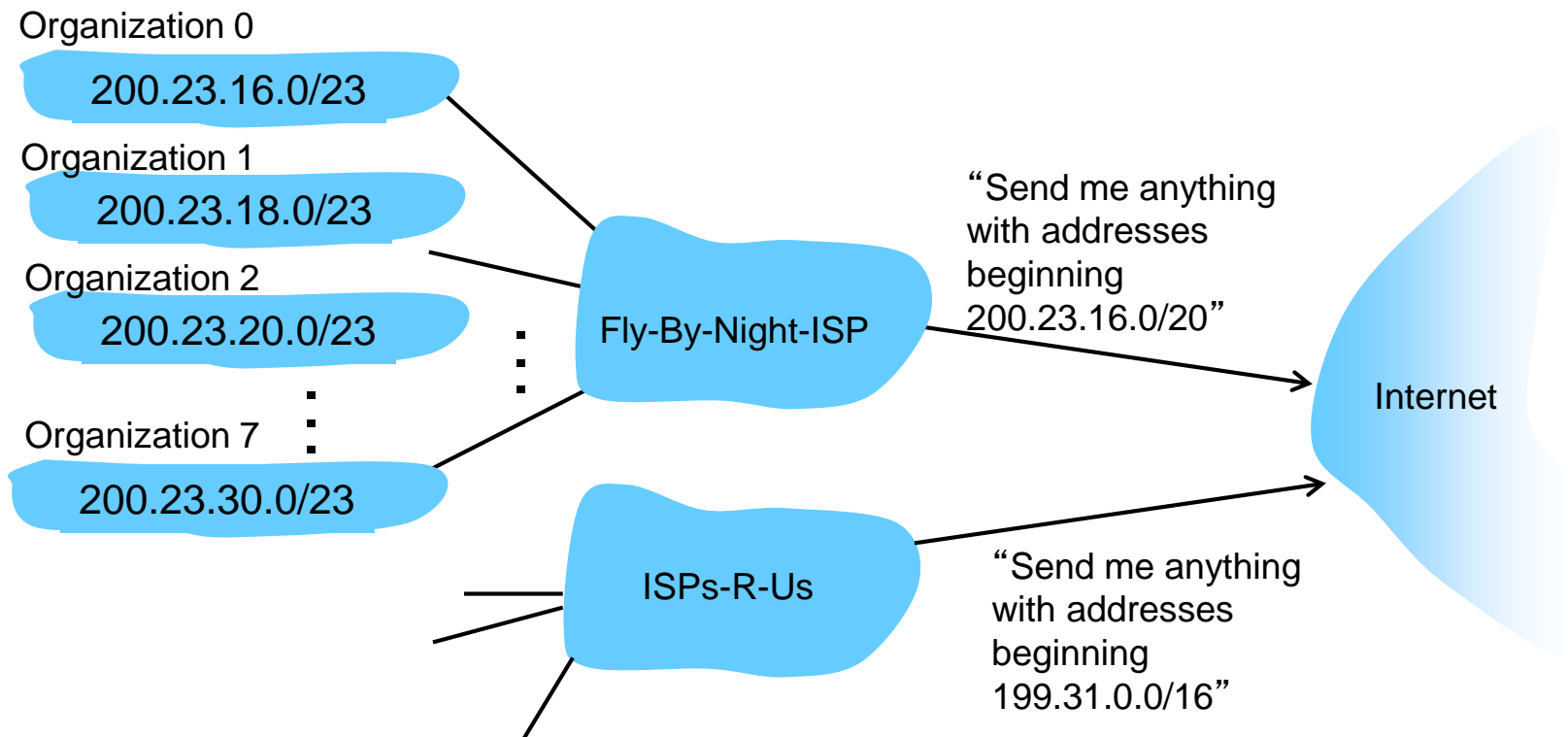
Q: Nola lortzen du sareak, IP helbidearen azpisarearen zatia?

A: ISPren helbide tartearen zatia lortzen da

ISP's block	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organization 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organization 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organization 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...
Organization 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

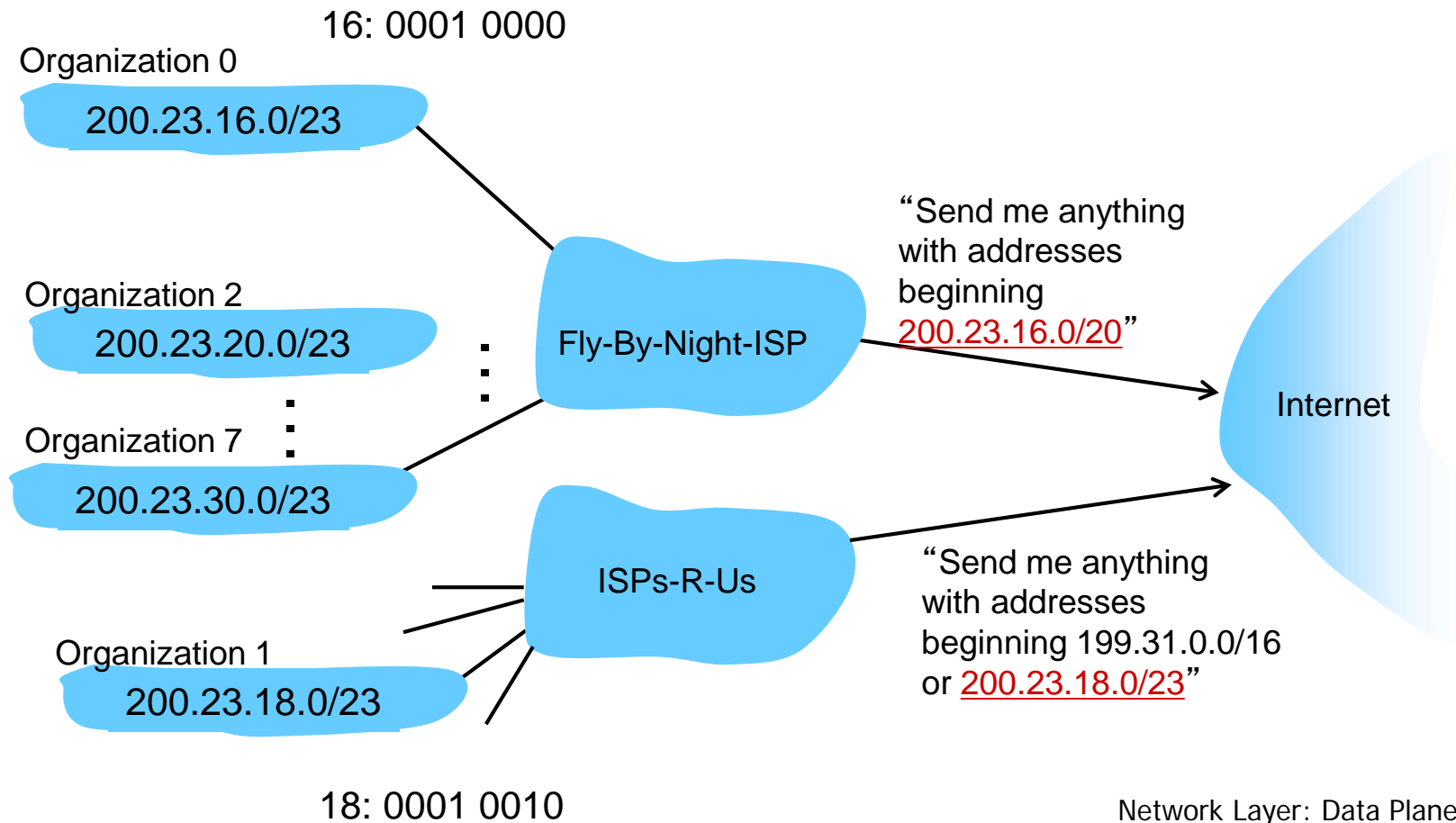
Helbide banaketa hierarkikoa: bide gehiketa

Helbide banaketa hierarkikoak, bideraketa eraginkorra ahalbidetzen du



Helbide banaketa hierarkikoa: biderik espezifikoa

ISPs-R-Us-k bide zehatzago du Organization I-era



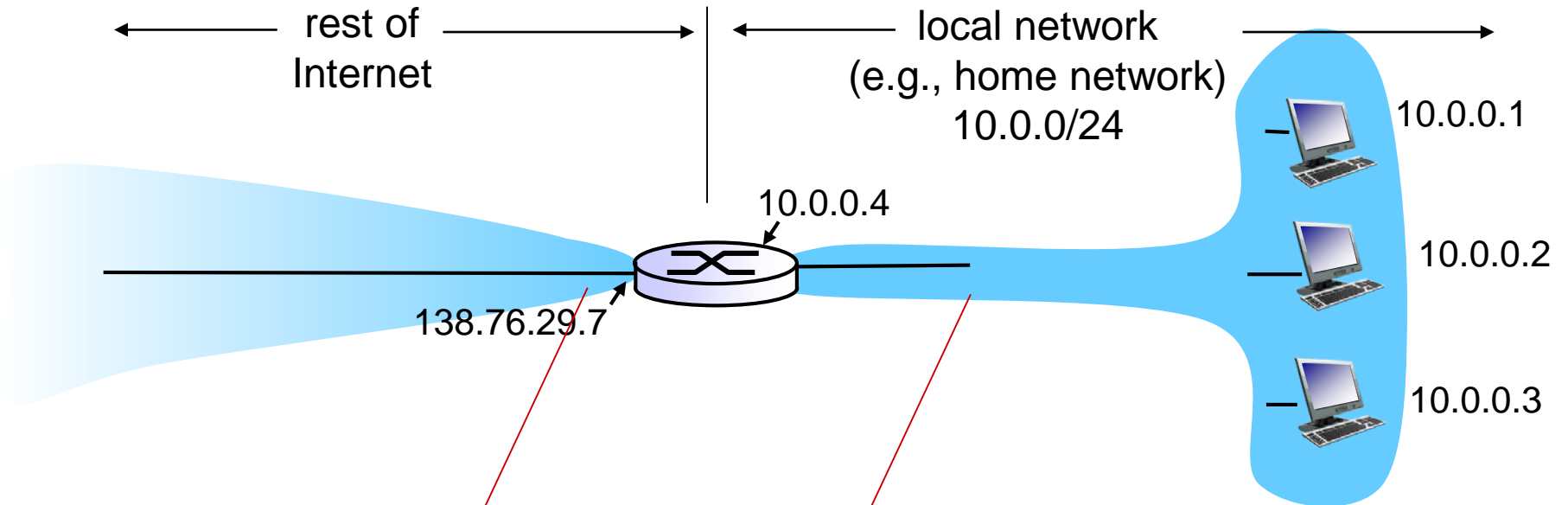
IP helbide banaketa: azken hitza...

Q: Nola lortzen du ISP batek helbide multzo bat?

A: **ICANN:** Internet Corporation for Assigned Names and Numbers <http://www.icann.org/>

- Helbideak banatzen ditu
- DNS-ak kudeatzen ditu
- Domeinu izenak esleitzen ditu, arasoak konpondu

NAT: network address translation



Sare lokala **uzten** duten datagrama **guztiek** NAT IP-ren helbide **bera** dute: 138.76.29.7, baina igorle portu desberdinak

Sare honetan mantentzen diren datagramak 10.0.0/24 helbidea dute

NAT: network address translation

Zergatia: Sare lokalak IP helbide bakarra erabiltzen du mudura ateratzeko:

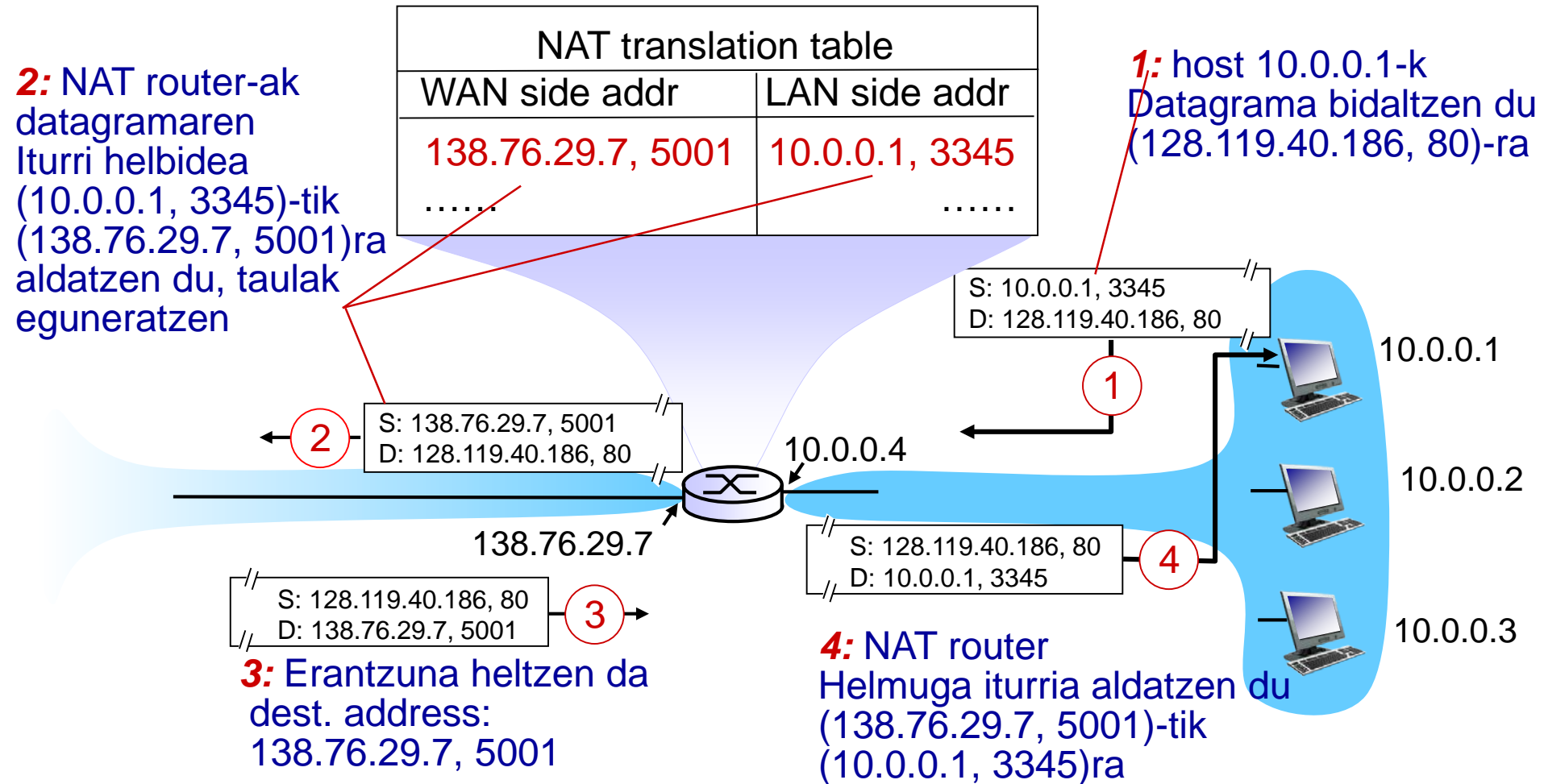
- ISP-ak ez du IP helbide multzo bat eman behar, irteerako IP helbide bakarra baizik (ekipo guztientzat)
- Barneko sareko helbideak alda daitezke kanpoko inor enteratu barik
- ISP-a alda daiteke helbideak aldatu gabe
- Azpisare barnean dauden ekipoak izin dira helbidez atzitu kanpotik (a security plus)

NAT: network address translation

implementazioa: NAT router must:

- *Ateratzen diren datagramak: ordezkatu* (source IP address, port #) ateratzen den datagrama guztietan (NAT IP address, new port #) . . . Urruneko bezero/zerbitzariak erantzungo dute (NAT IP address, new port #) erabiliz
- *gogoratu (NAT itzulpen taulan):* (source IP address, port #) - (NAT IP address, new port #) itzulpen bikoteak
- *Sartzen diren datagramak: aldatu* (NAT IP address, new port #) sarrerako datagramen “helmuga eremuetan” eta dagokion (source IP address, port #) erabili (NAT taulan gordeta)

NAT: network address translation



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

NAT: network address translation

- 16-bit port-number field:
 - Aldibereko 60.000 konexio IP helbide bakarra erabilita!
- NAT is controversial:
 - routers 3 maila arte baino ez dute prozesatzen
 - Helbide falta IPv6rekin konpondu beharko litzateke
 - violates end-to-end argument
 - NAT erabiltzeko aukera kontutan hartu behar da app garatzaileengatik, e.g., P2P applications
 - NAT traversal: zer gertatzen da zerbitzari bat NAT baten atzean badago?

Chapter 4: outline

4.1 Sare geruza, gainbegirada bat

- Informazio planoa
- Kontrol planoa

4.2 Zer dago Router batean?

4.3 IP: Internet Protocol

- datagramen formatua
- zatikaketa
- IPv4 helbideraketa
- network address translation (NAT)
- IPv6

4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

IPv6: motivation

- *Hasierako arrazoia:* hasierako 32-bit helbideak berehala bukatuko dira (azpaldi esanda).
- Beste arrazoiak:
 - Goiburuaren formatoa bideraketa abiadura azkartzen du
 - Goiburuaren aldaketak QoS errazten du

IPv6 datagram format:

- Luzera finkoko 40 byte-eko goiburua
- Ez da zatiketarik onartzen

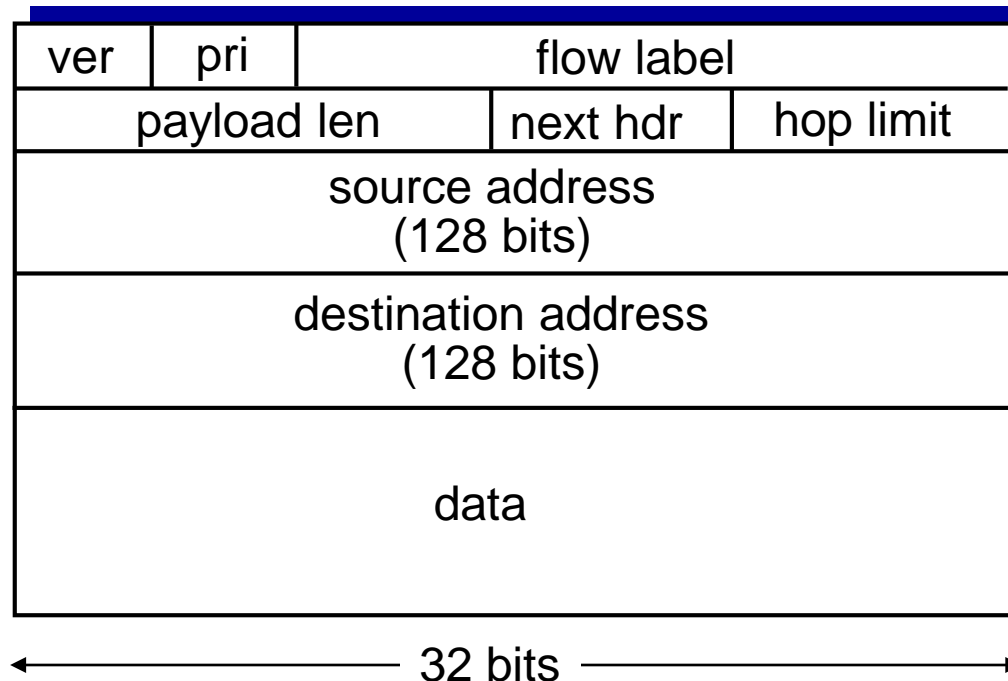
IPv6 datagram format

priority: identify priority among datagrams in flow

flow Label: identify datagrams in same “flow.”

(concept of “flow” not well defined).

next header: identify upper layer protocol for data



IPv6 datagramaren formatua

0										10										20										30						
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1					
Bertsioa				Lehentasuna					Fluxuaren etiketa																											
Edukiaren luzera															Hurrengo goiburua										Jauziaren muga											
Iturburuko helbidea (128 bit =16 byte)																																				
...																																				
...																																				
...																																				
Helburuko helbidea (128 bit =16 byte)																																				
...																																				
...																																				
...																																				

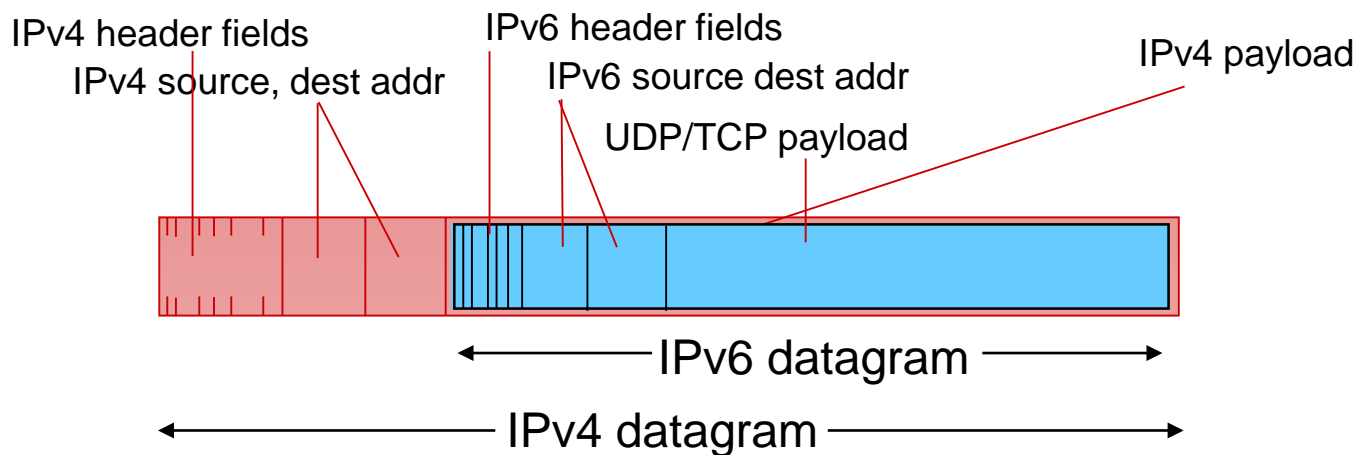
- Bertsioa.** Bideratzaileek IP paketearen zein bertsio prozesatzen ari diren jakiteko
- Lehentasuna.** Kontrol-fluxua egiteko ala ez egiteko
- Fluxuaren etiketa.** Iturburuaren eta helburuaren arteko eskakizun bereziak adosteko. Oraindik probatan.
- Edukiaren luzera.** Goiburukoaren ondoren zenbat byte datozen adierazteko.
- Hurrengo goiburukoa.** Goiburuko osagarriak adierazten ditu. IPren azken goiburukoa bada, eremu honetan garraio-mailari zein protokolo erabili behar duen adierazten dio. 1 ICMP, 2 IGMP, 6 TCP eta 17 UDP.
- Jauzien muga.** IP4bren iraupena eremuaren parekoa da.
- Iturburuko eta helburuko helbideak**

Other changes from IPv4

- *checksum*: kenduta, prozesatzen abiadura arintzeko
- *options*: baimendutak, baina goiburutik kanpo, “Next Header” eremuaren bidez eskuragarri
- *ICMPv6*: new version of ICMP
 - Mezu mota gehiago, e.g. “Packet Too Big”
 - Taldeko multicast onartzen du

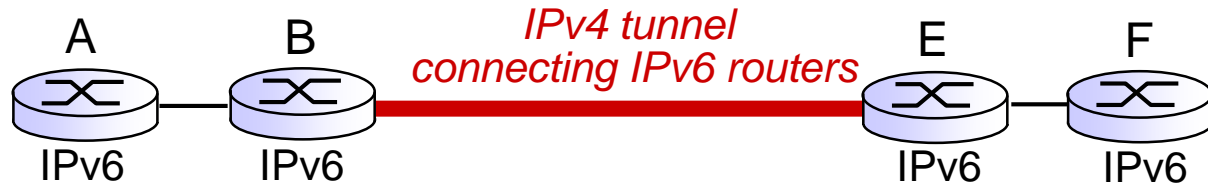
Transition from IPv4 to IPv6

- Ezindaitezke router guztiak batera eguneratu
 - no “flag days”
 - Nola fuuntziona daiteke sarea IPv4 eta IPv6 router nahasketarekin?
- *tunneling*: IPv6 datagrama informazio (*payload*) bezalain IPv4 datagrametan IPv4 router-en zehar

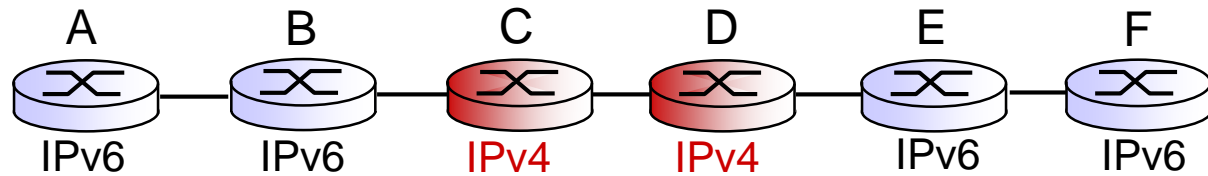


Tunneling

logical view:

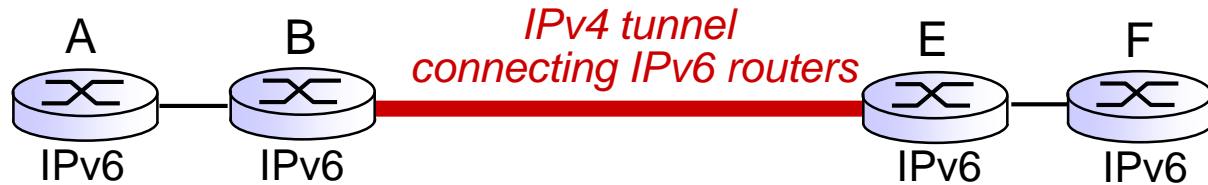


physical view:

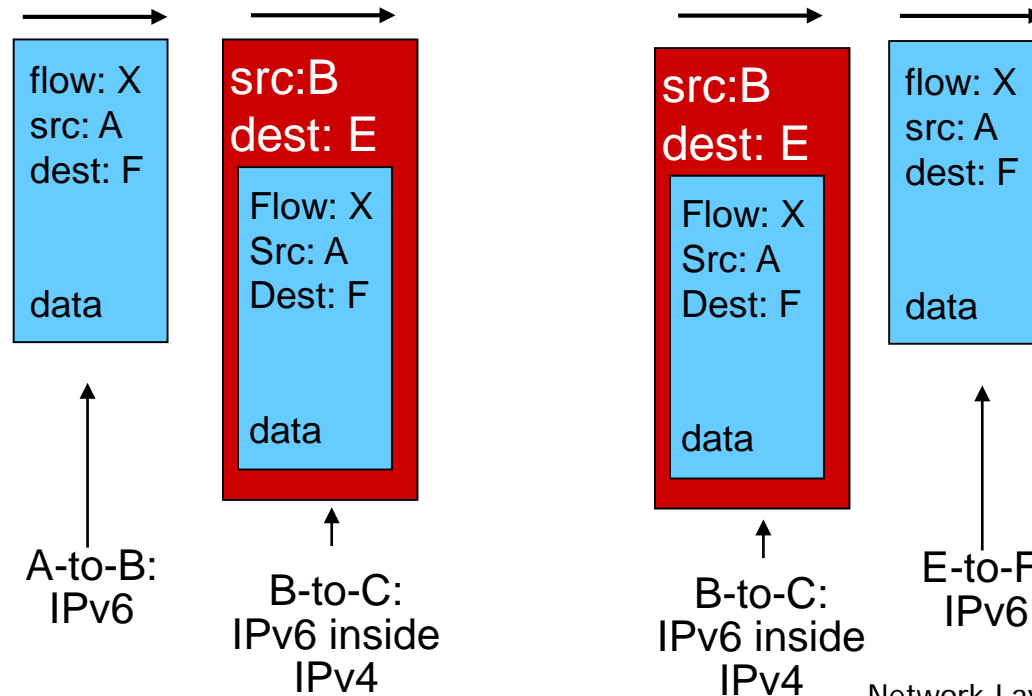
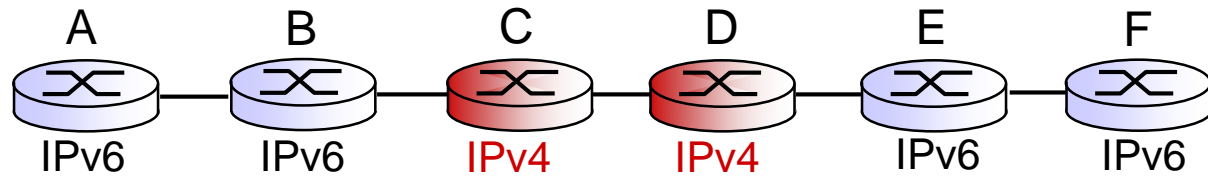


Tunneling

logical view:



physical view:



IPv6: adoption

- Google: 8% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable
- *Long (long!) time for deployment, use*
 - 20 years and counting!
 - think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, ...
 - *Why?*

Chapter 4: outline

4.1 Sare geruza, gainbegirada bat

- Informazio planoa
- Kontrol planoa

4.2 Zer dago Router batean?

4.3 IP: Internet Protocol

- datagramen formatua
- zatikaketa
- IPv4 helbideraketa
- network address translation (NAT)
- IPv6

4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

(bukaeran, informazioa)

Chapter 4: done!

4.1 Overview of Network layer: data plane and control plane

4.2 What's inside a router

4.3 IP: Internet Protocol

- datagram format
- fragmentation
- IPv4 addressing
- NAT
- IPv6

4.4 Generalized Forward and SDN

- match plus action
- OpenFlow example

Question: how do forwarding tables (destination-based forwarding) or flow tables (generalized forwarding) computed?

Answer: by the control plane (next chapter)

Chapter 4: outline

4.1 Overview of Network layer

- data plane
- control plane

4.2 What's inside a router

4.3 IP: Internet Protocol

- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

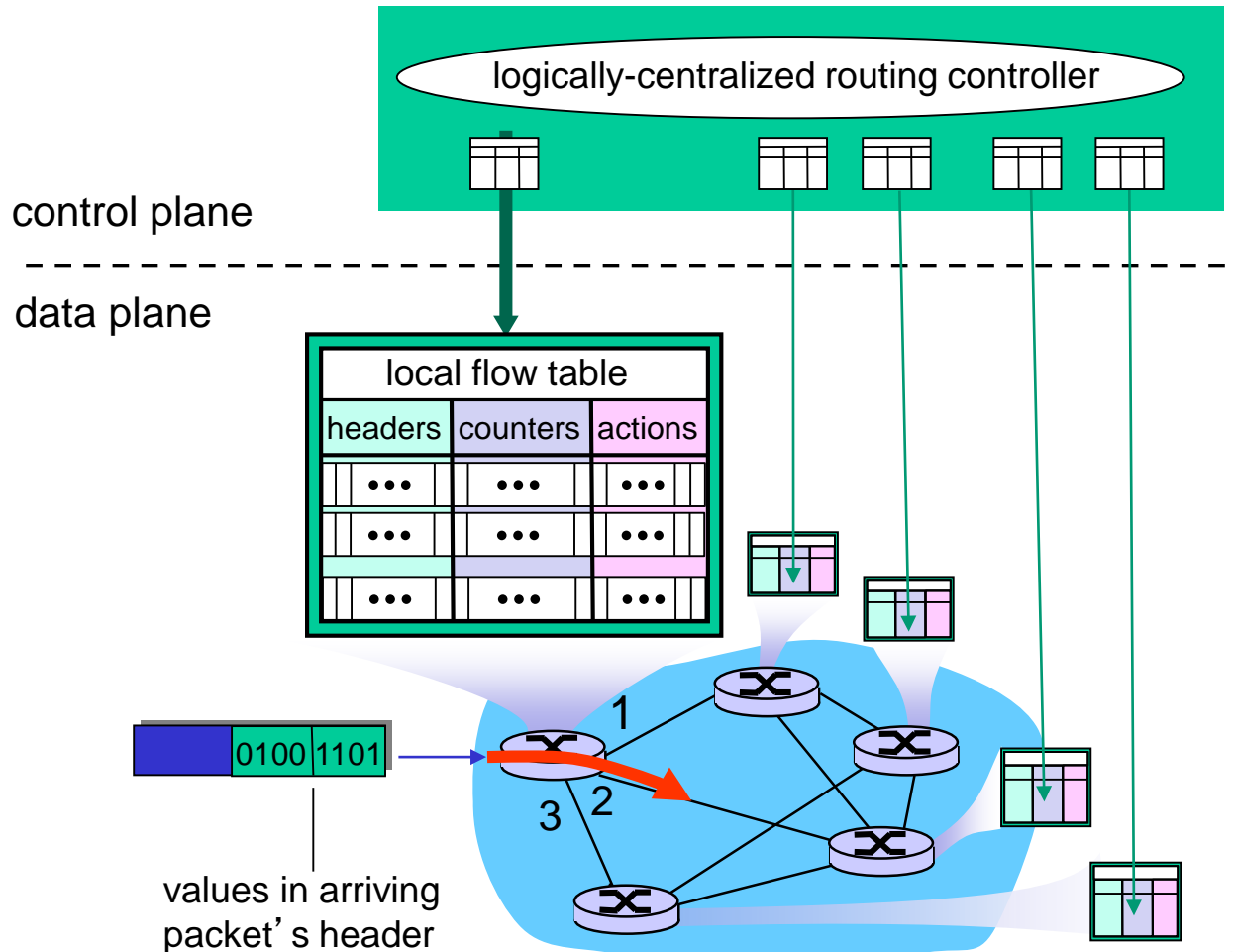
4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

(bukaeran, informazioa)

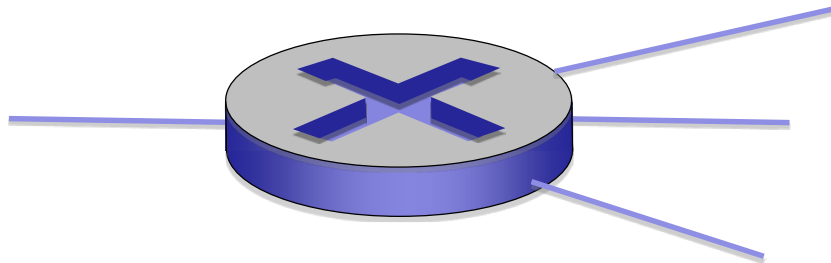
Generalized Forwarding and SDN

Each router contains a *flow table* that is computed and distributed by a *logically centralized routing controller*



OpenFlow data plane abstraction

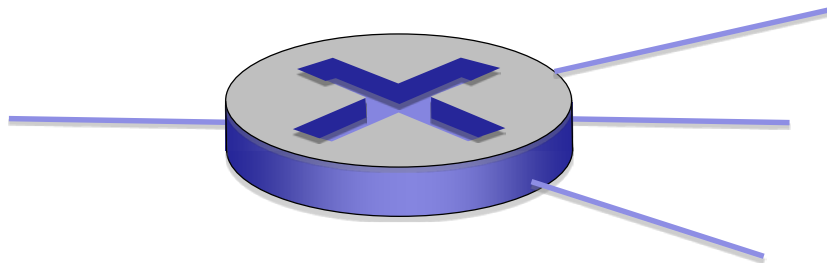
- *flow*: defined by header fields
- generalized forwarding: simple packet-handling rules
 - *Pattern*: match values in packet header fields
 - *Actions: for matched packet*: drop, forward, modify, matched packet or send matched packet to controller
 - *Priority*: disambiguate overlapping patterns
 - *Counters*: #bytes and #packets



Flow table in a router (computed and distributed by controller) define router's match+action rules

OpenFlow data plane abstraction

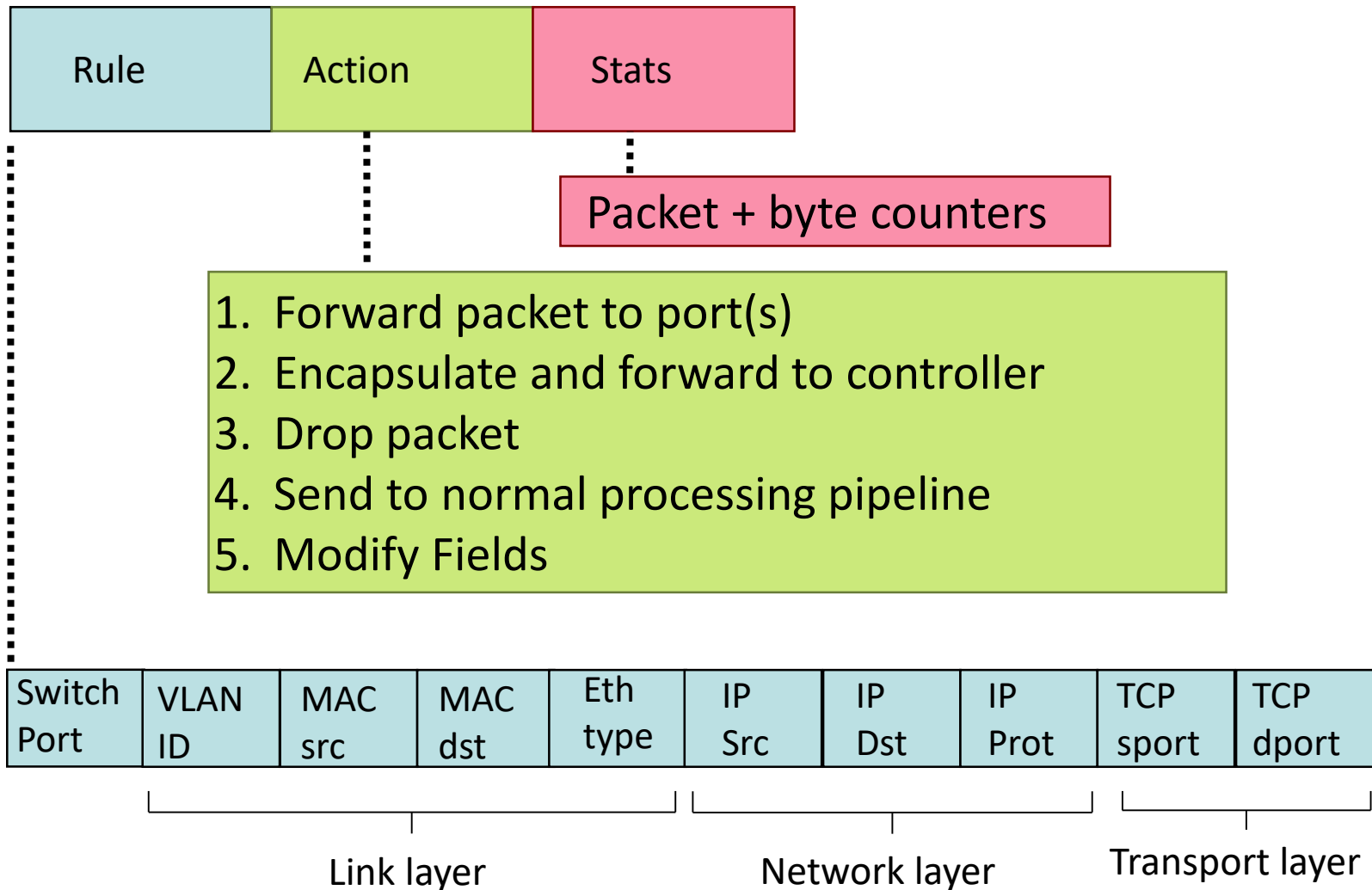
- *flow*: defined by header fields
- generalized forwarding: simple packet-handling rules
 - *Pattern*: match values in packet header fields
 - *Actions: for matched packet*: drop, forward, modify, matched packet or send matched packet to controller
 - *Priority*: disambiguate overlapping patterns
 - *Counters*: #bytes and #packets



* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

OpenFlow: Flow Table Entries



Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

Examples

Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	port3

*layer 2 frames from MAC address 22:A7:23:11:E1:02
should be forwarded to output port 6*

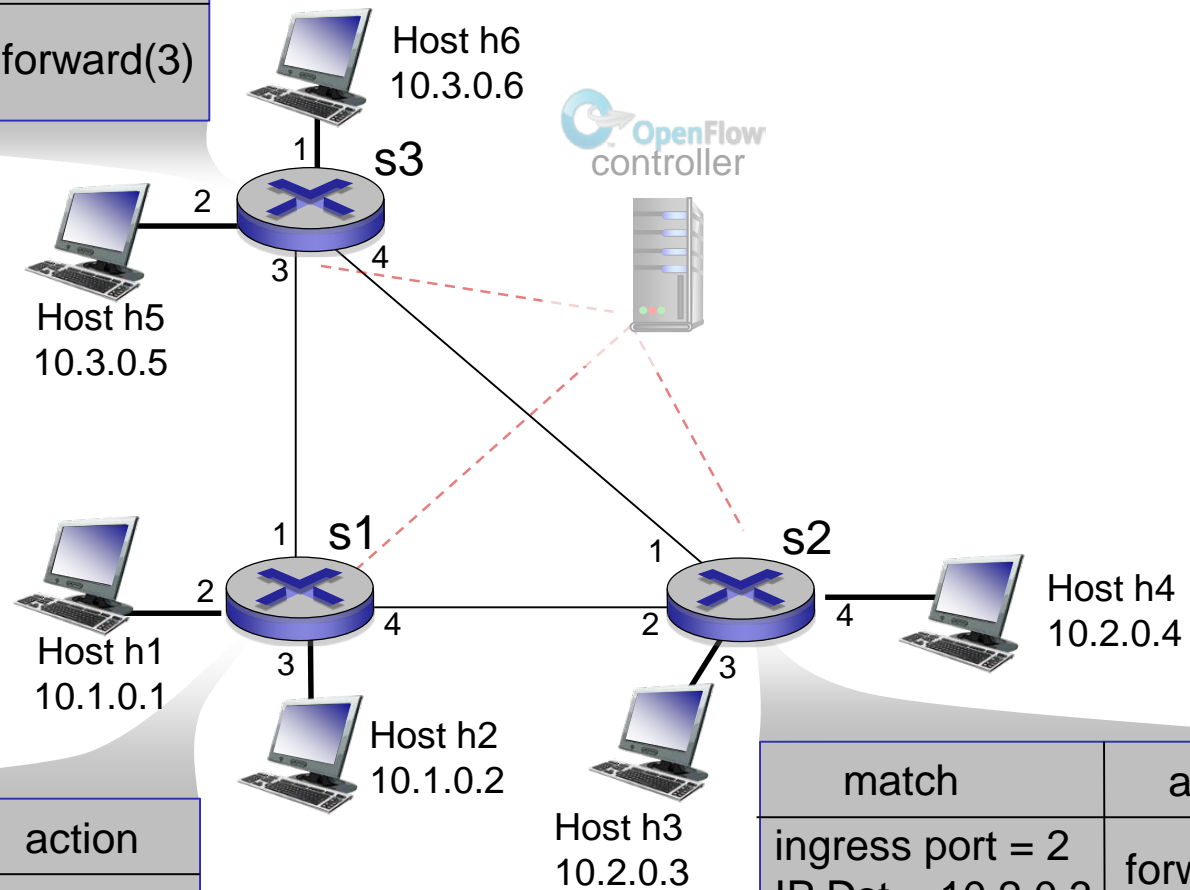
OpenFlow abstraction

- *match+action*: unifies different kinds of devices
- Router
 - *match*: longest destination IP prefix
 - *action*: forward out a link
- Switch
 - *match*: destination MAC address
 - *action*: forward or flood
- Firewall
 - *match*: IP addresses and TCP/UDP port numbers
 - *action*: permit or deny
- NAT
 - *match*: IP address and port
 - *action*: rewrite address and port

OpenFlow example

Example: datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)