Master Course Computer Networks IN2097

Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.
Stephan Günther

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Transport Layer

- continuation -

Roadmap

- Advanced Transport Layer Concepts
- Internet Protocol, The Internet

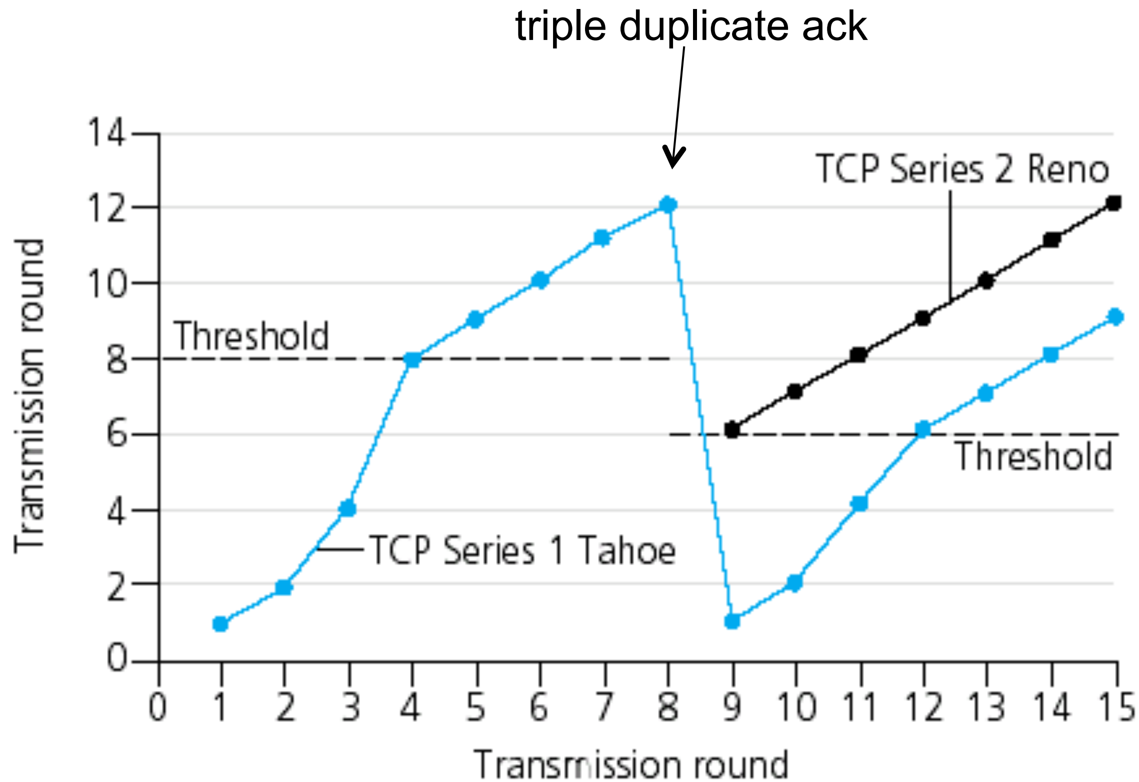
Advanced Transport Layer Concepts

- Buffer bloat
- TCP for high bandwidth long distance connections

- TCP Throughput Formula
- Overview of Deployment of TCP variants
- Detection of TCP-unfriendly Flows
- Multipath TCP

TCP Congestion Control

- Variable Threshold
- At loss event, Threshold is set to 1/2 of CongWin just before loss event



TCP Reno

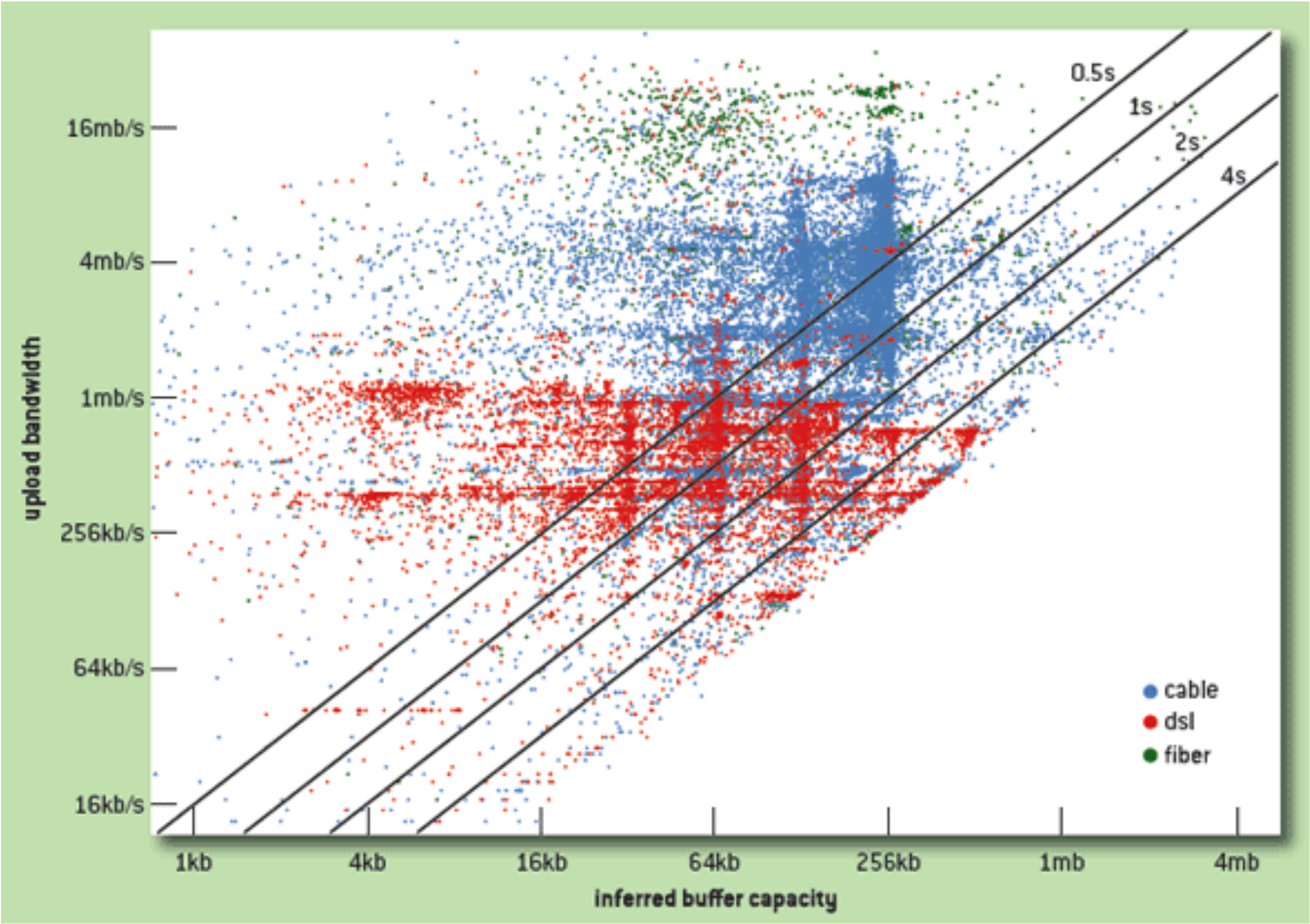
- TCP Fast Recovery algorithm described in RFC 2581
- Implementation introduced 1990 in BSD Reno release
- Behaviour
  - sender only retransmits a packet
    - after a retransmit timeout has occurred
    - or after three duplicate acknowledgements have arrived

    triggering the Fast Retransmit algorithm.
  - a single retransmit timeout might result in the retransmission of several data packets
  - each invocation of the Fast Retransmit algorithm leads to retransmission of only a single data packet
  - problems may arrive when multiple packets are dropped from a single window

TCP and Buffer Bloat

- Capacities of router queues
  - "Large queue = good: Less packet losses at bottlenecks"
  - Do you agree? What would happen to TCP?
- Effects of large buffers at bottleneck on TCP connections
  - Once queues are full: Queueing delays increase significantly or even dramatically
  - TCP congestion control gets no early warning
    - No duplicate ACKS ⇨ no Fast Retransmit
    - Instead: Sudden timeouts
  - Congestion windows way too large
  - Many parallel TCP connections over same link get warning way too late
    - Synchronisation: Oscillation between "All send way too much" and "all get frightened by timeouts and send way too little"
    - Huge variations in queueing delays ⇨ DevRTT becomes very large ⇨ Timeout value becomes very large

Buffer bloat - ICSI Netalyzr Measurements



http://www.broadbandreports.com/shownews/The-ICSI-Netalyzr-Explored-113972

http://www.icir.org/christian/publications/2010-imc-netalyzr.pdf

TCP for High Bandwidth Long Distance Connections

- Several transport protocol variants for high bandwidth long distance connections (LFNs - Long Fat Networks) exist
- Frequent property
  - Effectively use available bandwidth
  - Unfriendly – "doesn't play nicely with others"
  - Unfair to different RTT flows
  - achieves better performance than standard TCP
  - is not fair to standard TCP
- General approaches for congestion control
  - loss-based: NewReno, CUBIC
  - delay-based: Vegas, CAIA Delay Gradiant (CDG)

TCP NewReno

- c.f. RFC 3782 - April 2004, Proposed Standard
- Properties
  - addresses problems that may arrive when multiple packets are dropped from a single window
  - Base algorithm described in RFC 2582 did not attempt to avoid unnecessary multiple Fast Retransmits after timeout.
  - RFC 2582 also defined "Careful" variant that avoids these unnecessary Fast Retransmits
  - „Careful“ variant of RFC 2582 NewReno as default

TCP CUBIC

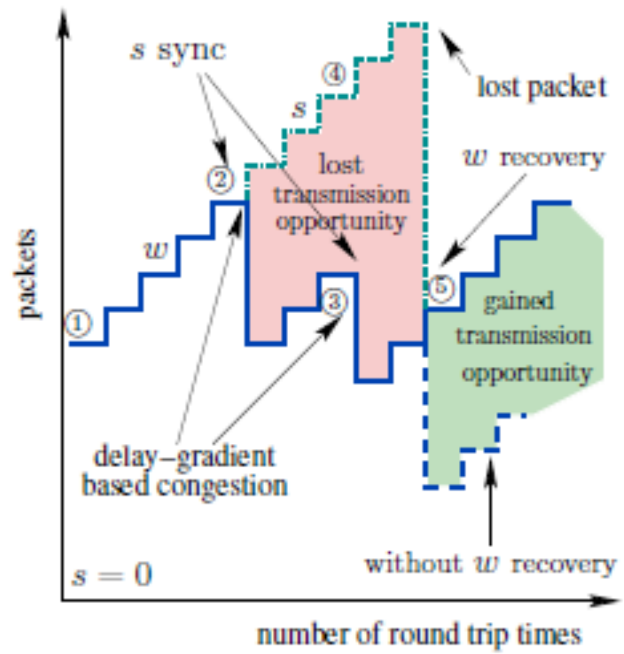
- CUBIC
  - Loss-based congestion control
    optimised for high bandwidth, high latency
- Properties
  - modified window-growth-control algorithm
  - window grows slowly around W_{max}
  - fast "probing" growth away from W_{max}
  - Standard TCP outperforms CUBIC's window growth function in short RTT networks.
  - CUBIC emulates standard (time-independent) TCP window adjustment algorithm, select the greater of the two windows (emulated versus cubic)
- Implemenation:
  - in Linux since kernel 2.6.19, in FreeBSD 8-STABLE

TCP Vegas

- TCP Vegas
  - by Lawrence Brakmo, Sean W. O'Malley, Larry L. Peterson at University of Arizona
  - published at SIGCOMM 1994
- Properties
  - delay-based congestion control
  - uses i^{th} RTT > min RTT + delay threshold, delay measured every RTT
  - Additive Increase Additive Decrease (AIAD) to adjust cwnd
- Properties
  - implementations available for Linux and BSD

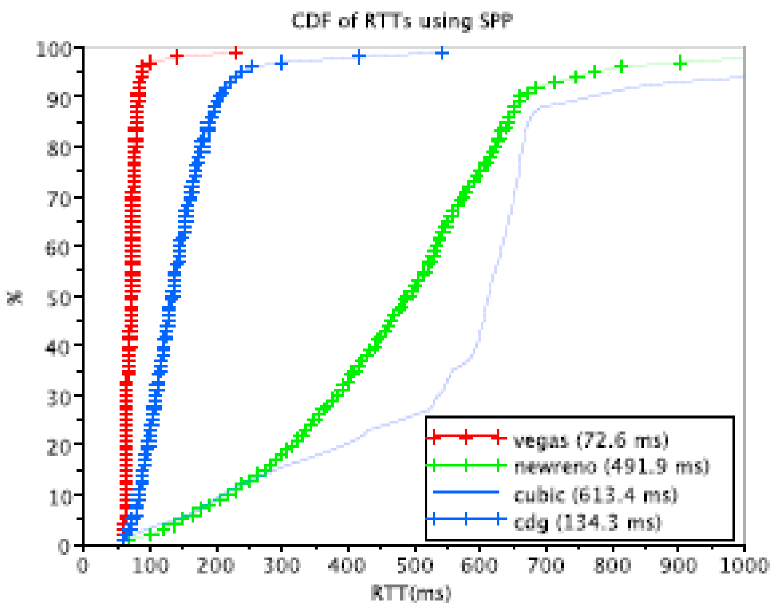
Delay Gradient TCP

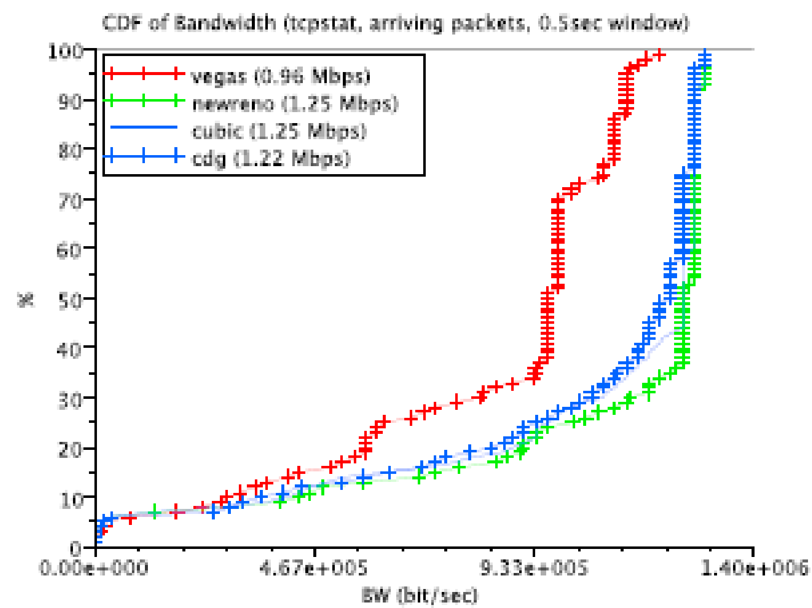
- D. Hayes, G. Armitage, "Revisiting TCP Congestion Control using Delay Gradients," IFIP/TC6 NETWORKING 2011, Valencia, Spain, 9-13 May 2011 http://caia.swin.edu.au/cv/dahayes/content/networking2011-cdg-preprint.pdf
- CDG ("CAIA Delay-Gradient") modified TCP sender behaviour:
  - uses delay gradient as a congestion indicator
  - tolerates non-congestion packet loss, and backoff for congestion related packet loss
  - works together with loss-based congestion control flows, e.g. NewReno



Comparison of TCP Variants

- Grenville Armitage: A rough comparison of NewReno, CUBIC, Vegas and 'CAIA Delay Gradient' TCP (v0.1), CAIA Technical report 110729A, 29 July 2011 http://caia.swin.edu.au/reports/110729A/CAIA-TR-110729A.pdf
- SPP Synthetic Packet Pairs Tool http://caia.swin.edu.au/tools/spp/





Multipath TCP

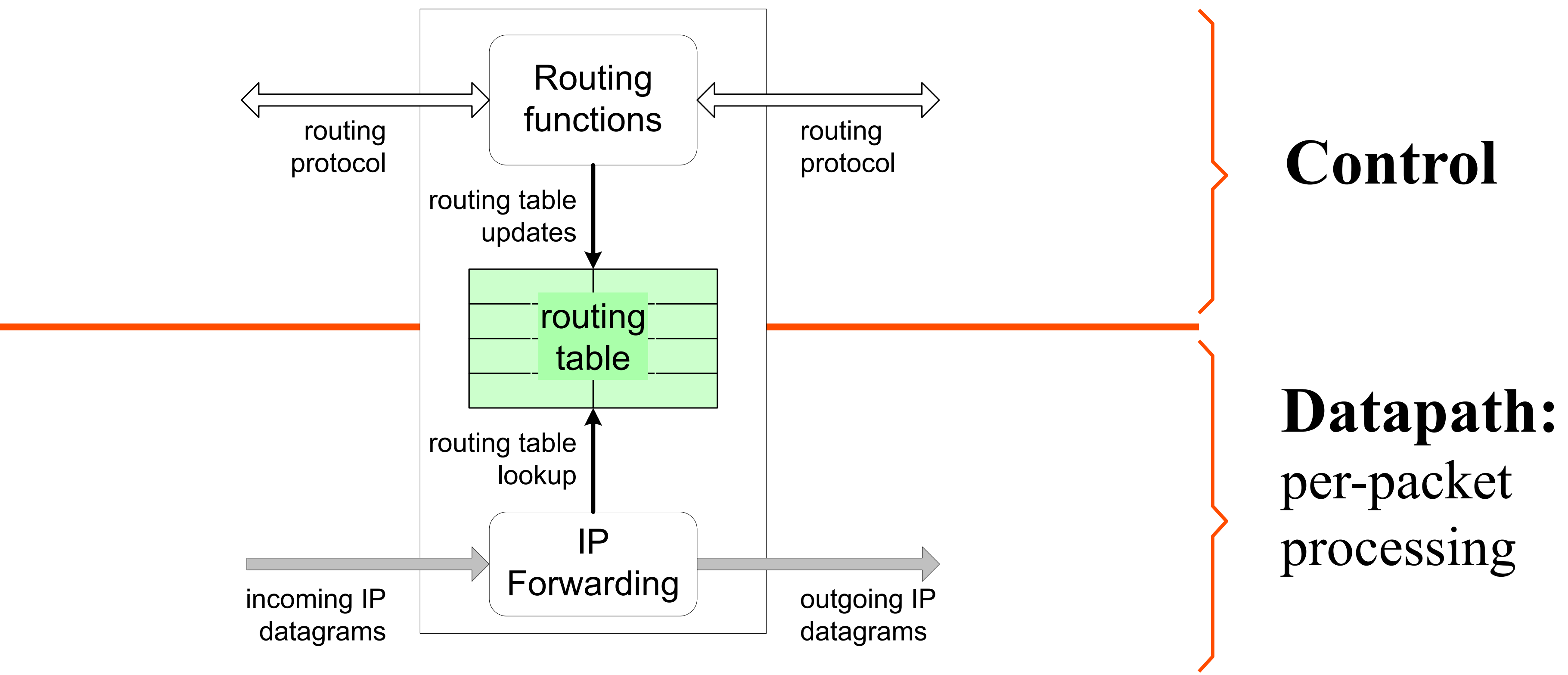
- **IETF Working Group Multipath TCP (mptcp)**
  http://datatracker.ietf.org/doc/charter-ietf-mptcp/
- Key goals
  - deployable and usable without significant changes to existing Internet infrastructure
  - usable by unmodified applications
  - stable and congestion-safe, including NAT interactions
- Objectives
  a. An architectural framework for congestion-dependent multipath transport protocols
  b. A security threat analysis for multipath TCP
  c. A coupled multipath-aware congestion control algorithm
  d. Multi-addressed multipath extensions to current TCP
  e. Application interface considerations

Multipath TCP

- http://bgp.potaroo.net/ietf/html/ids-wg-mptcp.html
  - TCP Extensions for Multipath Operation with Multiple Addresses, draft-ietf-mptcp-multiaddressed
  - MPTCP Application Interface Considerations, draft-ietf-mptcp-api-06.txt
- Milestones
  - Submit to IESG architectural guidelines and security threat analysis as informational RFC(s)
  - Submit to IESG basic coupled congestion control as an experimental RFC
  - Consensus on what high-level changes are needed to the current MPTCP Experimental document in order to progress it on the standards track
  - Apr 2013 Implementation advice (Informational) to IESG
  - Aug 2013 Use-cases and operational experiences (Informational) to IESG
  - Dec 2013 MPTCP-enabled middleboxes (Informational) to IESG
  - Dec 2013 MPTCP standards track protocol to IESG
  - Jan 2014 Re-charter or close

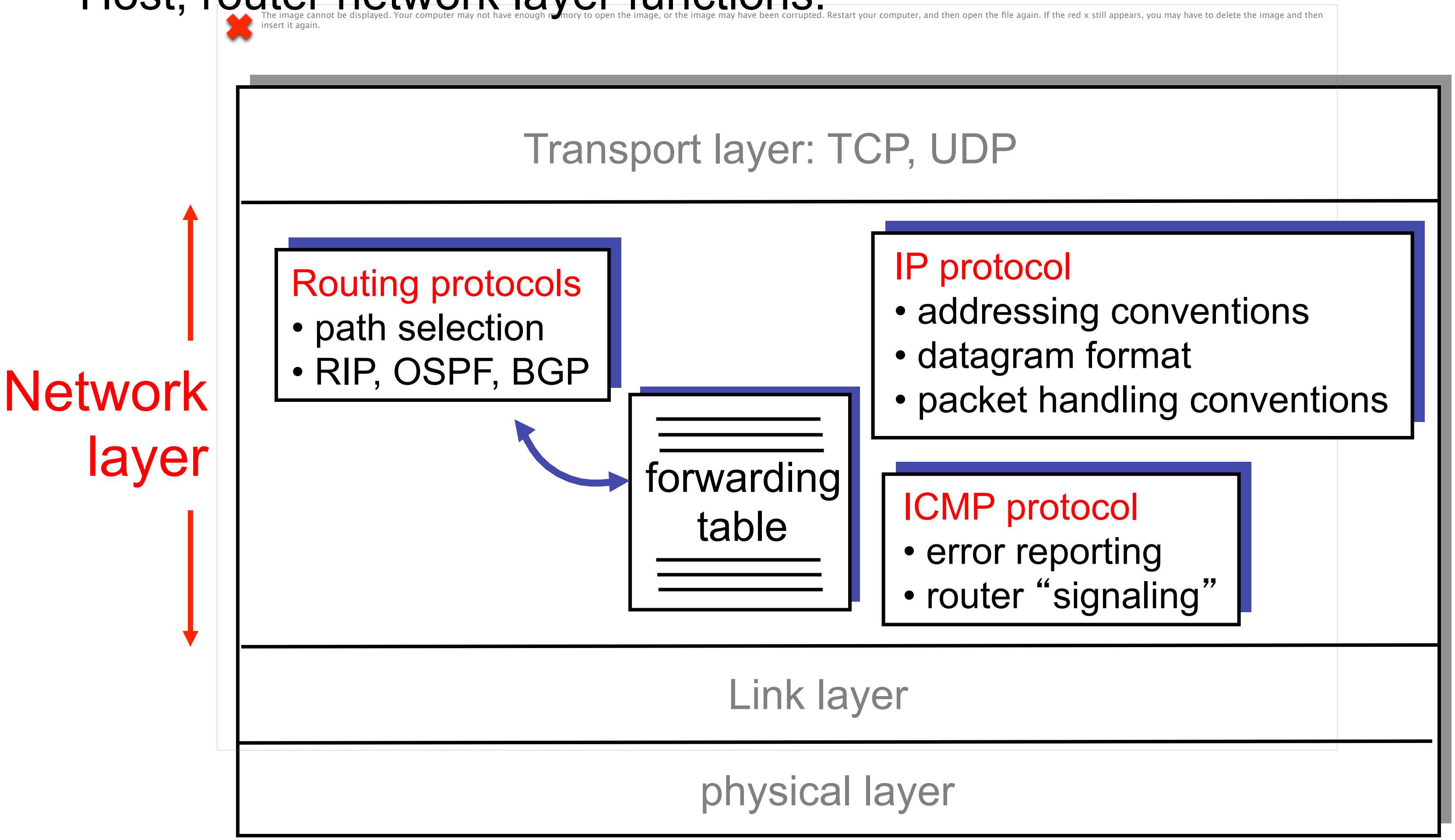
Internet Protocol

Functional Components



The Internet Network layer

Host, router network layer functions:



ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute and ICMP

- Source sends series of UDP segments to destination
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP "dest port unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops.

IP addressing: CIDR

CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

subnet part ← → host part

11001000 00010111 00010000 00000000

200.23.16.0/23

IP addresses: how to get one?

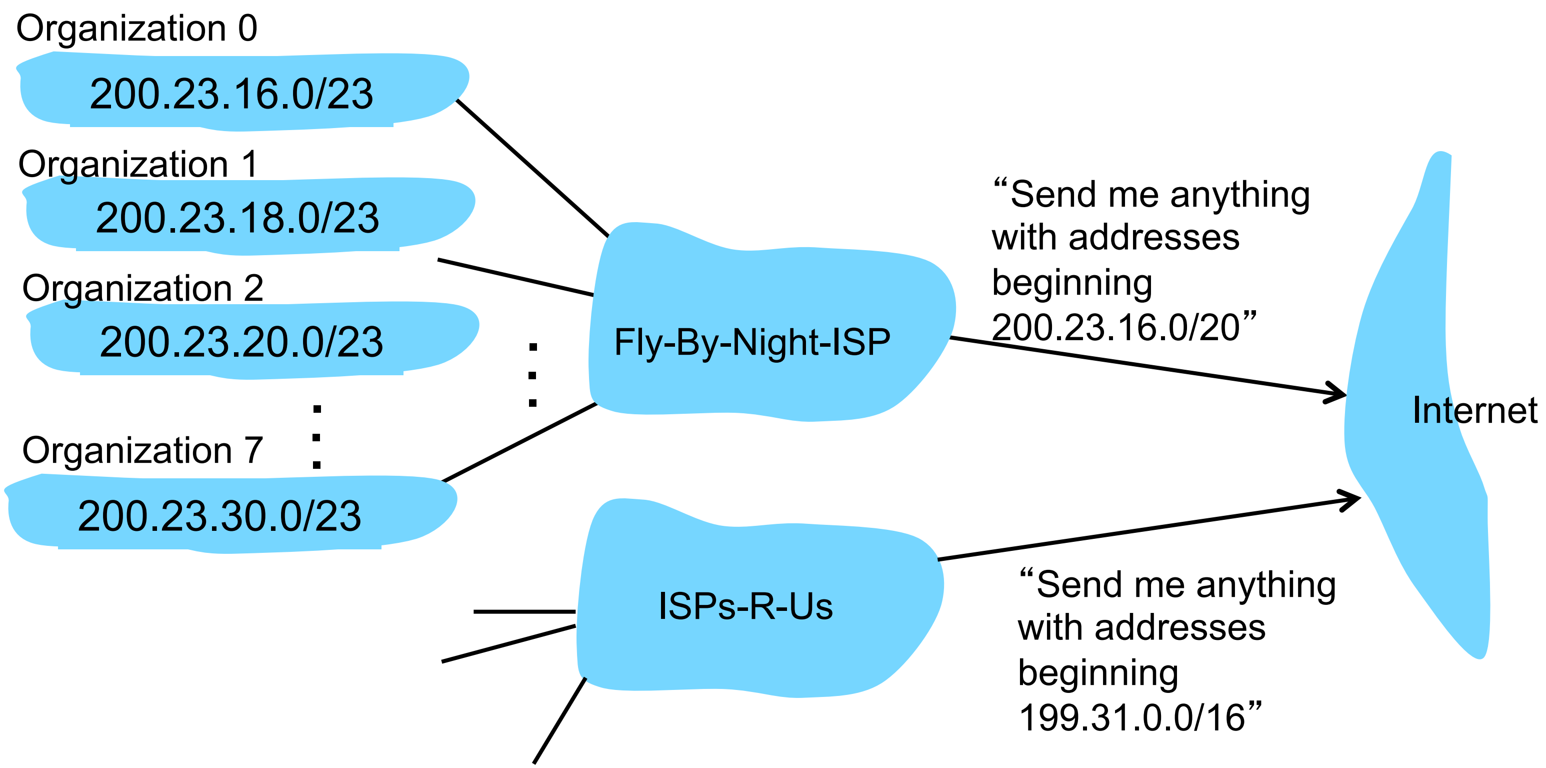
Q: How does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

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

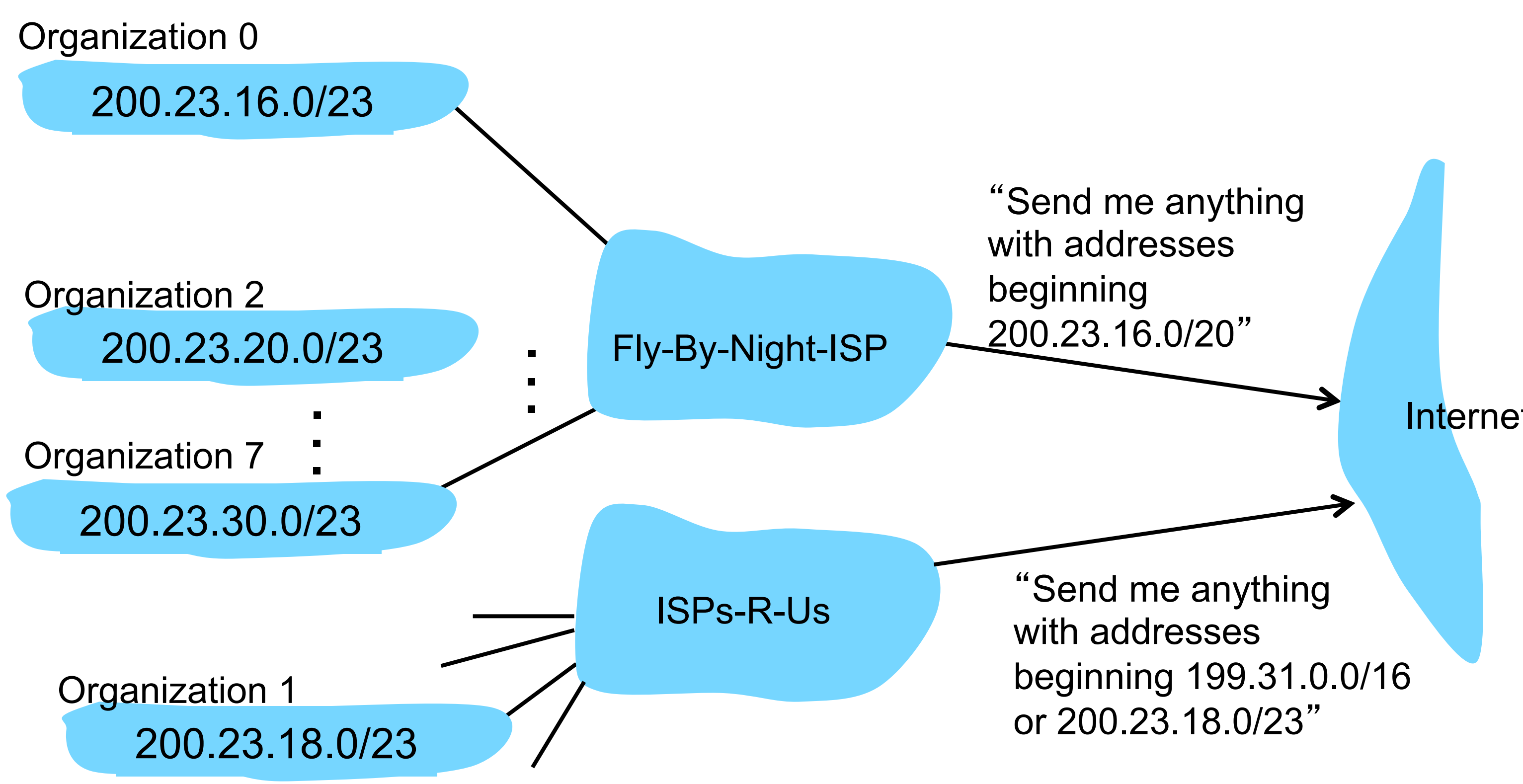
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



IP addressing: the last word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

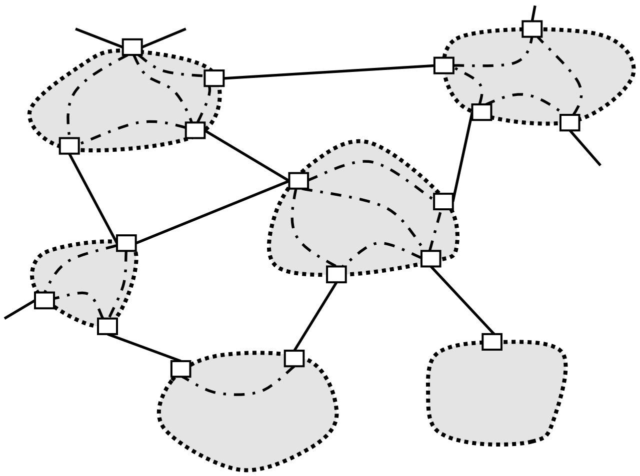
How big is the Internet?

- Many measures:
  - networks (routed entities)
  - domains, host names (but: several names per host!)
  - directly (continuously) attached hosts ("ping'able")
  - IP-connected hosts (including dialin, e.g. PPP)
  - firewalled hosts
  - e-mail reachable

- What is the German Internet?
  - Entities within Germany
  - Entities operated by Germans / German organisations
  - Entities used by Germans / German organisations

Counting

- Worldwide
  - > 700.000.000 Hosts
  - > 37.000 Autonomous Systems
  - > 3.000.000.000 Assigned IP Addresses
  - > 2.180.000.000 Reachable IP Addresses
- Europe
  - > 126.600.000 Hosts
  - > 19.000 Autonomous Systems
  - > 420.000.000 Reachable IP Addresses
  - > 500.000.000 Assigned IP Addresses
- Germany
  - > 13.300.000 Hosts
  - > 1.200 Autonomous Systems
  - > 70.700.000 Assigned IP Addresses (5.500 prefixes)
  - > 62.700.000 Reachable IP Addresses



Snapshot 2011