# PATTERNS IN NETWORK ARCHITECTURE:

# ROUTING, FORWARDING, AND

# REACHABILITY

# PATTERNS IN NETWORK ARCHITECTURE:

# ROUTING, FORWARDING, AND REACHABILITY

## OUTLINE

- Survey of Concerns and Patterns
- Reachability Properties
  - Discussion of “Checking beliefs in dynamic networks”
  - Presentation of an Alloy model
  - Comparison between Alloy and NoD
- Reachability Properties and Layered Composition
  - Discussion of “Resilient overlay networks”
  - Resource allocation in layered composition

# WHAT DO WE CARE ABOUT?

## PERFORMANCE

- latency, bandwidth, jitter

## RELIABILITY

- packet loss

## RESOURCE UTILIZATION

- minimum link utilization (for efficiency)
- maximum link utilization (for reliability)

## FAILURES

- link failure
- node failure

## SAFETY AND PROGRESS REQUIREMENTS

- many!

## RESOURCE ALLOCATION

- load balancing on paths
- queueing and rate limiting

## COSTS OF ROUTING

- network monitoring
- signaling, computation, and security for routing algorithms
- forwarding-table updates

## COSTS OF FORWARDING

- number of routers
- size of forwarding table at each router
- forwarding delay at each router
- header bits in packets

## SCALABILITY!

# WHAT ARE SOME ROUTING/FORWARDING PATTERNS?

# ROUTING AND FORWARDING PATTERNS 1: DECENTRALIZED

| "PUSH" ROUTING | "PULL" ROUTING |
| --- | --- |
| routes are disseminated by advertising ("push") | routes are learned passively or requested when needed |
| forwarding tables are "dense", in the sense that they should have an entry for every name | forwarding tables are "sparse" |
| names are hierarchical and aggregated | names are "flat" and unstructured |
| name assignment is decentralized by means of the hierarchy | there is no hierarchy, but name assignment can be decentralized by other means (e.g., manufacture) |

# ROUTING AND FORWARDING PATTERNS 2

## CENTRALIZED CONFIGURATION

- routing algorithm is centralized, it gives forwarding-table entries to routers
- with SDN, controller updates forwarding tables automatically
- with MPLS, there may be manual configuration of routers

## MOBILE AD-HOC NETWORKS

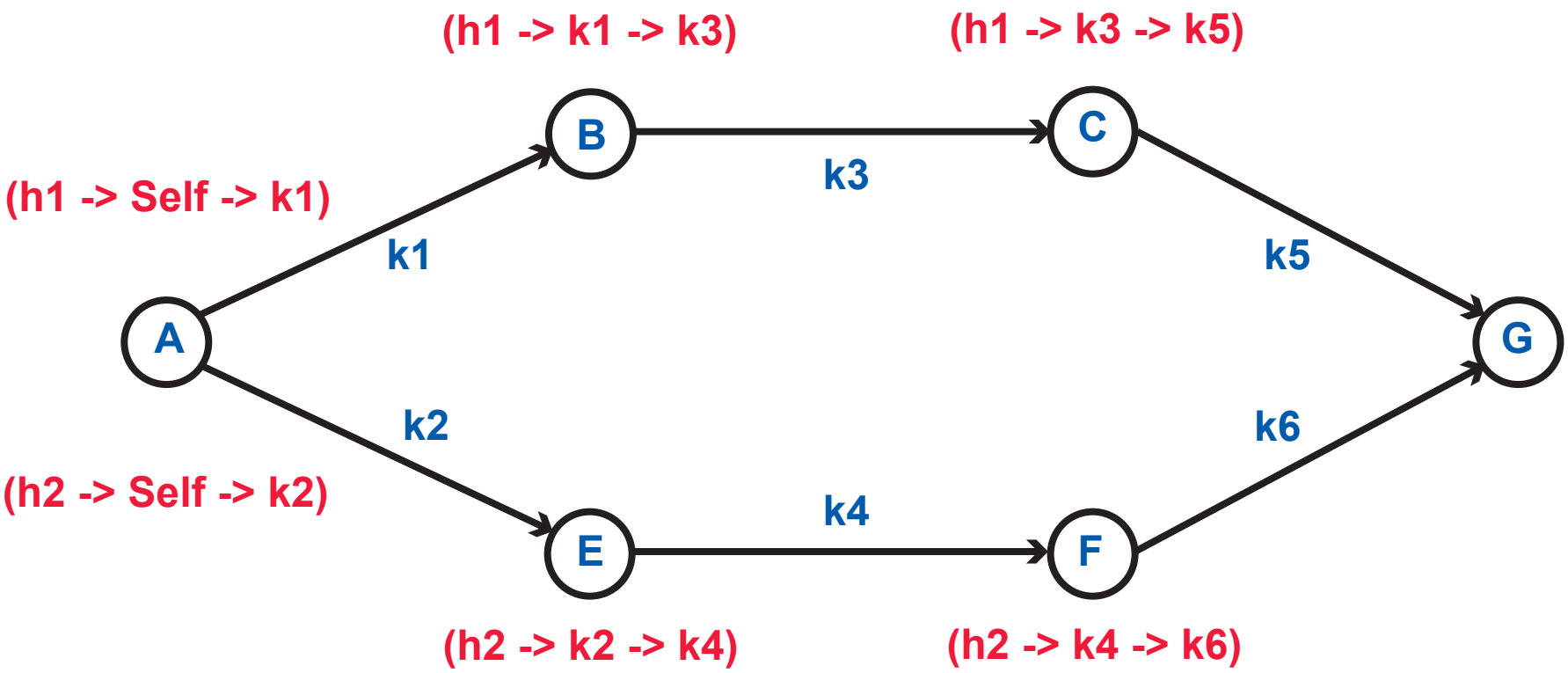
- no distinction between hosts and routers
- no point-to-point links
- forwarding is broadcasting to all in radio range
- nodes may keep some forwarding state, based on which nodes have been reachable (close) in the recent past

# DISCUSSION OF
# “CHECKING BELIEFS IN DYNAMIC NETWORKS”
# AND
# net2.als

# MULTIPATH ROUTING

h1: source = A, destination = G, hash = x1

h2: source = A, destination = G, hash = x2




reaches = { (h1 -> A -> B), (h2 -> A -> E),
(h1 -> A -> C), (h2 -> A -> F),
(h1 -> A -> G), (h2 -> A -> G) }

receives = { (h1 -> A -> G),
(h2 -> A -> G) }

# DISCUSSION OF "RESILIENT OVERLAY NETWORKS"

*recall from Lecture 1:*
*one of the purposes of a model is to provide well-defined terminology*

**Did you find this paper difficult to read?**

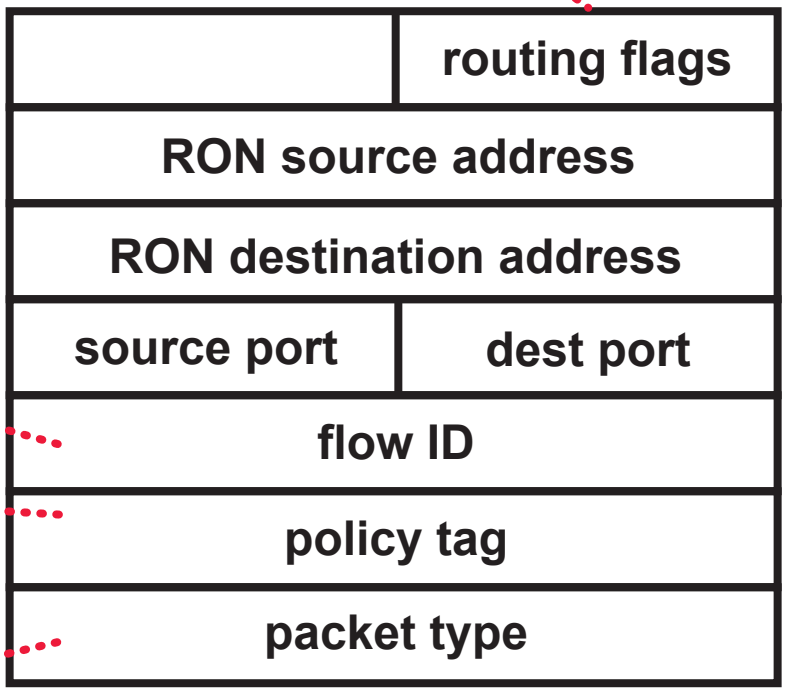
4.2, routing metric
Fig 5, routing preference
Fig 6, routing flags
4.4, routing preference flags

4.2, tag
4.2, flow ID

4.1, tag
4.2, policy routing
4.3, type of traffic
4.3, policy tag
4.3, packet tag
Fig 5, policy type
Fig 6, policy tag

while "policy classifier" means something different!

4.1, client
4.1, conduit
Fig 6, packet type (can also select protocol, like a well-known port)



# NETWORK PERFORMANCE

## PATHS

- from a source to a destination; this is what we really care about
- path performance is derived from link performance:

***the sum of latencies,***
***the minimum of bandwidths,***
***the product of packet losses,***
***the logical-or of failures***

## RESOURCE ALLOCATION

- routing allocates sessions to paths, does load-balancing
- forwarding enforces routing decisions, also secondarily controls resource allocation with queueing and rate limiting

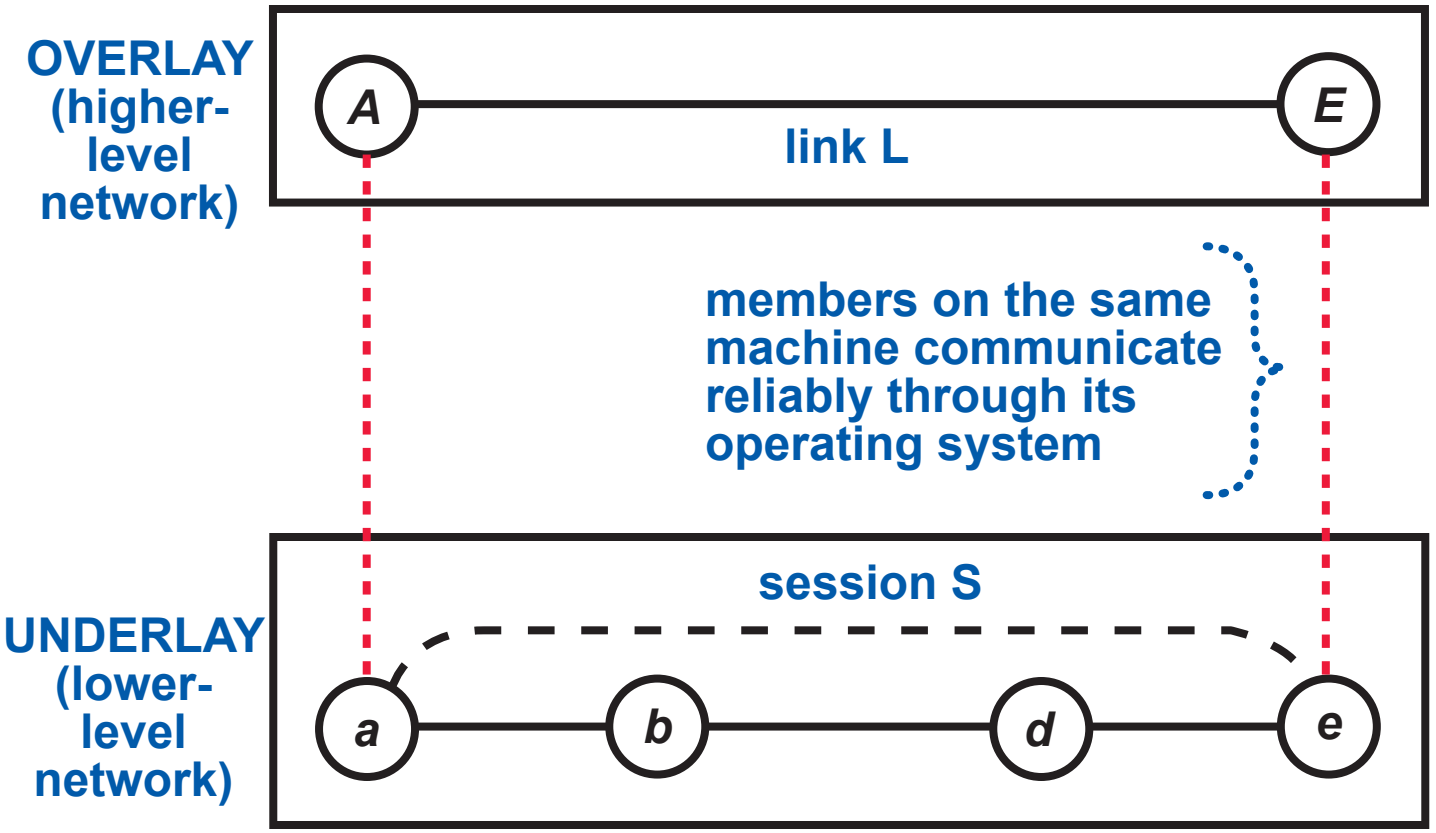
**link performance is partially dependent on its load/utilization, so optimization can be complex**

this is what routing algorithms are good at!

typically, one routing pattern per network

# A NETWORK MODEL: THE "USES" HIERARCHY OR VERTICAL COMPOSITION

when an overlay uses an underlay,
a link in the overlay is implemented
by a session in the underlay




in a directory, . . .

E is attached to e,
or . . .

e is the location of
E.

at the endpoints,...

L uses S, or . . .

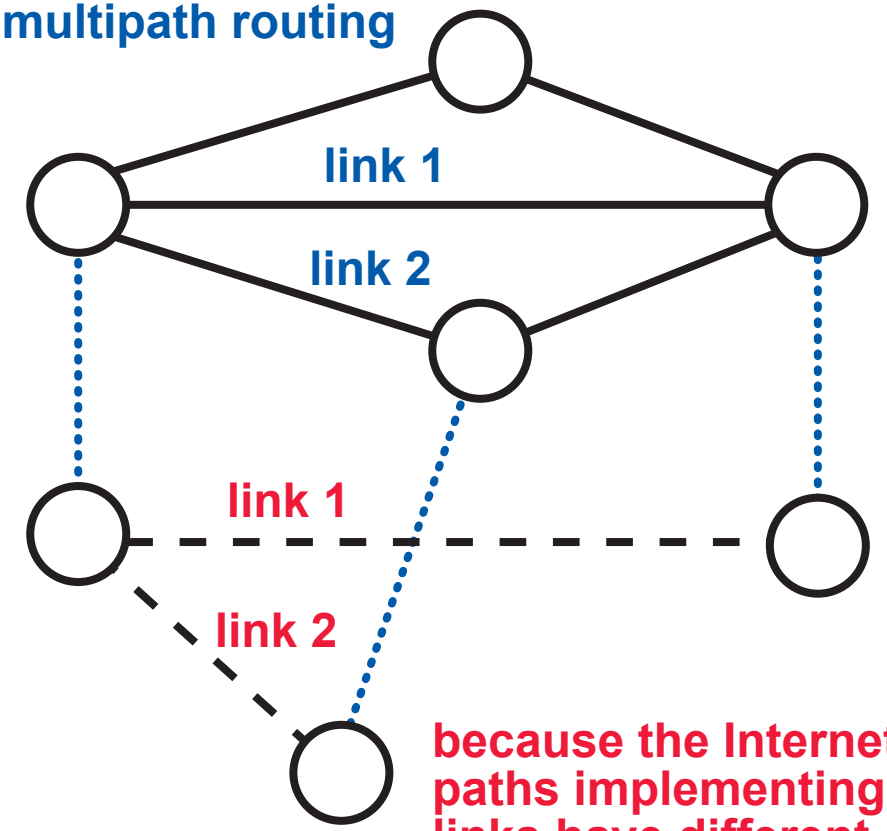
S implements L.

# VERTICAL COMPOSITION FOR IMPROVED PERFORMANCE

**GOAL: MAKE USE OF MULTIPLE IMPLEMENTATION PATHS (USE BEST OR ALL)**

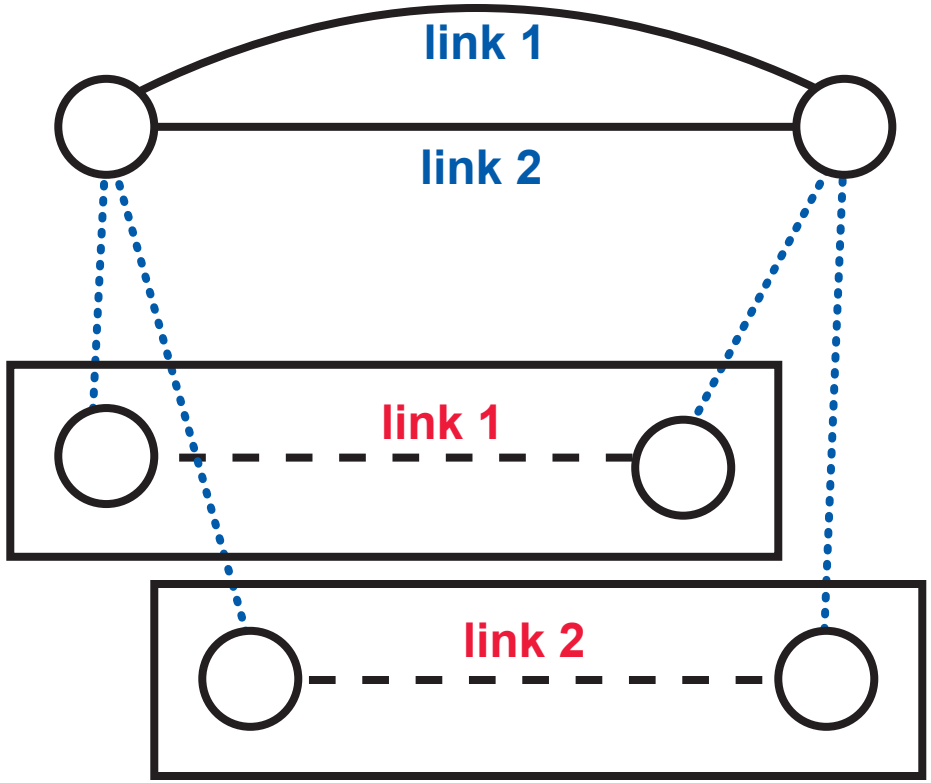
**WARNING: DOES NO GOOD IF THE MULTIPLE IMPLEMENTATION PATHS USE THE SAME RESOURCES!**

**RON uses explicit multipath routing**




**because the Internet paths implementing links have different endpoints, we expect them to use different resources**

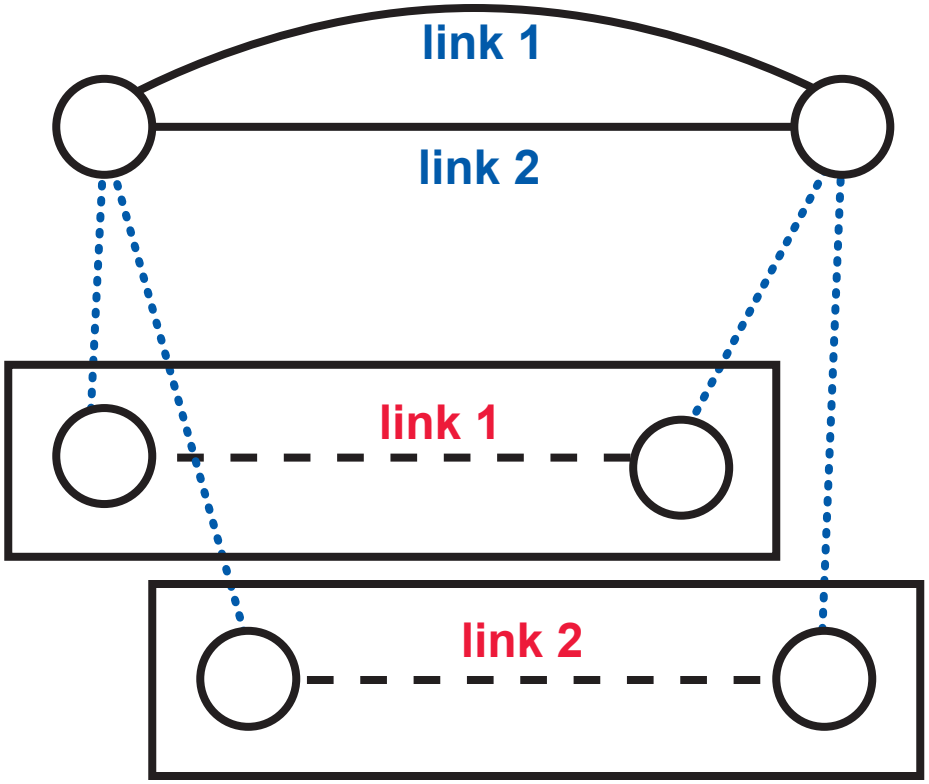
**could have overlay links with the same endpoints**




**links are implemented on different networks with different resources**

# VERTICAL COMPOSITION FOR IMPROVED PERFORMANCE

could have overlay links with the same endpoints




why is it necessary to have two links in the overlay?

links are implemented on different networks with different resources