UiO : Faculty of Mathematics and Natural Sciences
University of Oslo

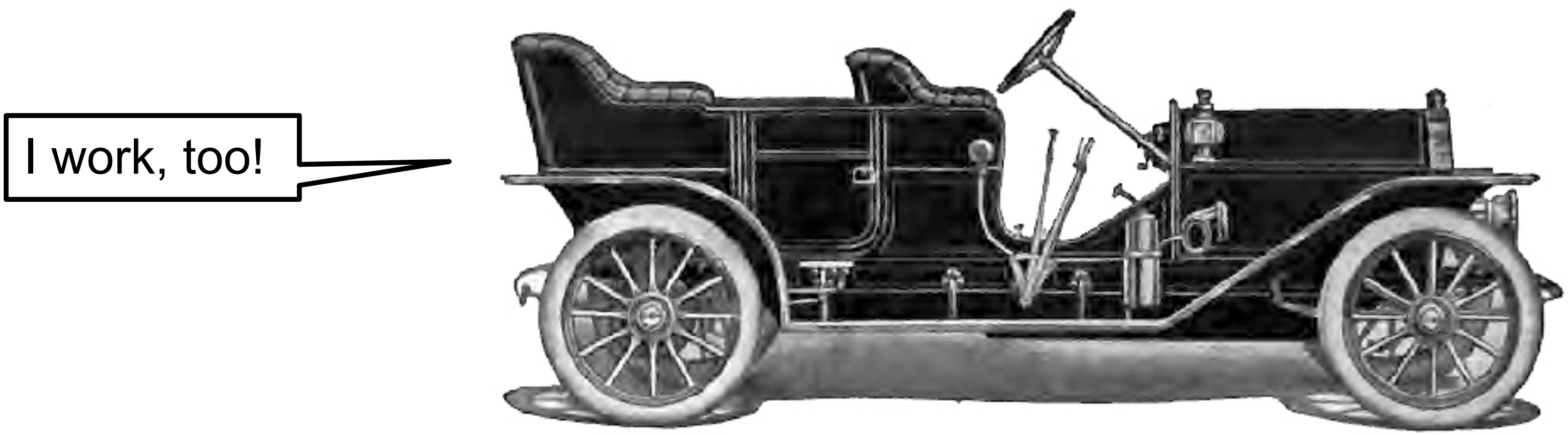
ifi

INF3190
A critical look at the Internet / alternative network architectures

Michael Welzl

What’s the problem?

- The Internet works!



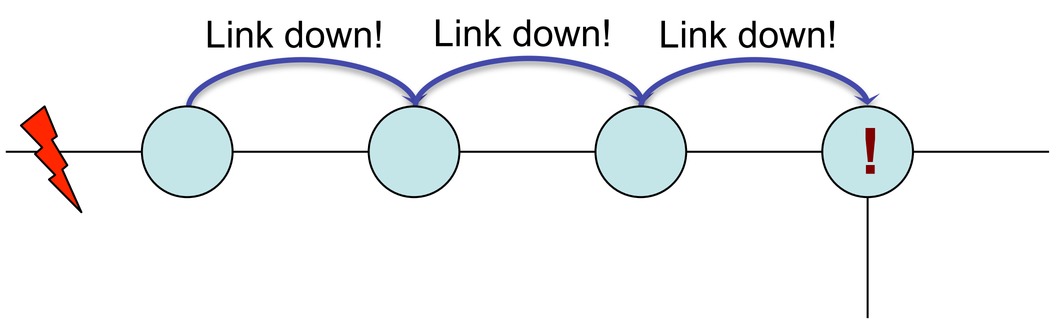
- Could be faster
- Could be safer
- Could be more reliable
- (Internet): could be simpler / easier to manage

Reliable?

- "The Internet started as a military network designed to survive a nuclear attack"
  - You can find this in lists of "Internet myths", next to "Al Gore invented the Internet"
- The Internet (then ARPANET) was the first large packet network
  - As such, it redirects in case of failure
- But how reliable is packet forwarding?

Path failures

- Internet routing: based on informing neighbours
  - Reason: scalability – no computer can store all information about all links on the Internet

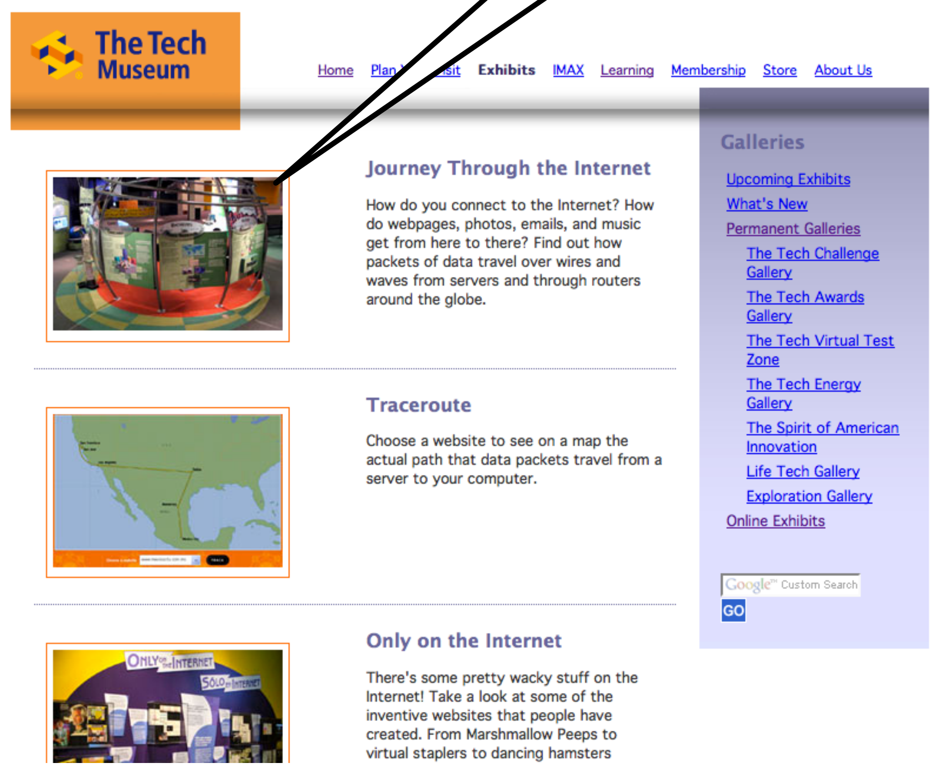


- Common default value for these updates: 30 seconds
- Packets are dropped until routing reconverges

Route around congestion

- The Internet does not do this.
  - No it really doesn't. This is another myth.

I have seen it here.



- Explanation
  - Path A congested ➔ Take Path B.
  - Path B congested ➔ Take Path A.
  - Dividing traffic well: must know A and B ➔ Scalability problem

More problems with routing

- Internet routing / forwarding is hierarchical
  1) ➔ Oslo
  2) ➔ Blindern

- The Internet grows fast
  – Number of "Cities" (Autonomous Systems (AS)) grows roughly exponentially
  – Problem for memory in core routers
    - Old routers may cease to work
  – Larger routing tables ➔ slower convergence

Another problem with Internet growth

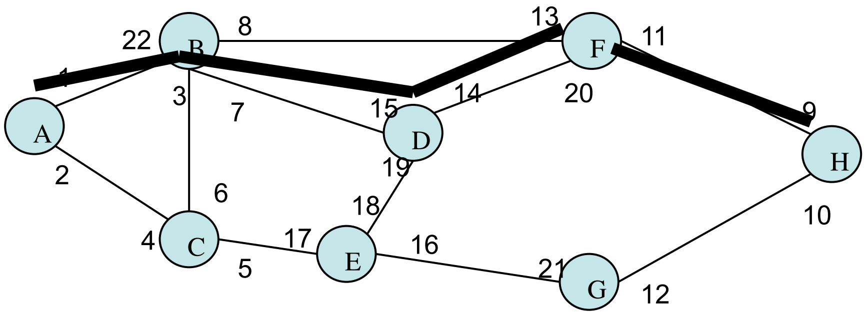
- We're running out of addresses!
- Internet addresses are hierarchical (Norway ➔ Oslo ➔ Street ➔ house number)
  - The last top-level address was given out on 3.2.2011
- Solution: update address format (more bits)
  - IPv6 (replacing the old "Internet Protocol", IPv4)
  - First standardized 1998
  - Still a cool new thing. Do you use it?

Another problem with IP addresses

- I moved from Bogstadveien to Thorvald Meyers Gate
  - I registered to have mail forwarded for a month for free (postens flyttehjelp)
  - This gives me time to inform everyone about my new address (posten offers an easy service to do that)
  - After this month, I have no more business with the old address

- Imagine a world without addresses
  - But where, instead, my name is Michael Bogstadveien Welzl
  - Moving: use "postens flyttehjelp" forever!
  - Imaging changing from world 1 to world 2 or vice versa… easy?

Multihoming (slides by John Day, adapted)



- Internet addresses name the interfaces. Let's see what this does
- Assume: A wants to send a PDU to H, so it goes to DNS and looks up the address and gets 9. Now these nodes have been running a routing algorithm and the route is A, B, D, F, H.
- So what do the router tables look like? (next slide)

Forwarding Tables for Each Router

Using Interface Addressing

Outgoing Interface

Destination Address	A	B	C	D	E	F	G	H
1	-	22	6	15	18	20	21	9
2	-	22	4	19	17	20	21	10
3	1	-	6	19	17	20	21	10
4	2	1	-	15	18	20	12	9
5	2	3	-	19	17	20	21	10
6	1	3	-	19	15	20	12	9
7	2	-	5	15	15	20	21	9
8	2	-	6	14	18	13	12	9
9	1	7	5	14	18	11	21	-
10	2	3	5	19	16	20	12	-
11	2	3	5	19	16	-	12	9
12	1	3	5	14	18	11	-	10
13	1	7	5	15	17	-	21	10
14	1	8	5	-	16	20	12	9
15	1	7	6	-	17	20	21	9
16	2	7	5	14	-	11	21	10
17	2	3	5	15	-	20	12	9
18	2	7	4	19	-	11	12	9
19	2	3	5	-	18	11	21	10
20	1	7	5	14	18	-	21	10
21	2	3	5	19	16	20	-	9
22	1	-	4	19	17	20	21	10

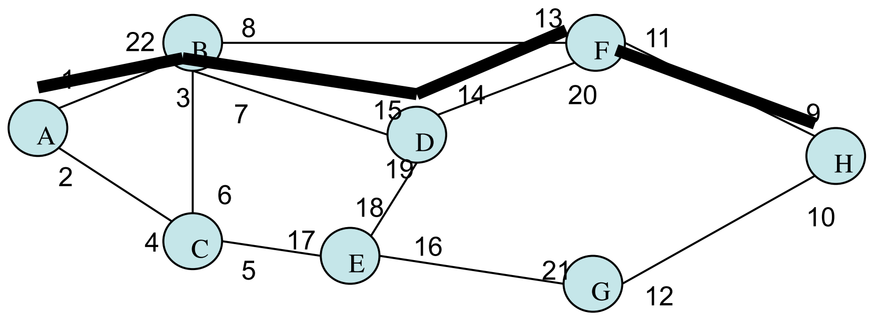
Forwarding Tables for Each Router

Using Next Hop Addressing

Next Hop

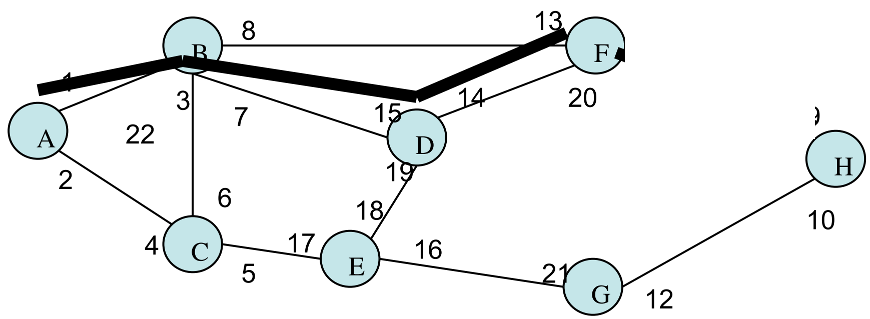
Destination Address	A	B	C	D	E	F	G	H
1	-	1	3	7	19	14	16	11
2	-	1	2	18	17	14	16	12
3	22	-	3	18	17	14	16	12
4	4	1	-	7	19	14	10	11
5	4	6	-	18	17	14	16	12
6	22	6	-	18	7	14	10	11
7	4	-	17	7	7	14	16	11
8	4	-	3	14	19	13	10	11
9	22	15	17	14	19	9	16	-
10	4	6	17	18	21	14	10	-
11	4	6	17	18	21	-	10	11
12	22	6	17	14	19	9	-	12
13	22	15	17	7	17	-	16	12
14	22	13	17	-	21	14	10	11
15	22	15	3	-	17	14	16	11
16	4	15	17	14	-	9	16	12
17	4	6	17	7	-	14	10	11
18	4	15	2	18	-	9	10	11
19	4	6	17	-	19	9	16	12
20	22	15	17	20	19	-	16	12
21	4	6	17	18	21	14	-	11
22	22	-	2	18	17	14	16	12

A PDU is sent with destination address 9 from A



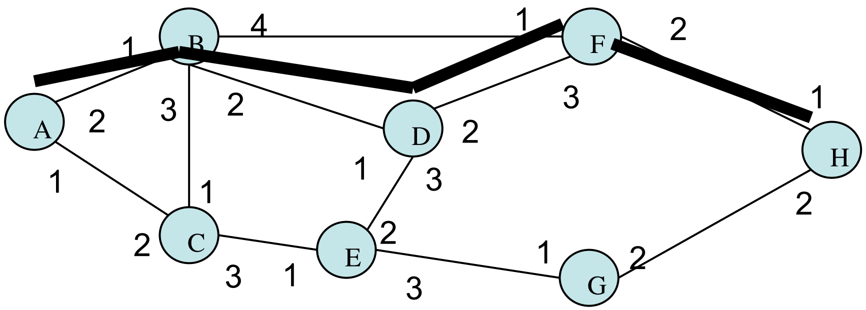
- A consults its Forwarding Table and sends it on outgoing address 1, next hop 22.
- B consults its FT and sends it on outgoing address 7, next hop 15.
- D consults its FT and sends it on outgoing address 14, next hop 20.
- F consults its FT and sends it on outgoing address 11, next hop 9.
- Now another PDU is sent from A to address 9, just after it leaves, the interface goes down.

What happens when link F - H fails?



- What happens if the link with address 9 goes down?
  - In a few 10s of ms, a routing update is done, and Addresses 9 and 11 are eliminated from the forwarding table.
  - After several seconds and many retries, A determines that Address 9 is not responding,
  - All TCP connections with Address 9 are terminated.
  - All PDUs enroute to 9 are lost.
  - Hopefully, there is a second DNS entry that lists H as also at Address 10.
  - Connections are re-established using address 10. Several seconds have elapsed.

Now: node addresses



- Since we want to emphasize that we are naming nodes, lets just use the letters for addresses. But we still have to say which wire to send them on.
  - There are two cases in general:
    - Point-to-point Wire: No need for lower layer addresses use local identifiers.
    - Multi-access wired or wireless: Here we need addresses, use MAC addresses
  - We have only wires, so lets assign small integers as the local interface identifiers.
- Now lets say that A wants to send a PDU to H, so it goes to DNS and looks up the address and gets 9. Now these nodes have been running a routing algorithm and the route is A, B, D, F, H.
- So what do the router tables look like? (see next slide)

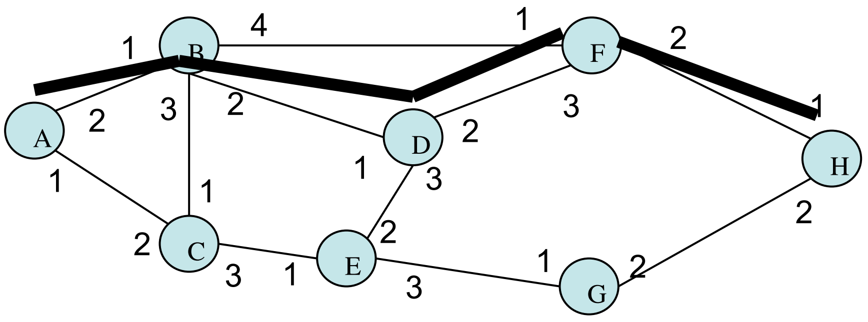
Forwarding Tables for Each Router
Using Node Addresses
Outgoing Interface

Destination Address	A	B	C	D	E	F	G	H
A	-	1	2	1	1	3	1	1
B	2	-	1	1	1	3	1	1
C	1	3	-	1	1	3	1	1
D	2	2	1	-	2	3	1	1
E	1	2	3	3	-	3	1	1
F	2	2	1	2	2	-	1	1
G	1	2	3	3	3	3	-	2
H	2	2	1	2	2	2	2	-

Forwarding Table for Each Router
Using Next Hop

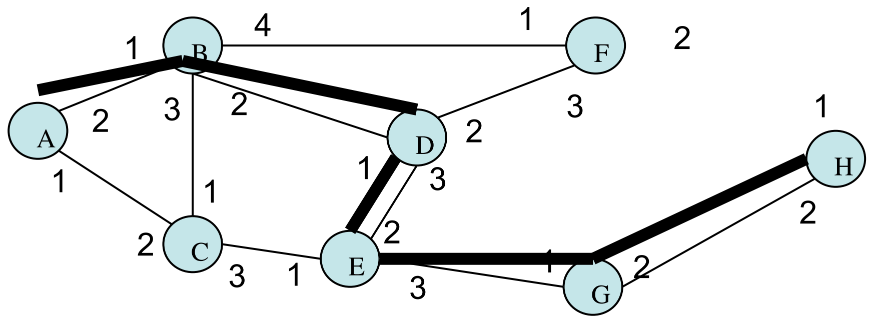
Destination Address	A	B	C	D	E	F	G	H
A	-	A	A	B	C	D	E	F
B	B	-	B	B	C	D	E	F
C	C	C	-	B	C	D	E	F
D	B	D	B	-	D	D	E	F
E	B	D	E	E	-	D	E	F
F	B	D	B	F	D	-	E	F
G	B	D	E	E	E	D	-	G
H	B	D	B	F	D	H	H	-

Sending a PDU from A to H



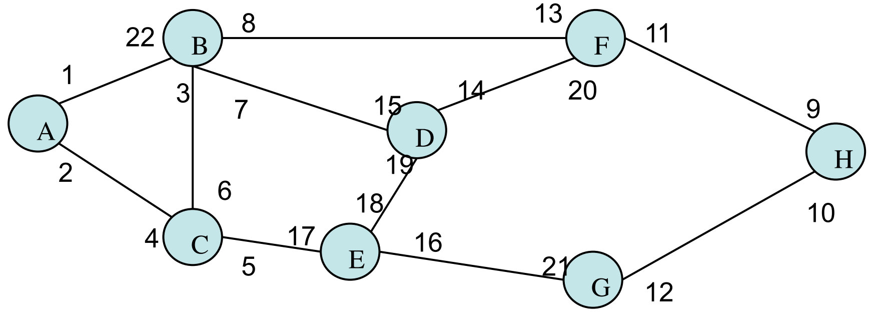
- A has a PDU addressed to H:
  - A consults its Forwarding Table and sends the PDU on interface 2 to B.
  - B consults its Forwarding Table and sends the PDU on interface 2 to D.
  - D consults its Forwarding Table and sends the PDU on interface 2 to F.
  - F consults its Forwarding Table and sends the PDU on interface 2 to H.

What happens when link F - H fails?



- What happens if just after the PDU is sent the Link from F to H fails?
  - In a few 10s of ms, a routing update is done, and a new Routing Table is generated.
  - The PDU gets to D after the routing update has concluded and is delivered to H as if nothing happened.
  - PDUs that might have been between B, D, and F might get re-routed. Only PDUs on the wire from F to H would be lost.

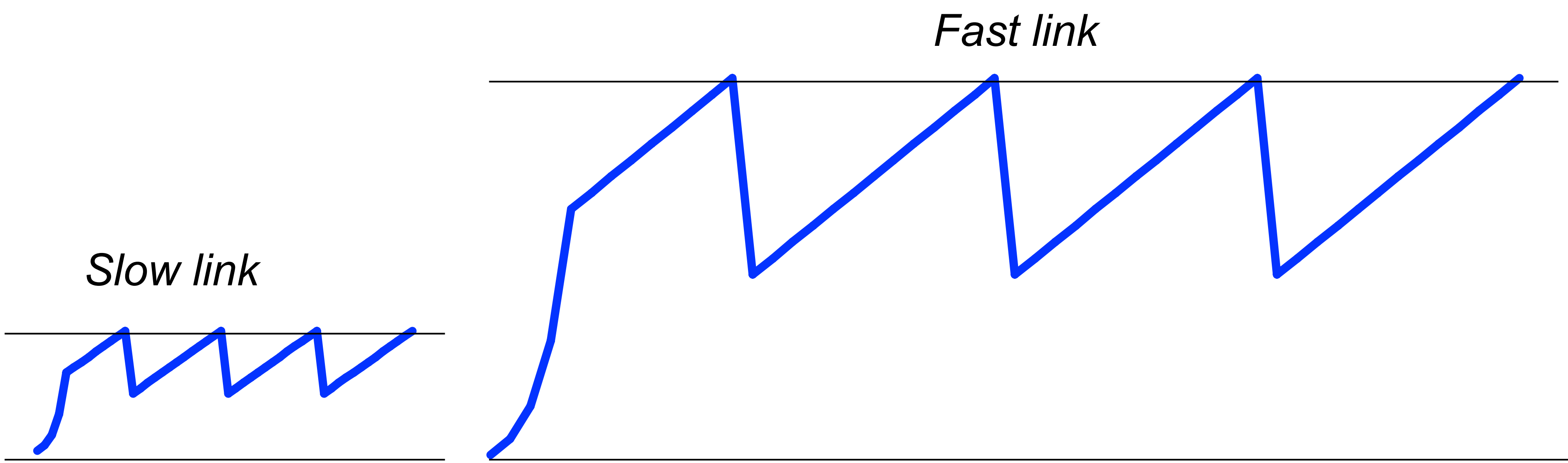
The new Internet solution to this: MPTCP



- A and H initially tell each other about all their addresses (A: 1, 2; H: 9, 10)
- MPTCP subflows are established; how many, depends on the path manager configuration
  - E.g. fullmesh: 4 subflows (1-9, 1-10, 2-9, 2-10)
  - Then, normally: use all of them (schedule traffic, e.g. round-robin or preferring least-RTT path until cwnd full; “cautious” congestion control – be at most as aggressive as 1 TCP across a common bottleneck)
- F-H link goes down: one subflow becomes unavailable, no TCP failure for application
- This is done e.g. by iPhones since iOS 7: https://support.apple.com/en-us/HT201373

Keeping up with growing speed

- TCP (Transmission Control Protocol):
  - Linear increase doesn't scale well with capacity
  - Short flows often terminate in Slow Start; when capacity is large, limitation is RTT; keeping it low becomes key!
  - Many flows compete for capacity; no coordination, even when they share bottlenecks... but the diversity of capacities grows...



Why is security so hard?

- There is no “best layer”
  - e.g. authenticate user or IP address? each has value
- A lot of hard decisions for the app programmer

Table 1: Some layering choices to incorporate security. Note: (D)TLS is a transport security method (could also use TCPcrypt); IPSec is either EH or AH negotiated, doubling the number of choices related to IPSec.

Transport Service	**Example Protocol Stacks**
Reliable Stream	TLS/TCP, TCP/IPSec, TLS/TCP/IPSec
Message Stream	DTLS/SCTP, SCTP/IPSec, DTLS/SCTP/IPSec, SCTP/DTLS/UDP/IP, DTLS/SCTP/DTLS/UDP/IP, SCTP/DTLS/UDP/IPSec
Datagram	DTLS/UDP, UDP/ IPSec, DTLS/UDP/ IPSec, DTLS/DCCP, DCCP/ IPSec, DTLS/DCCP/ IPSec, DCCP/UDP, DTLS/DCCP/UDP, DCCP/UDP/ IPSec, DTLS/DCCP/IPSec/UDP
Error-Tolerant	DTLS/UDPL, UDPL/IPSec, DTLS/UDPL/IPSec

Conclusion on Internet criticism

- A lot of work needed to make the Internet
  - Faster
  - More reliable
  - Survive ☺ while remaining affordable

- And this was only scratching the surface…

Alternative Network Architectures

"Alternative"? What's the point?

- Alternative architecture X can be overlay, underlay, or used on a part of a path
  - X-over-IP
  - IP-over-X
  - IP – translate – X – translate – IP
  - Endpoint: X – translate – IP
  - Endpoint: IP – translate – X
- Note: a lot of this is happening with IPv6

Software-Defined Networking (SDN)

- Potential as enabling technology for many ideas
- Decoupling of forwarding plane from control plane; enables efficient hardware
- Centralized controller instructs switches on what to do with traffic via an open interface (best known: OpenFlow)
- Makes network easier configure / manage

Network Functions Virtualization (NFV)

- Today, there are special boxes for e.g. NAT, firewalling, DNS, caching, etc.
- NFV envisions generic boxes where "Virtualized Network Functions" (VNFs) – NAT, firewalling, DNS, caching, etc. – can be installed
- Somehow related to SDN but different focus

Example “new architecture” #1: ICN

- Criticism: Internet establishes circuits (telephone model), does not match usage (WWW is not telephony, browser cares about content, not a physical server)

- Information-Centric Networking (ICN)
  - Also: Content-Centric Networking (CCN), Named Data Networking (NDN)

ICN: pub/sub as basic primitive

- Publish: advertise content; Subscribe: request content
- Decoupling in time & space: publisher, subscriber don't have to know each other's location or be online at the same time
- In ICN, pub/sub…
  - is not a Distributed System over the Internet, but the way the whole network operates
  - has different names (REGISTER / FIND etc.)
  - only operates on the name
  - Contains a one-time fetch variant of "subscribe" (but often also typical "subscribe")

ICN: caching and security

- Interior nodes (routers) get a request
  - Content in cache? Then send it
  - Else, forward request onward… (intermediate nodes will cache)

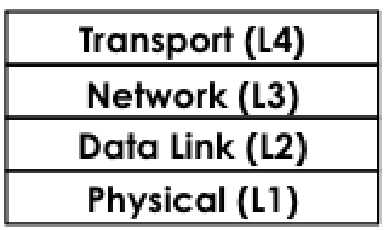
- Node that publishes content cannot be used for security anymore ➔ content is digitally signed

ICN: issues and directions

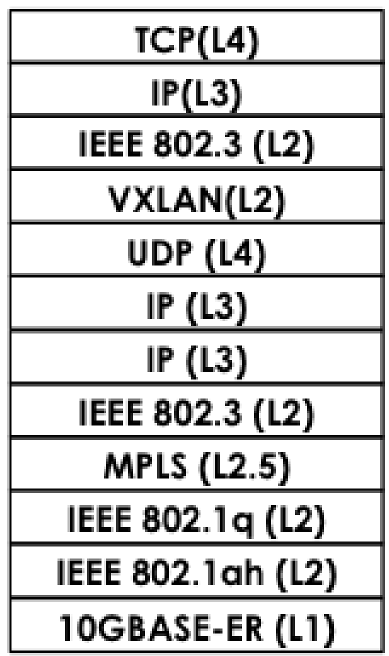
- Consumer must know:
  - name of content
  - public key of provider
- …and ICN system must be able to connect them
- How to do name-based routing? (Internet-like or totally different?) Inter-domain: policies or not?
- End-to-end congestion control no longer reasonable…
- HTTP caching is very common today
  - Is ICN much better, or even needed?

Example “new architecture” #2: RINA

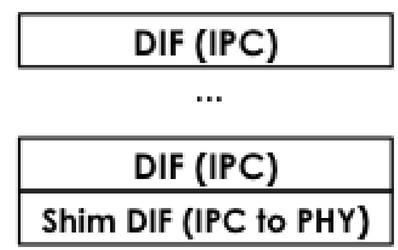
- Recursive InterNetwork Architecture
  - Layers only provide scope



The simple model

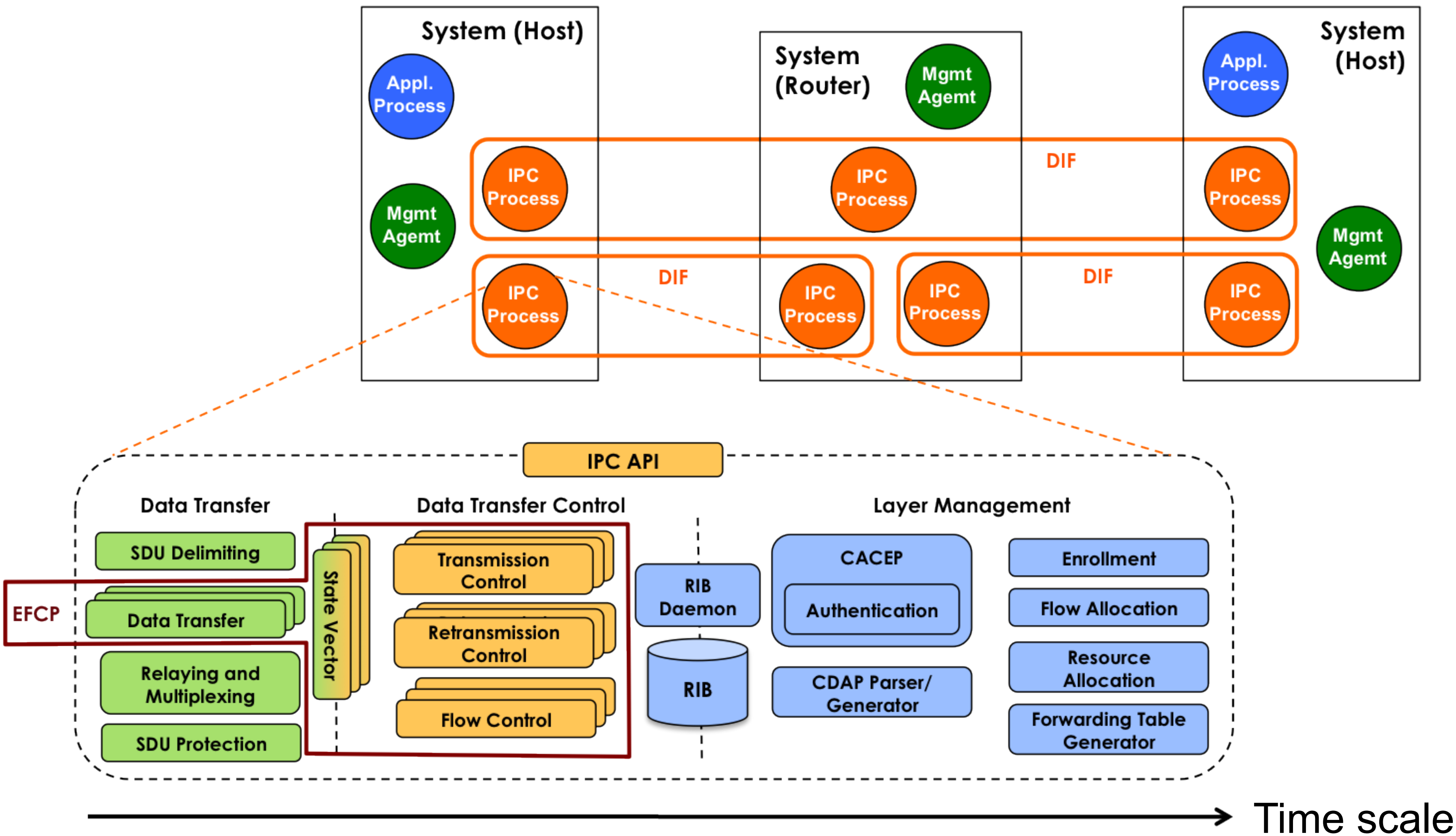


The complex reality (just an example)

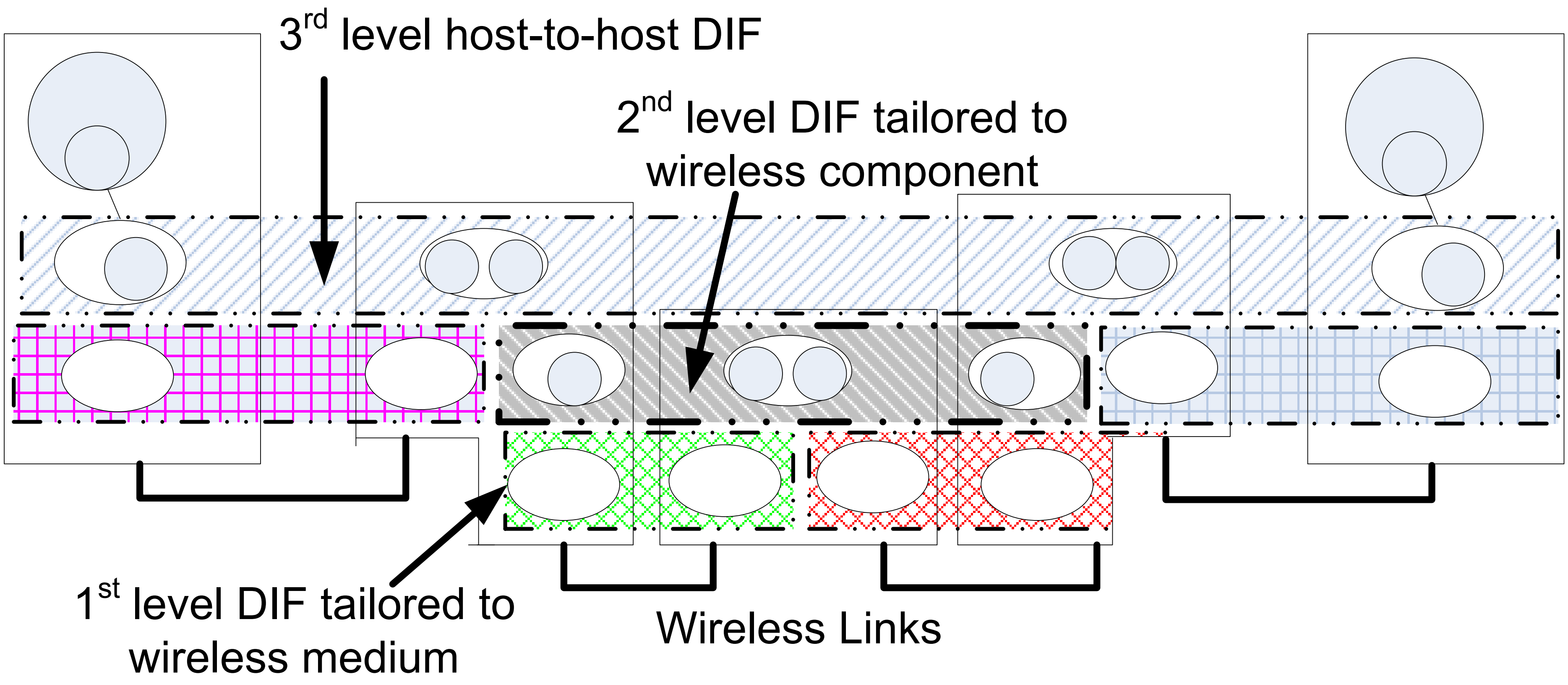


RINA

RINA: DIF: Distributed IPC Facility
(IPC: Inter Process Communication)



RINA: DIF implementations differ (policies)



RINA benefits

- Getting addressing right facilitates topological routing, mobility, multihoming
- Should be more secure: enrollment authenticated, else can’t even reach a node
- Management should be easier