Sidemeeting IETF 121

KIRA – Scalable Zero-Touch Routing

Roland Bless, Martina Zitterbart
Institute of Telematics, KIT

Zoran Despotovic, Artur Hecker
Huawei Research Center, Munich

KIRA: Kademlia-directed ID-based Routing Architecture

R. Bless, M. Zitterbart, Z. Despotovic and A. Hecker, „KIRA: Distributed Scalable ID-based Routing with Fast Forwarding“, 2022 IFIP Networking Conference (IFIP Networking), 2022, https://s.kit.edu/KIRA

Side Meeting – IETF Rules Apply

- **All public side meetings are subject to the IETF Note Well**
- participants are expected to follow the usual IETF policies on personal conduct, IPR disclosure obligations, etc.
- You are contributing to IETF work
- Policies on patents and code of conduct apply
- Disclose relevant IPR or do not make contributions
- Be respectful and courteous even/especially when you disagree

Agenda

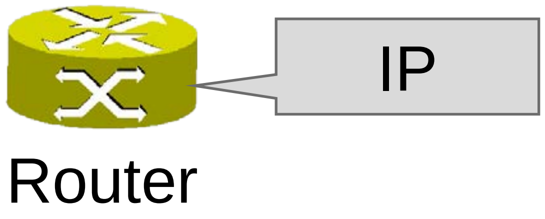
Karlsruhe Institute of Technology

- Introduction to KIRA (20min)
- Questions & Answers
- Discuss next steps
  - Supporting the IETF activities (BarBoF)

Introduction

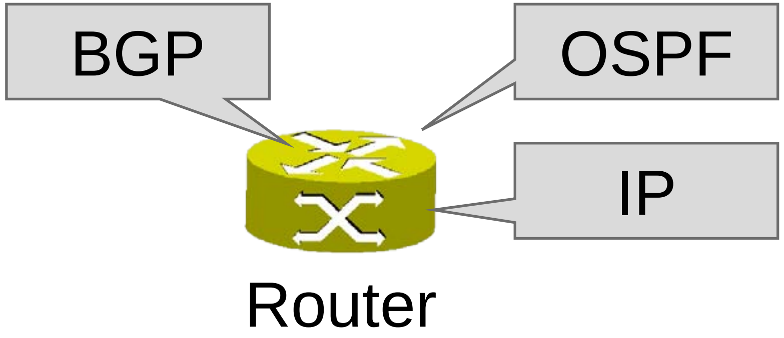
Internet Infrastructure…

- is becoming more complex
  - higher interdependencies of services
- must be reliable → resilient operation



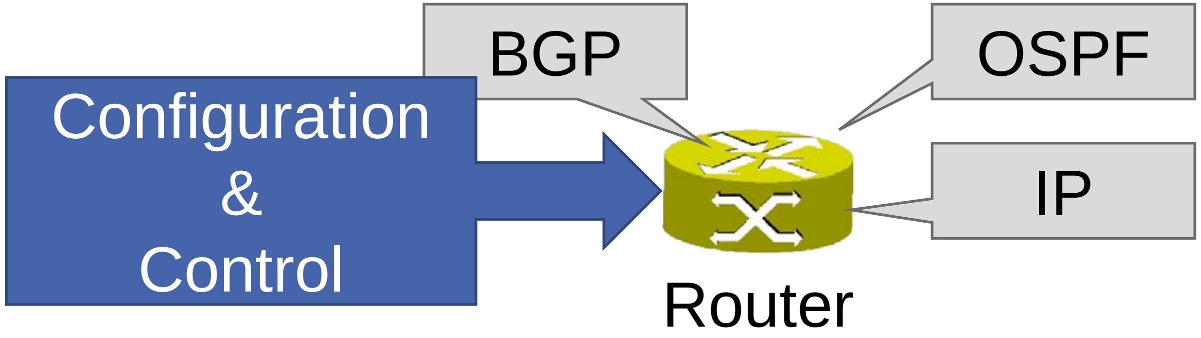
Internet Infrastructure…

- is becoming more complex
  - higher interdependencies of services
- must be reliable → resilient operation



Internet Infrastructure…

- is becoming more complex
  - higher interdependencies of services
- must be reliable → resilient operation



- Requires configuration via management/control plane

Working Control Plane Connectivity
=
Foundation for Resilient Internet Infrastructures

Controllability and Control Planes

Major Service Disruptions

Meta
Oct 2021

aws
Dec 2021

Controllability and Control Planes

Major Service Disruptions

Meta – Oct 2021

aws – Dec 2021

Services depend on resilient connectivity

Control plane connectivity inherently important

Controllability and Control Planes

Major Service Disruptions

Meta — Oct 2021

aws — Dec 2021

Services depend on resilient connectivity

Control plane connectivity inherently important

KIRA → provides **zero-touch resilient** control plane connectivity

Network services can always be restarted/restored from there

Infrastructure Complexity

A Decentralized SDN Architecture for the WAN

Alexander Krentsel, Google / UC Berkeley
Nitika Saran*, Cornell
Bikash Koley, Google
Subhasree Mandal, Google
Ashok Narayanan, Google
Sylvia Ratnasamy, Google / UC Berkeley
Ali Al-Shabibi, Google
Anees Shaikh, Google
Rob Shakir, Google
Ankit Singla, Google
Hakim Weatherspoon, Cornell
dsdn-sigcomm@google.com

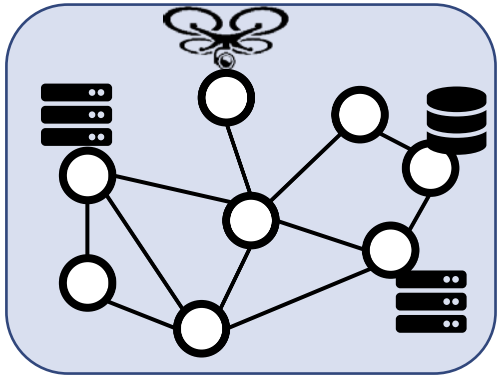
SIGCOMM 2024

The fundamental challenge in eliminating major failures lies in their **complexity**: they typically involve bugs or errors in multiple components spanning diverse teams and codebases, that **interact in unanticipated ways**. As a result, these outages persist, despite our extensive efforts to improve testing, diagnostics, change procedures, and verification.

…address complexity directly and focus on simplifying the network.

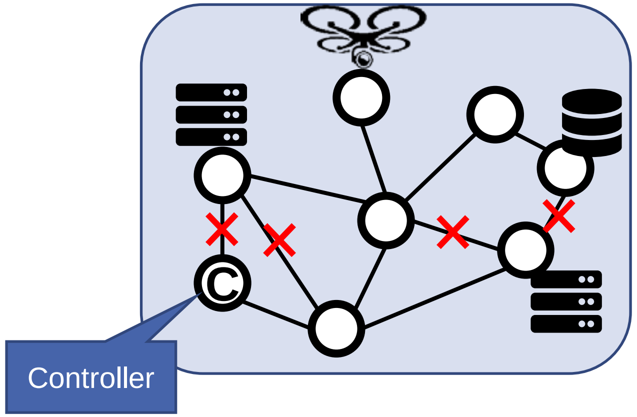
Control Planes of Future Networks Need to Support…

Interconnection of a Large Pool of Networked Resources

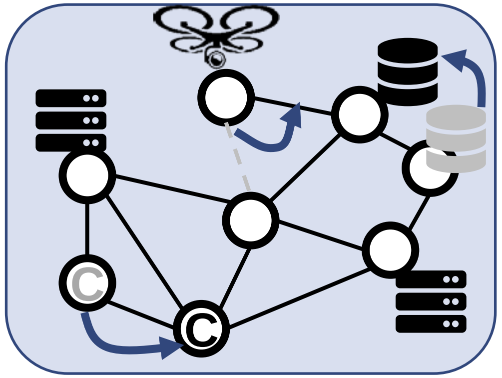


Compute, Storage, Network

Resilient Connectivity for Control Plane



Stable Addresses for Moving Resources



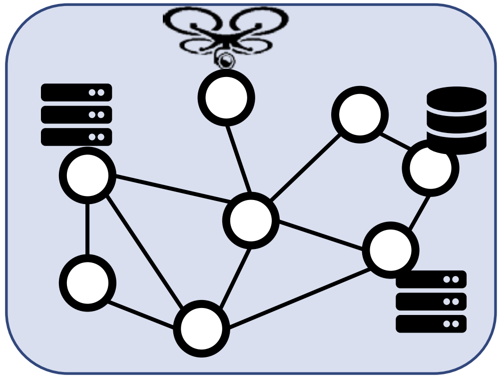
- Scalability
- In-band control
- High dynamics
- Multiple domains
- Various topologies

- Zero-touch
- Fast convergence
- Network split
- Nomadic networks

- ID-based addresses

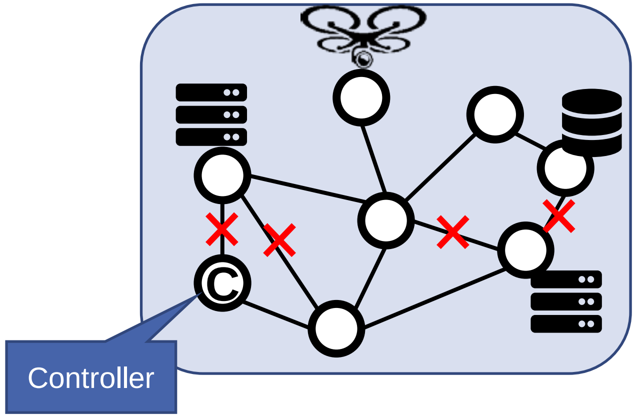
What KIRA achieves…

Interconnection of a Large Pool of Networked Resources

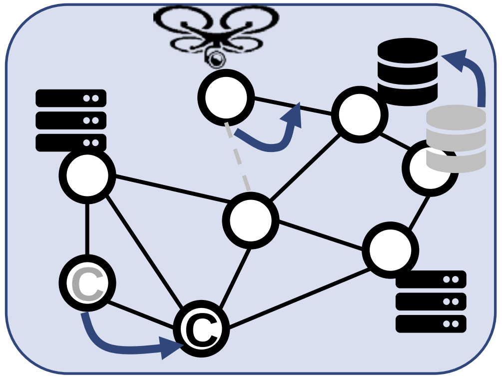


Compute, Storage, Network

Resilient Connectivity for Control Plane

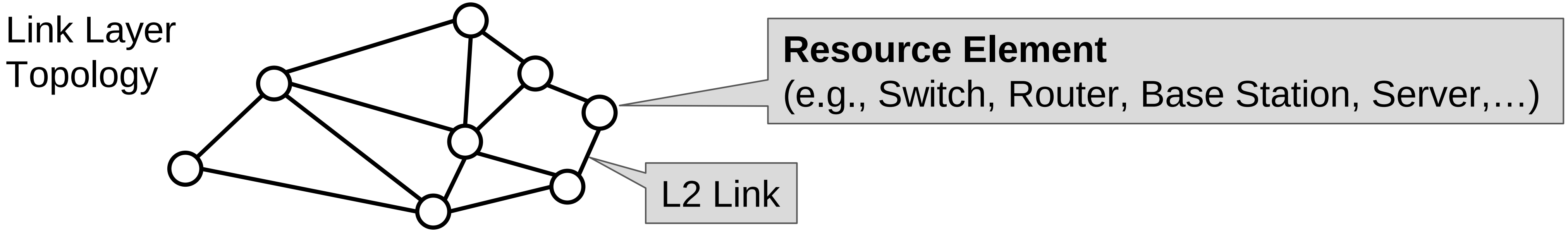


Stable Addresses for Moving Resources

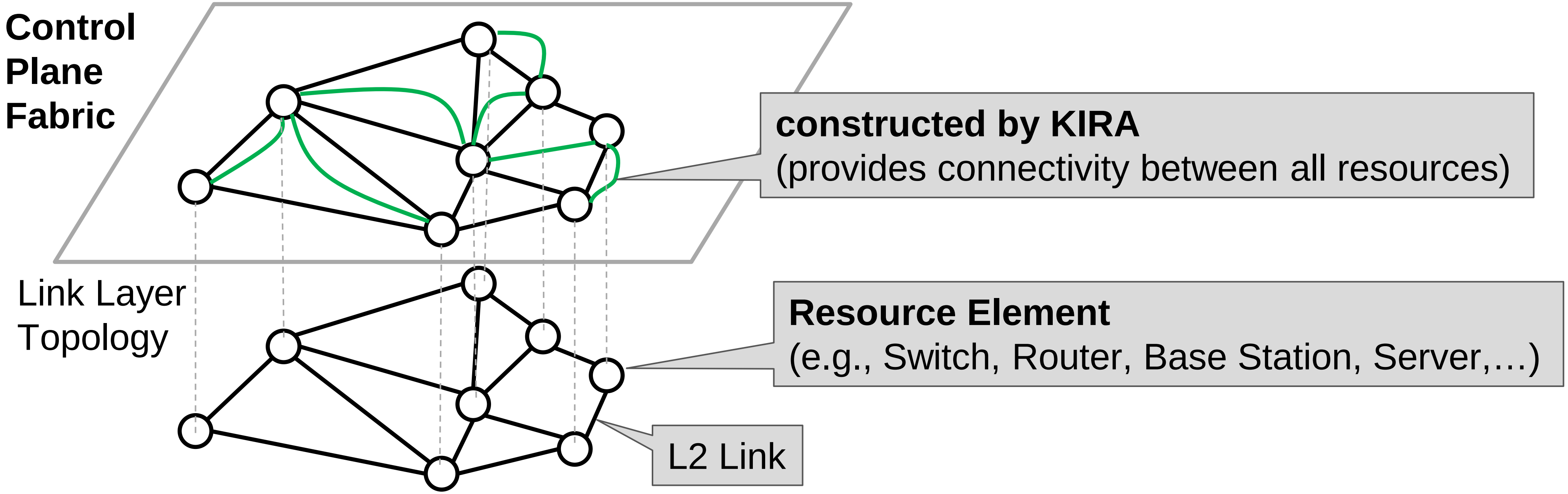


- KIRA provides (all-in-one)
  - Massive scalability (100,000s of nodes)
  - Zero-touch (no configuration + adaptation)
  - Dynamics: fast convergence, loop free
  - Topological versatility
  - Efficient routes
- Related Works (examples)
  - UIP: lacks dynamics and efficient routes
  - DISCO: lacks dynamics
  - RIFT, Data Center BGP/OSPF/IS-IS: specific topologies only, not ID-based
  - RPL: traffic concentration near root, zero-touch?
  - …

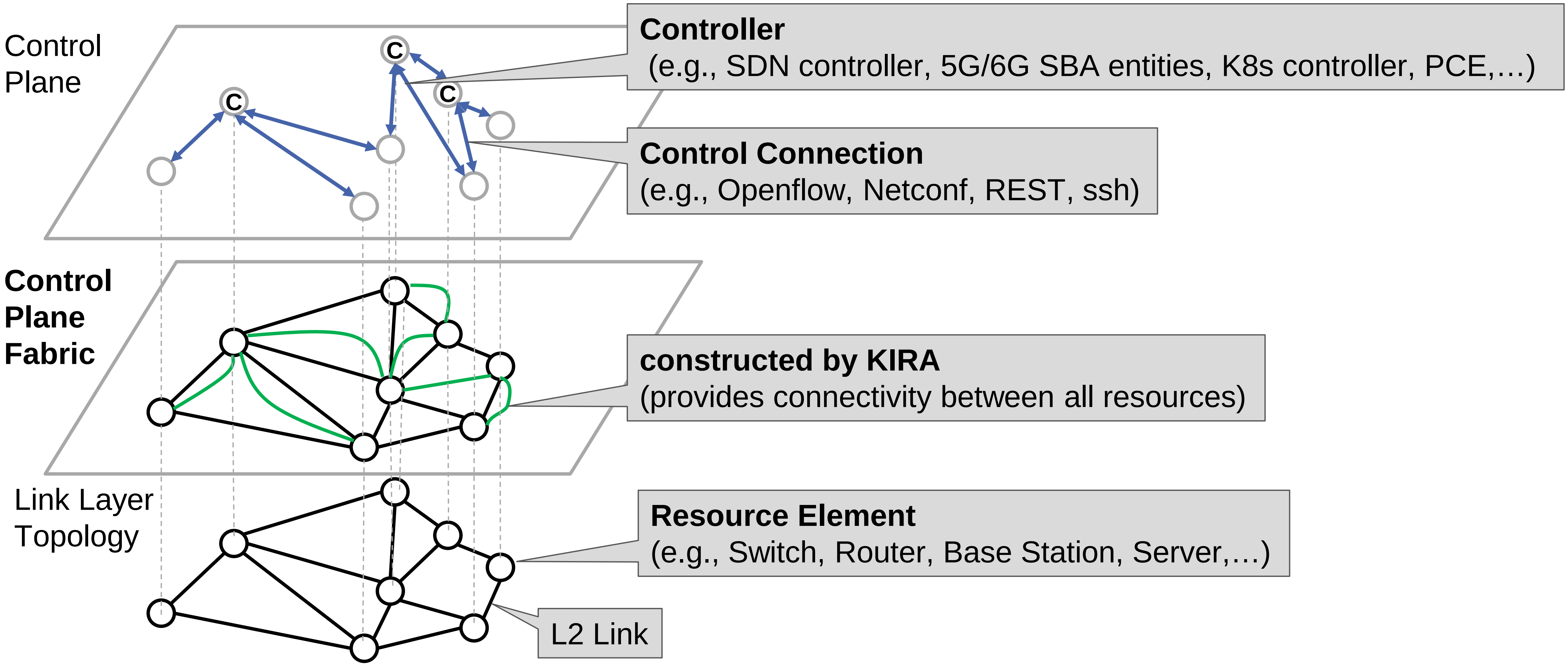
What KIRA provides…



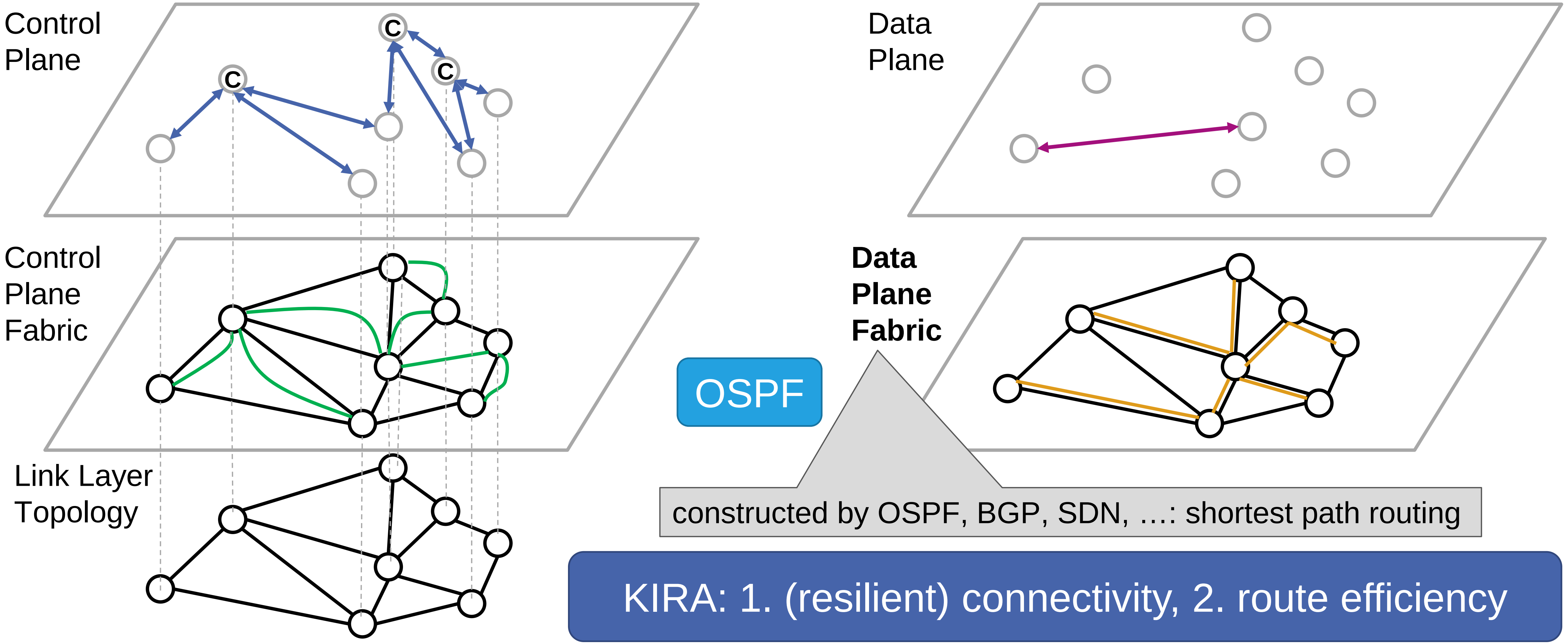
What KIRA provides…



What KIRA provides…



What KIRA provides…



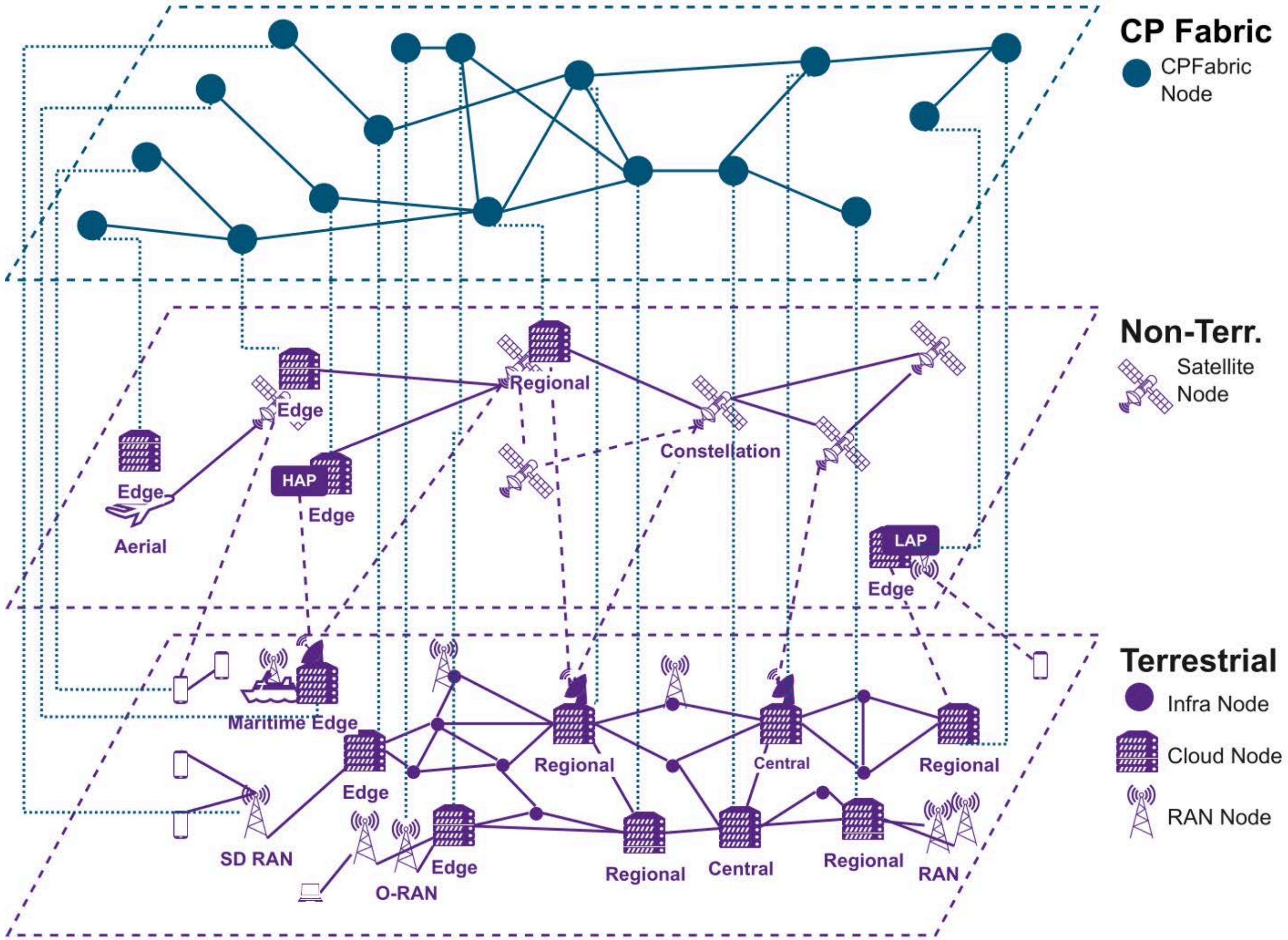
KIRA: 1. (resilient) connectivity, 2. route efficiency

Example

- KIRA bootstraps the control plane fabric
- Resources can register themselves to be found (e.g., run ANIMA on top)
- Topologically dependent routing can be built on top
  - e.g., divide the network into areas, assign and distribute prefixes, configure OSPF routers etc.
- Distributed controllers may claim control over some resource subset
- In case things go (horribly) wrong → restart from scratch

Use Case – 6G Control Plane

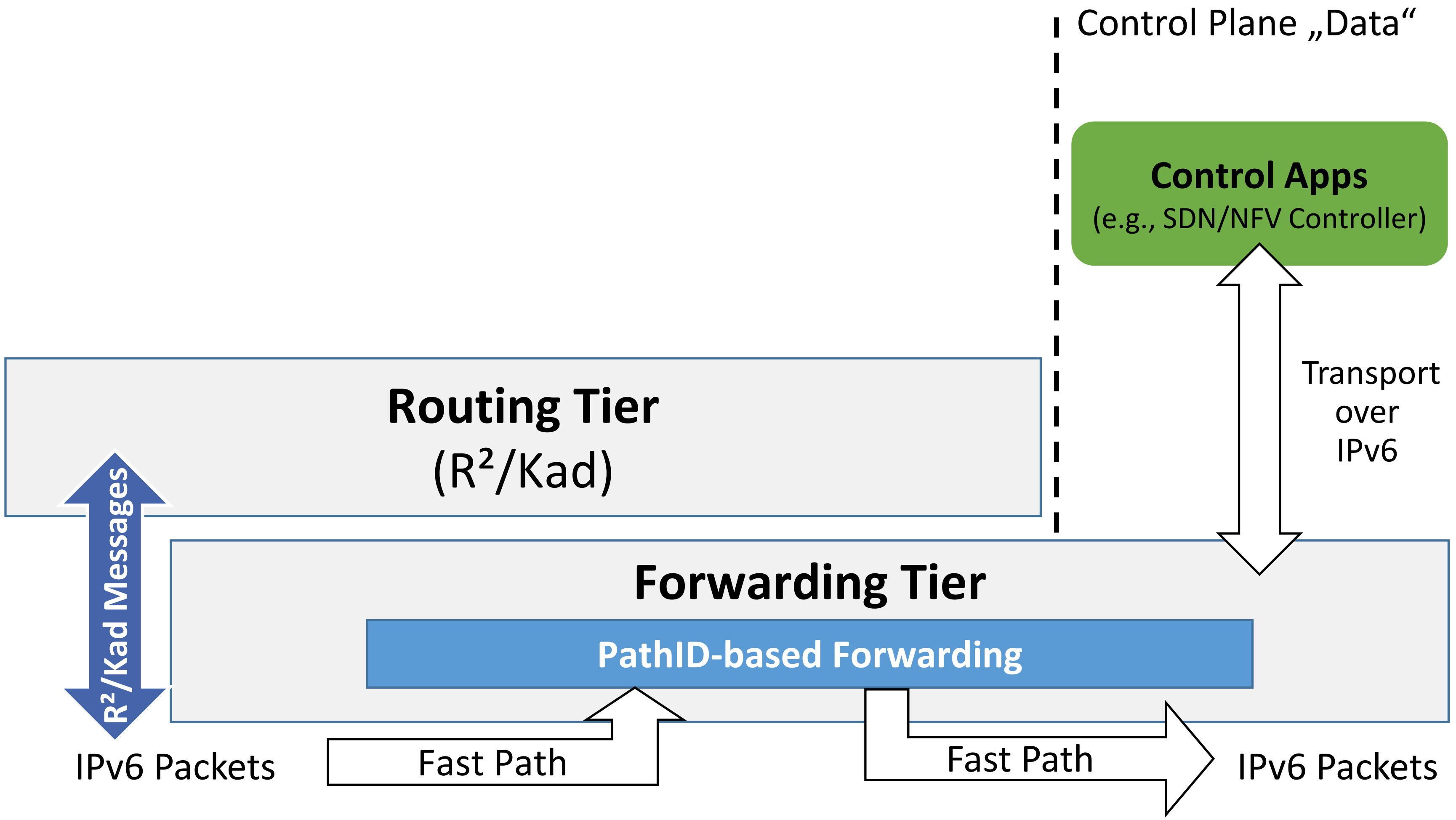
- Non-terrestrial Networks (Drones, Satellites)
  → dynamic and mobile
- Nomadic Networks
  → autonomous, self-organizing control plane
- 10^7 of base stations in China in a single provider network
  →scalability
- Single routing solution that interconnects everything for the Control Plane



Features

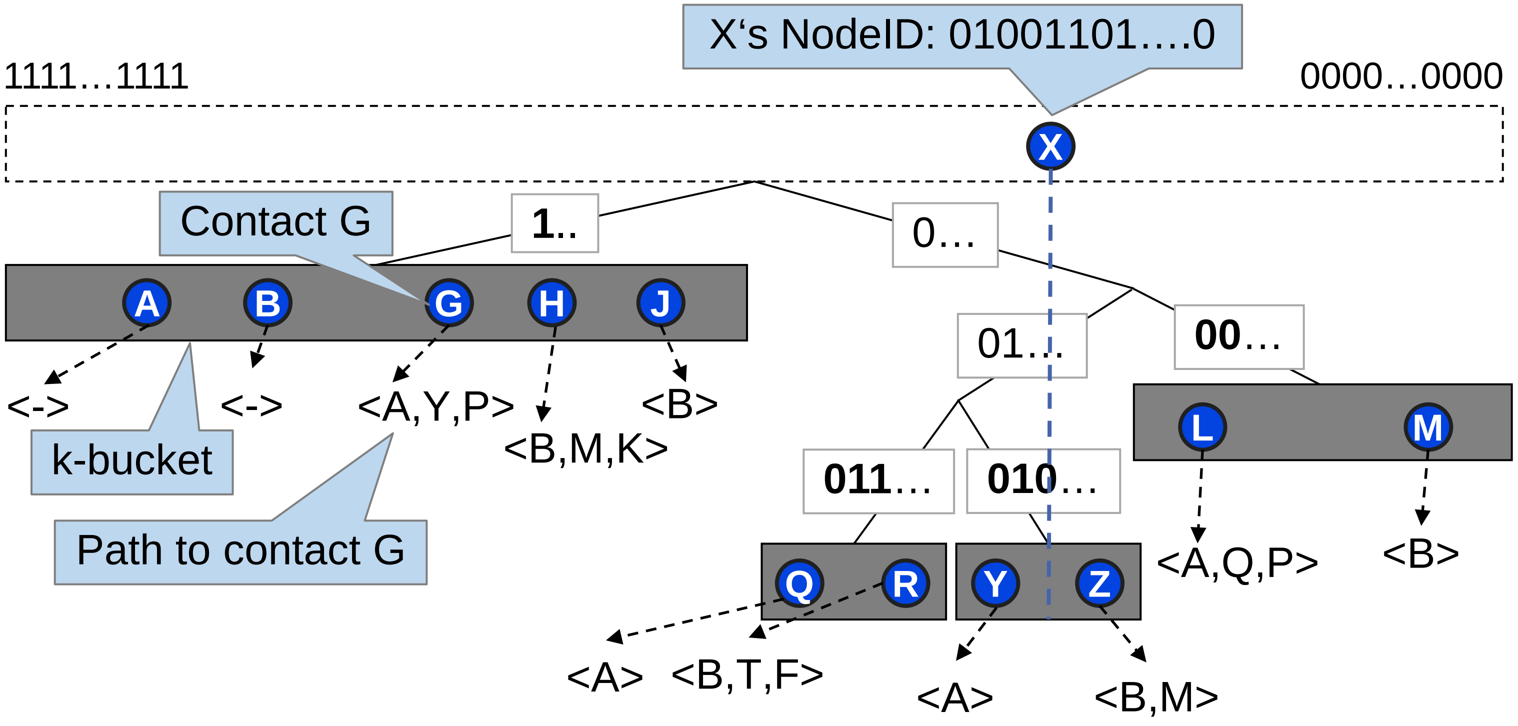
- Kademlia as ID-based Overlay (XOR metric)
  - 112bit NodeID
  - Similar to Virtual Ring Routing (that is less efficient)
  - Small routing tables – $O(\log n)$ n: number of nodes → Stretch
  - Supports multi-homing and mobility
- Proximity Neighbor Selection and Proximity Routing
- Path Discovery (First Path, Later Path) + Path Rediscovery (Dynamics)
- Loop-free even during convergence
- Source routing for routing messages
- PathID-based forwarding for data
  - no special forwarding hardware required (IPv6 support)

Architecture



R²/Kad – Routing Table

- Tree of buckets holding up to k contacts (e.g., k=40)
  - Arranged by XOR distance
  - Bucket split if it contains own NodeID
- Path vectors are stored for each contact
- Efficient routes
  - Shorter routes preferred
- Size k of k-buckets can be set per node → flexible memory / stretch trade-off
- Routing table size: $O(l_G \log n)$, l_G average path length



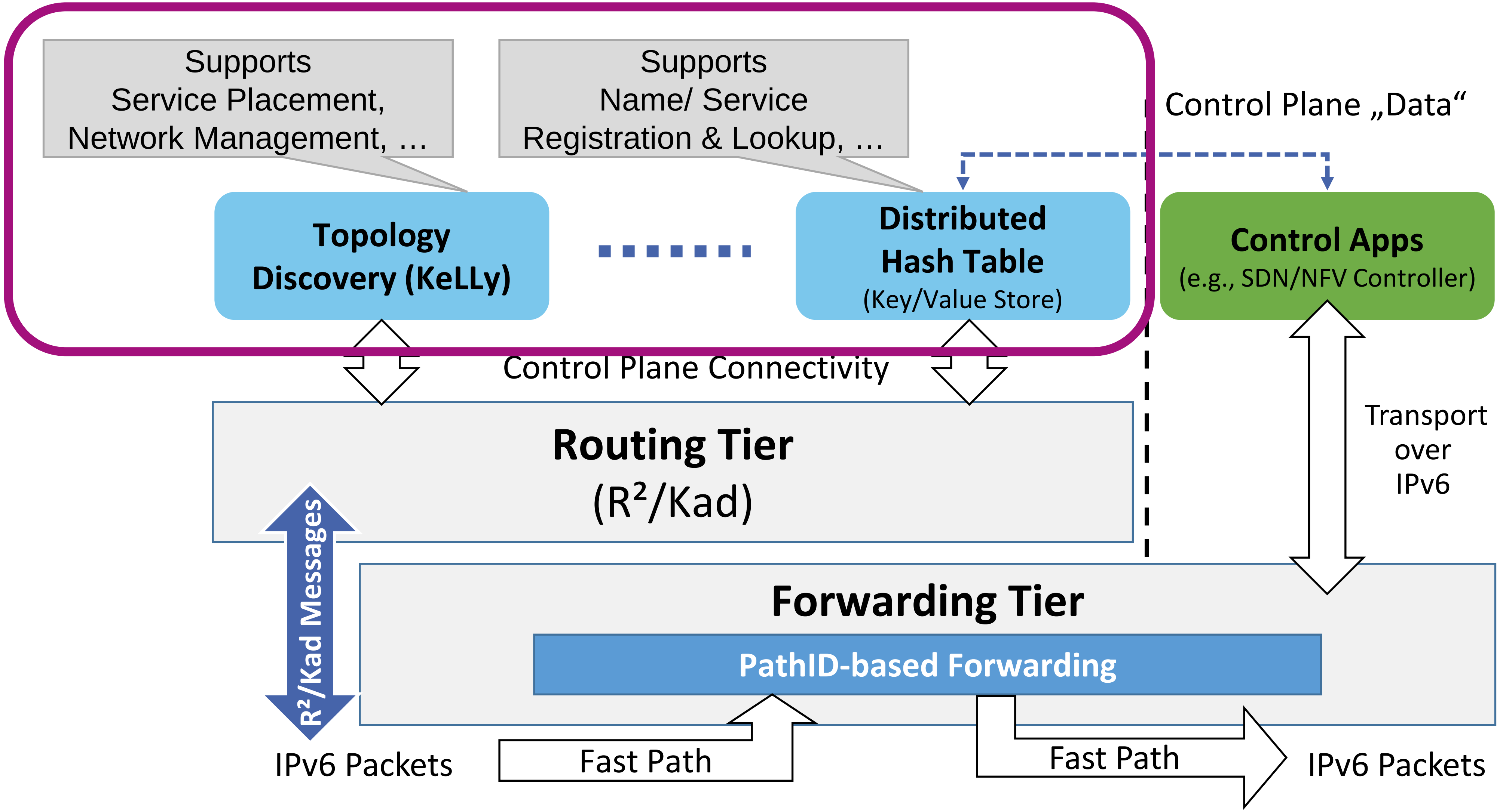
Forwarding Tier

- Approach: replace source routes with PathIDs
  - PathID(<A,Q,M,Z>)= Hash(A | Q | M | Z)
- Need PathID in addition to NodeIDs → encapsulation (e.g., GRE, SRv6)
- Use PathID as unique label for path segment → Label Switching
- Precalculate PathIDs for 2-hop (physical) vicinity
- Explicit path setup for paths ≥ 6 hops
- Cost?
  - 100,000 nodes powerlaw topology:
    99th percentile forwarding table ~ 2500 entries
- Uses existing forwarding plane mechanisms
- Works with Longest Matching Prefix, nftables, SDN, eBPF etc.

Further Features (Continued)

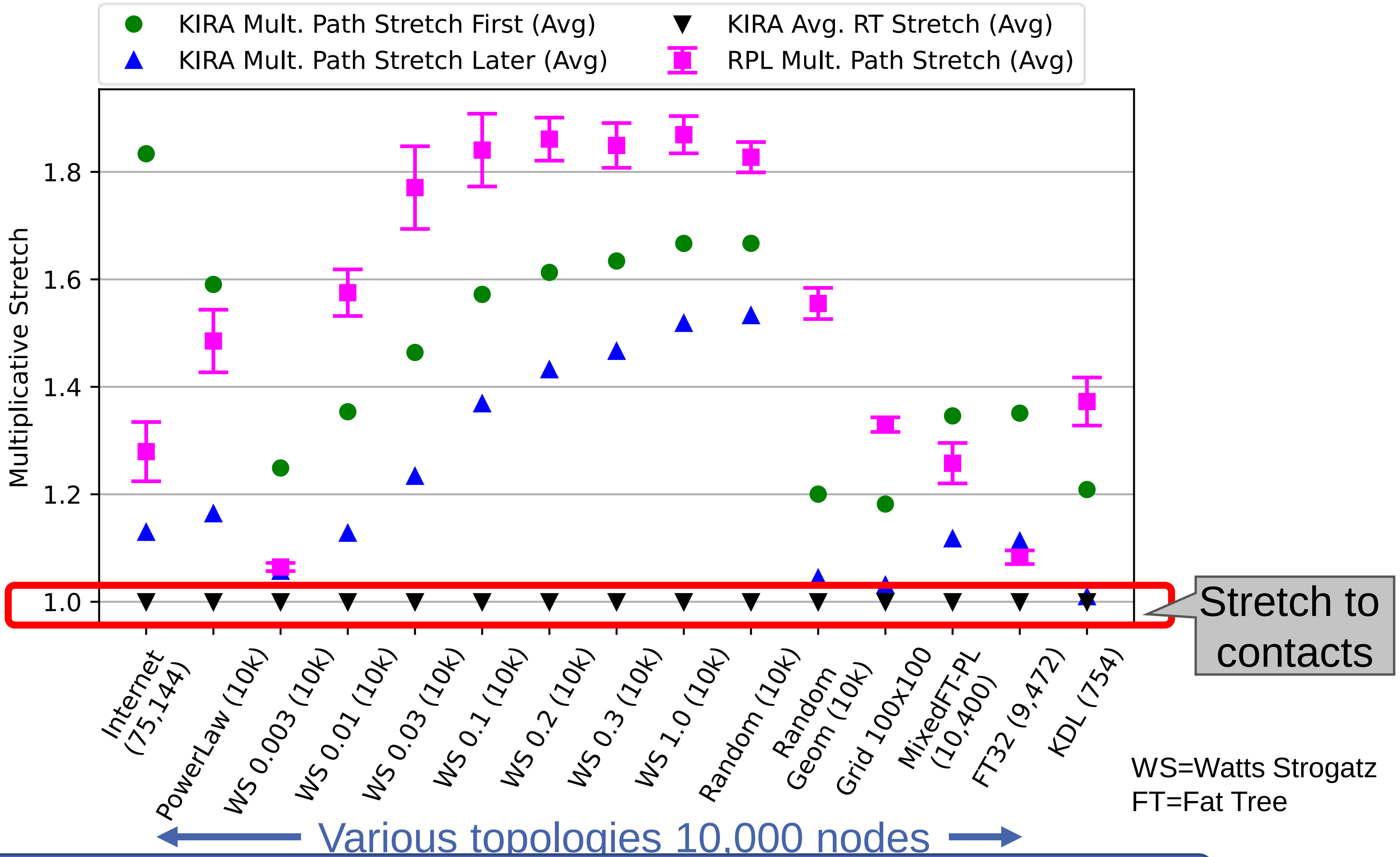
- Per-node decision on stretch/memory trade off
- Multi-path capable (due to small routing tables)
  - includes fast reroute
- Supports different routing metrics
- Built-in route flapping prevention
- Special end-system mode (non-routing)
- Supports Domains (topological, organizational)
- Can provide additional services like
  - key-value store, name registration and resolution service
  - KeLLy – efficient topology discovery
  - Structured information dissemination to all nodes

Architecture



Stretch in Different Fixed Network Topologies

- Multiplicative Stretch
- RPL-ACP:
  - Storing-mode
  - Single DODAG
  - Single DODAG version



WS=Watts Strogatz
FT=Fat Tree

Low stretch across various topologies + Shortest paths to contacts

The Plan…

- Status Quo
  - Networks and their operations will become more complex
  - Increasing number of interdependencies with growing scale (RFC 3439)
  - Root cause analysis extremely difficult
  - Need resilient operations
- Let‘s
  - make the Internet better…
  - make KIRA an IETF standard so that it can be built into networked devices
  - have a resilient foundation for network control that
    - you can rely on (cannot be broken by misconfiguration)
    - you can always use to restore your network to well-known states

Please Support KIRA!

- Internet-Draft https://datatracker.ietf.org/doc/draft-bless-rtgwg-kira/
  - Updated to -01: added action descriptions for sending/receiving messages
  - Please provide feedback!
  - Want IETF expertise

- Running Code available
  - Hackathon IETF 121
  - Native Routing Daemon Linux (Rust)
    → Zero-touch IPv6 Connectivity
    - Forwarding Tier uses nftables
    - Alternative eBPF implementation nearly complete

https://s.kit.edu/KIRA

Q&A

Next Steps

Next IETF Steps for KIRA

Karlsruhe Institute of Technology

- Raise community interest: spread the word…
- Move Base Internet-Draft forward (at least to experimental) in RTGWG
- Need feedback!
- Need implementers!

- Think big: WG Topics
  - Base Specification
    - different encapsulation possibilities
  - Security mechanisms
  - „API“ + protocols for supporting services (e.g., DHT, KeLLy, etc.)
  - Specially tweaked versions for IoT, MANETs etc.

Backup Slides

KIRA – Main Components

Karlsruhe Institute of Technology

- Routing Tier → connectivity

R²/Kad

- ID-based addresses
- Source routing
- On top of link layer

- Forwarding Tier → optimization

PathID-based Forwarding

- Eliminates source routing
- Label switching approach
- Reduces overhead

KIRA – Main Components

- Routing Tier → connectivity

Path Discovery, Routing

R²/Kad

Failure Recovery

- **ID-based addresses**
- **Source routing**
- **On top of link layer**

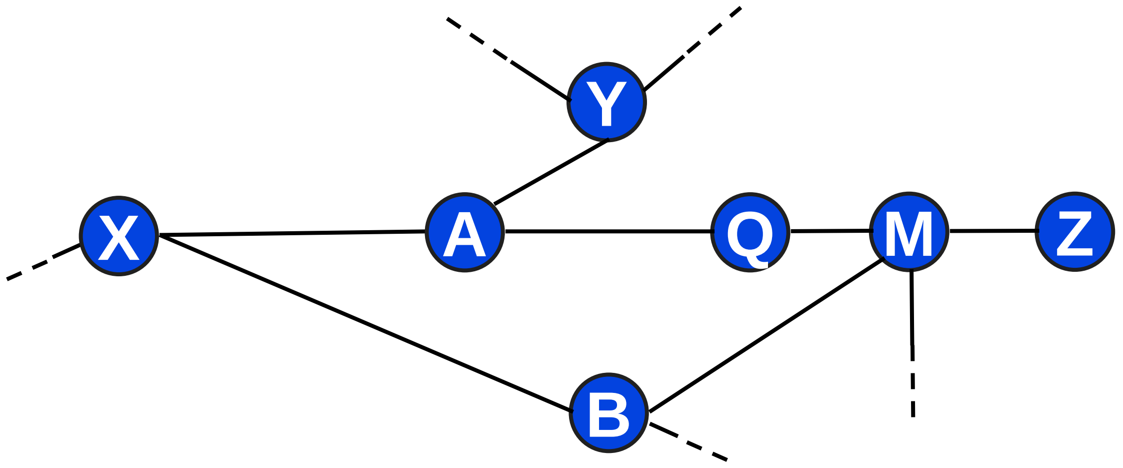
- Forwarding Tier → optimization

PathID-based Forwarding

- **Eliminates source routing**
- **Label switching approach**
- **Reduces overhead**

R²/Kad – Path Discovery

- Each node
  - randomly chooses its NodeID (Overlay)
  - explores its 2-hop vicinity (Underlay)
    - X learns contacts A, Y, Q, B, M, …

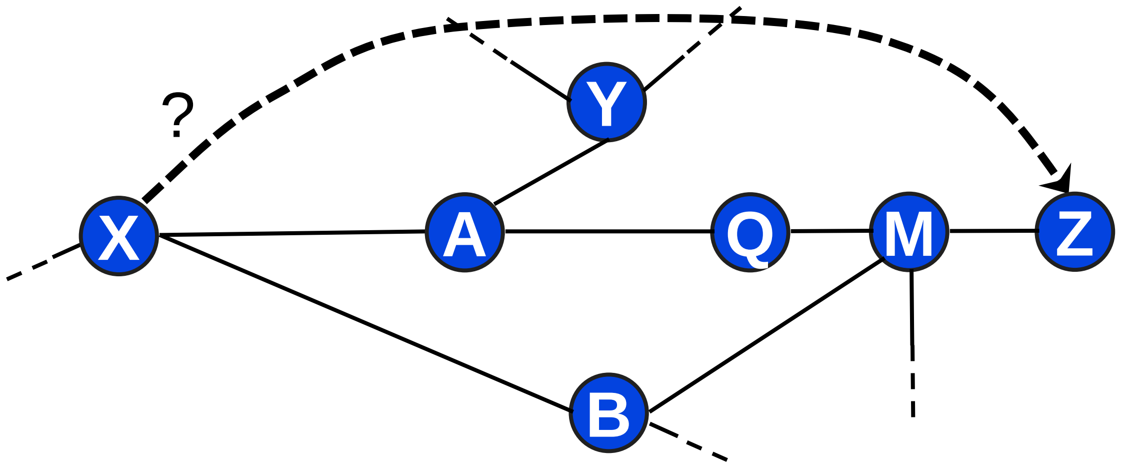


R²/Kad – Path Discovery

Karlsruhe Institute of Technology

Path Discovery, Routing | R²/Kad | Failure Recovery

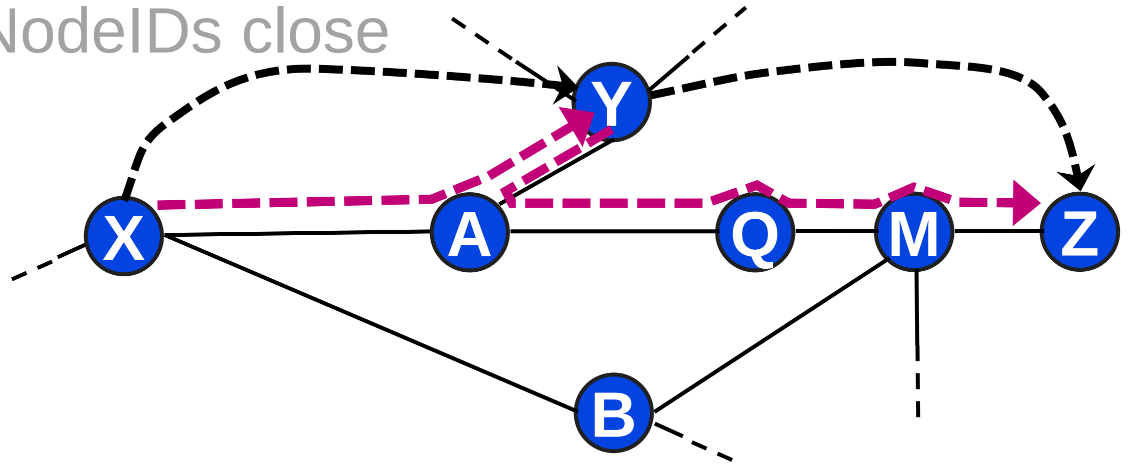
- Each node
  - randomly chooses its NodeID (Overlay)
  - explores its 2-hop vicinity (Underlay)
    - X learns contacts A, Y, Q, B, M, …
- X: path to Z?
- Approach:
  construct underlay routes
  by using the NodeID-based overlay
  - Source route to contact that is
    ID-wise closest to destination NodeID (→ recursively)
  - Distance of NodeIDs: XOR metric $d(X,Y) = X \oplus Y$
    - Longer shared prefix → closer



R²/Kad – Path Discovery Example

Path Discovery, Routing
R²/Kad
Failure Recovery

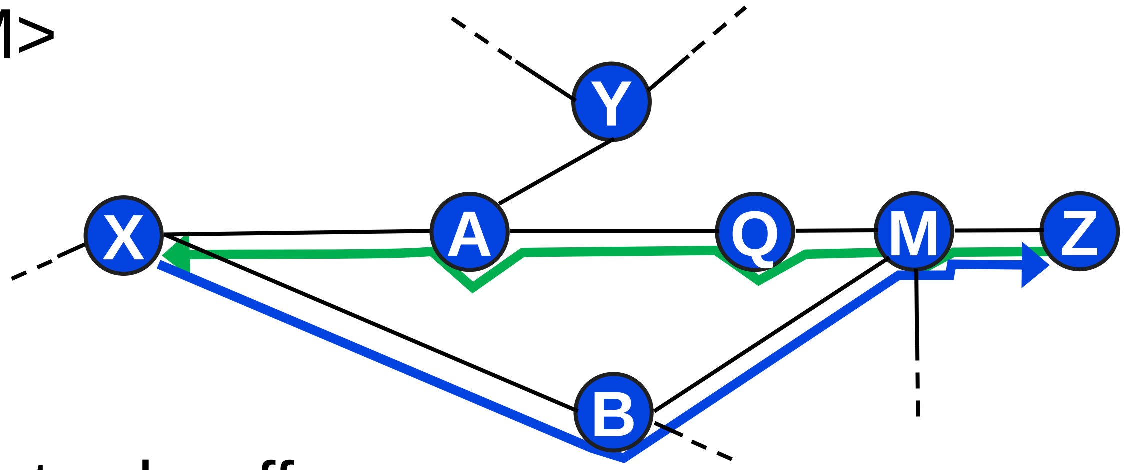
- X sends FindNodeReq to contact closest to NodeID Z
  - Example: letters close in alphabet ↔ NodeIDs close
  - Next (overlay) hop: Y
- X → Y via source route <A>
- Assume Y knows Z already
- Y → Z via source route <A,Q,M>
- FindNodeReq records complete route <X,A,Y,A,Q,M>
  - incurs path stretch: $\frac{|\text{selected path}|}{|\text{shortest path}|}$



R²/Kad – Path Discovery Example

- Shortened recorded route <A,Q,M> is returned to X in FindNodeRsp
- Later packets use shorter route <B,M>
  - if X already knows M via <B>

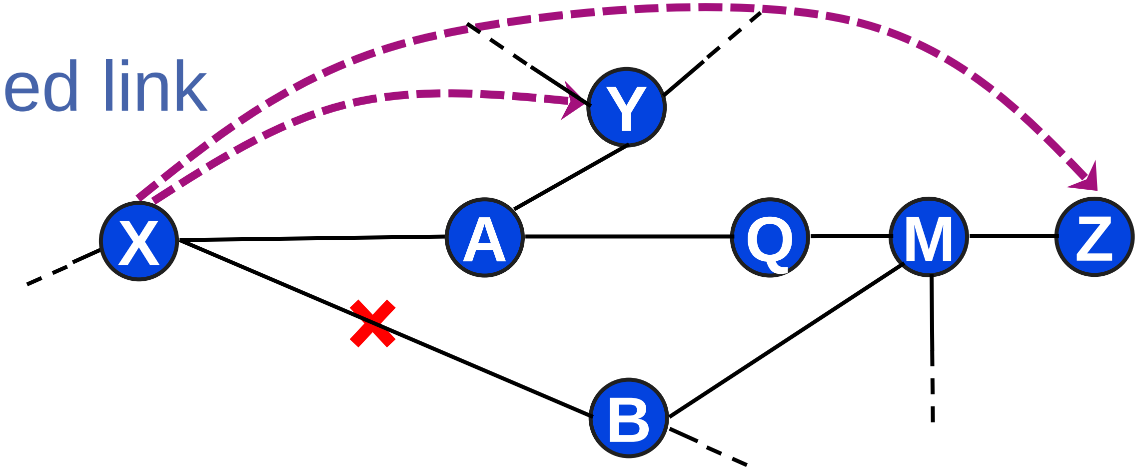
Initial stretch can be reduced for later packets!



- R²/Kad offers flexible memory/stretch trade-off…

R²/Kad – Dynamics: Rediscovery Procedure

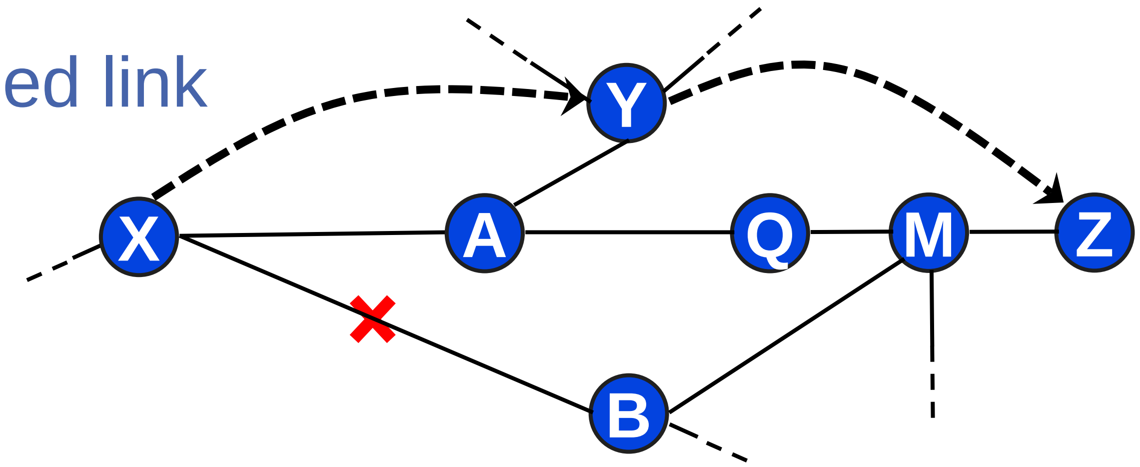
- Detection of node/link failure in the underlay
- Two step strategy
  1.) inform ID-wise neighbors about failed link
  2.) …



R²/Kad – Dynamics: Rediscovery Procedure

Path Discovery, Routing | R²/Kad | **Failure Recovery**

- Detection of node/link failure in the underlay
- Two step strategy
  1.) inform ID-wise neighbors about failed link
  2.) rediscover alternative paths via overlay routes (includes "Not Via" information)
- Validity
  - State sequence numbers
  - Path information age
- Periodically
  - probe contacts for broken paths
  - lookup own NodeID



KIRA – Main Components

- Routing Tier → connectivity

Path Discovery, Routing

Routing Table

R²/Kad

Failure Recovery

- ID-based addresses
- Source routing
- On top of link layer

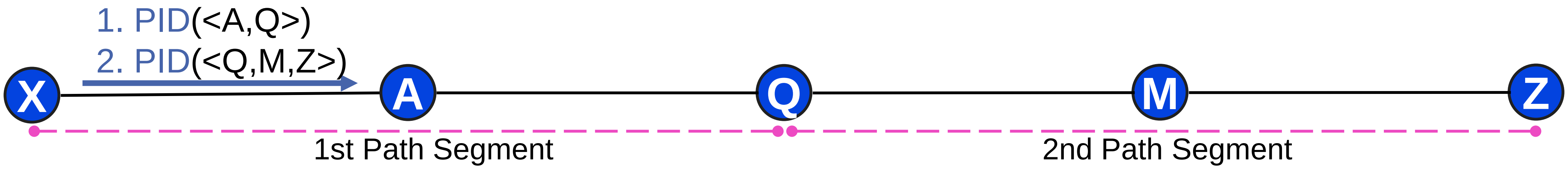
- Forwarding Tier → optimization

PathID-based Forwarding

- Label Switching Approach
- Eliminates Source Routing
- Reduces Overhead

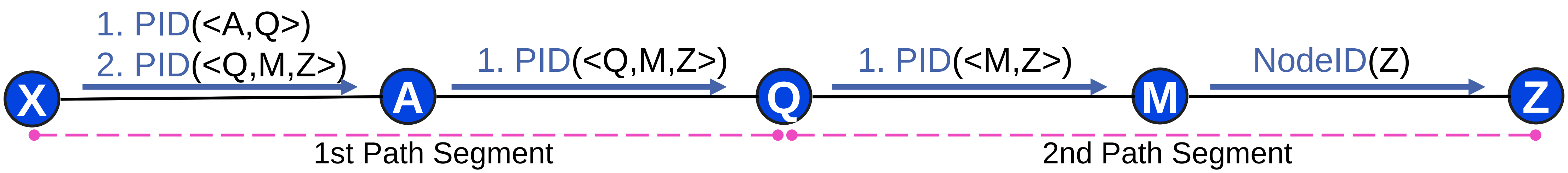
Forwarding Tier – Fast Forwarding

- Get rid of source routes for control plane traffic
  - Reduce per packet overhead
- Approach: replace source routes with PathIDs
  - PathID(<A,Q,M,Z>)= Hash(A | Q | M | Z)
- Use PathID as unique label for path segment → Label Switching



Forwarding Tier – Fast Forwarding

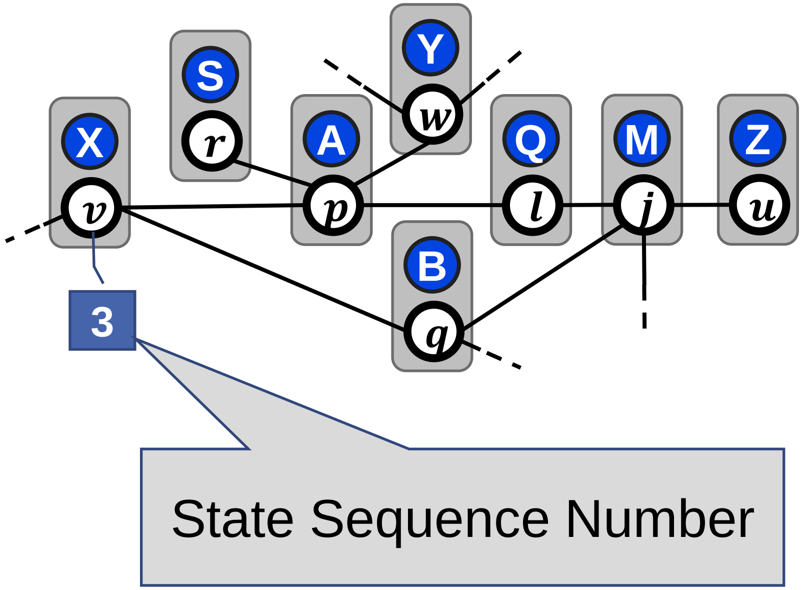
- Get rid of source routes for control plane traffic
  - Reduce per packet overhead
- Approach: replace source routes with PathIDs
  - PathID(<A,Q,M,Z>)= Hash(A | Q | M | Z)
- Use PathID as unique label for path segment → Label Switching



- Precalculate PathIDs for 2-hop (physical) vicinity
- Explicit path setup for paths ≥ 6 hops

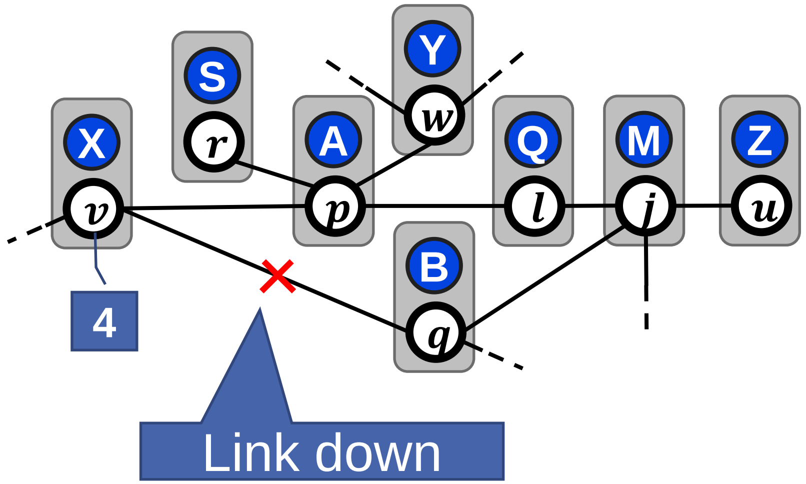
State Sequence Numbers

- Per Node: State Sequence Number
  - reflects changes in node's physical neighbor set
  - Link down
  - Link up (also detecting a new node)
- 32-bit
  - Wrap around and special comparison:
    - $s < s' \bmod 2^{32}$ if $0 < (s' - s) \bmod 2^{32} < 2^{31}$
- Get periodically synchronized
- Node crashes
  - Node either uses new NodeID after restart
  - or, node stores NodeID and State Sequence Number across restart



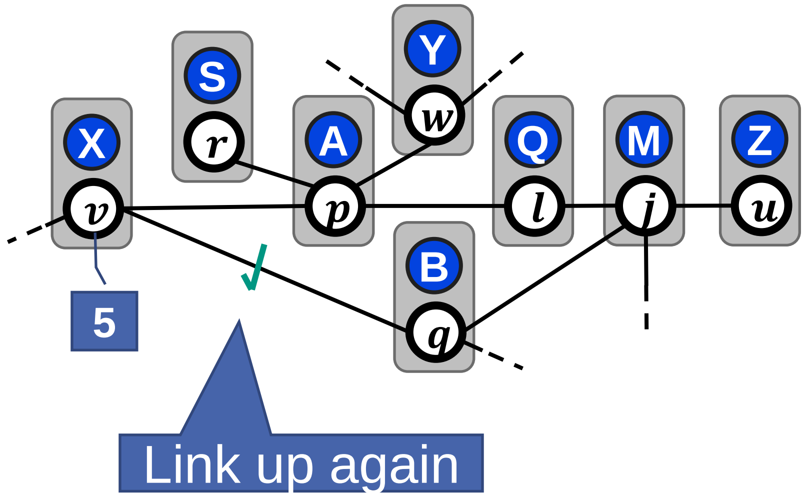
State Sequence Numbers

- Per Node: State Sequence Number
  - reflects changes in node's physical neighbor set
  - Link down
  - Link up (also detecting a new node)
- 32-bit
  - Wrap around and special comparison:
    - $s < s' \bmod 2^{32}$ if $0 < (s' - s) \bmod 2^{32} < 2^{31}$
- Get periodically synchronized
- Node crashes
  - Node either uses new NodeID after restart
  - or, node stores NodeID and State Sequence Number across restart



State Sequence Numbers

- Per Node: State Sequence Number
  - reflects changes in node's physical neighbor set
  - Link down
  - Link up (also detecting a new node)
- 32-bit
  - Wrap around and special comparison:
    - $s < s' \bmod 2^{32}$ if $0 < (s' - s) \bmod 2^{32} < 2^{31}$
- Get periodically synchronized
- Node crashes
  - Node either uses new NodeID after restart
  - or, node stores NodeID and State Sequence Number across restart



End-system Mode

- End-systems do not route, but may be multi-homed and mobile
- Reduce overhead by not transmitting routing updates to/from end-systems
- Routers are responsible to keep information on end-system reachability

KIRA – Domain Scopes

- Domain Scopes
  - Global, Organizational (e.g., ASes), Topological
  - KIRA nodes keep their NodeID across domains!