SkipNet: A Scalable Overlay Network with Practical Locality Properties

N.J.A. Harvey, M.B. Jones, S. Saroiu,
M. Theimer, and A. Wolman

Presented by:
David Hadaller
University of Waterloo

Outline

- Contributions
- How it Works
- Locality Properties
- Performance
- Beyond SkipNet

Motivation

- DHTs have achieved
  - Scalable and decentralized infrastructure
  - Uniform and random load and data distribution
- Outstanding problems
  - Data may be stored far from its users
  - Data may be stored outside its domain
  - Local accesses leave local organization

Motivation (cont'd)

- Data Controllability
  - Organizations want control over their own data
  - Even if local data is globally available
- Manageability
  - Data control allows for data administration, provisioning and manageability
  - Data center/cluster = constrained set of nodes
  - CLB ensures load balance across data center/cluster

Motivation (cont'd)

- Security
  - Content and path locality are key building blocks for dealing with certain external attacks
- Data availability
  - Local data survives network partitions
- Performance
  - Data can be stored near clients that use it

Contributions

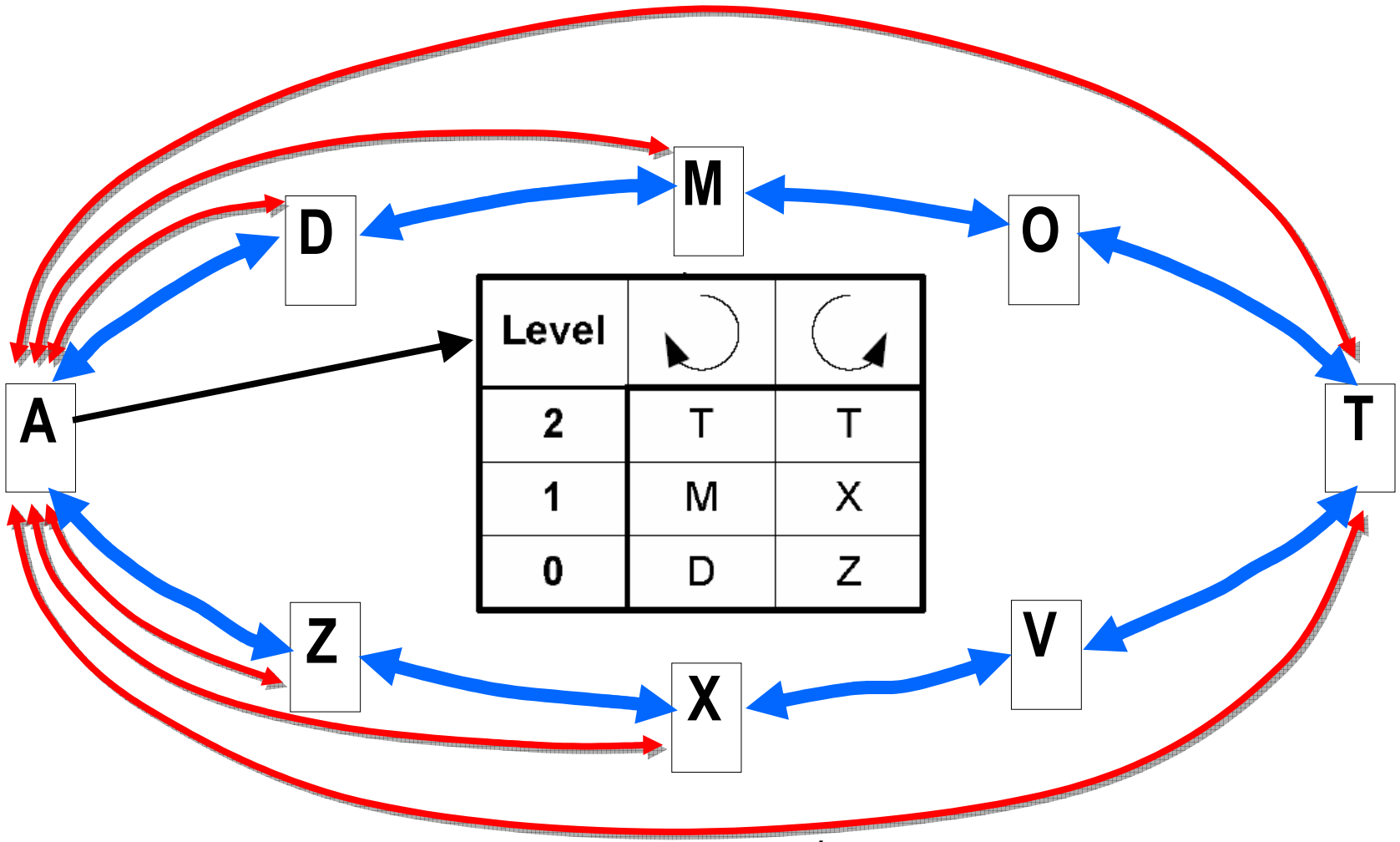
- SkipNet provides all of the above
  - Content locality
    - Ability to explicitly place data
  - Path locality
    - Ability to guarantee that local traffic remains local
  - Constrained Load Balancing
    - Data balanced across a subset of nodes
  - Plus all the benefits of standard DHTs
    - e.g. O(logN) search time

Outline

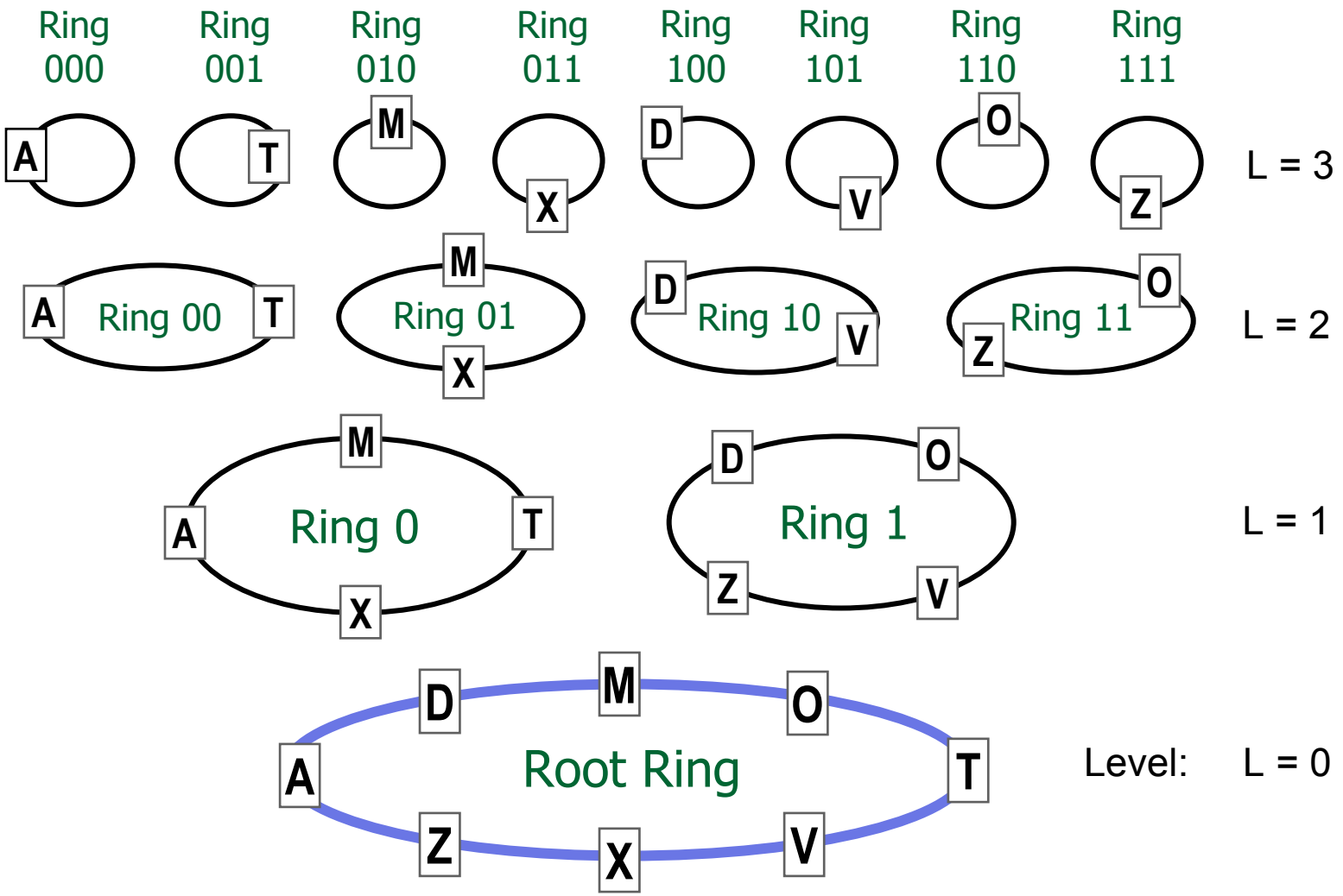
- Contributions
- **HOW IT WORKS**
- Locality Properties
- Performance
- Beyond SkipNet

How it Works

- Built on the ideas of previous DHTs
  - Chord, Pastry
- Key property: two address spaces
  1. Name ID space: nodes are sorted by their names (e.g. DNS names)
  2. Numeric ID space: nodes are randomly distributed

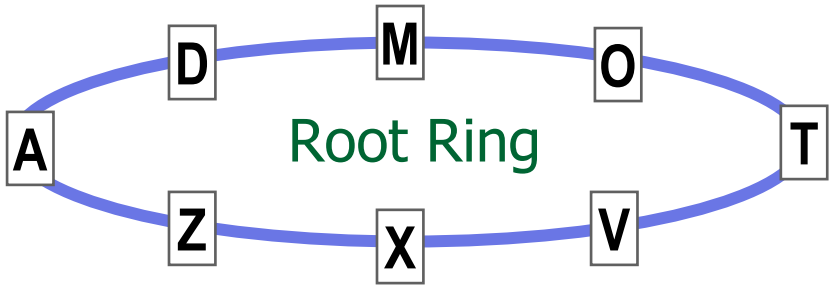


- Pointers at level *h* skip over 2^h nodes
- Nodes are ordered by names


Ring 000
Ring 001
Ring 010
Ring 011
Ring 100
Ring 101
Ring 110
Ring 111
A
T
M
X
D
V
O
Z
L = 3
A
Ring 00
T
M
Ring 01
X
D
Ring 10
V
Z
Ring 11
O
L = 2
M
A
Ring 0
T
X
D
O
Ring 1
Z
V
L = 1
D
M
O
A
Root Ring
T
Z
X
V
Level:
L = 0

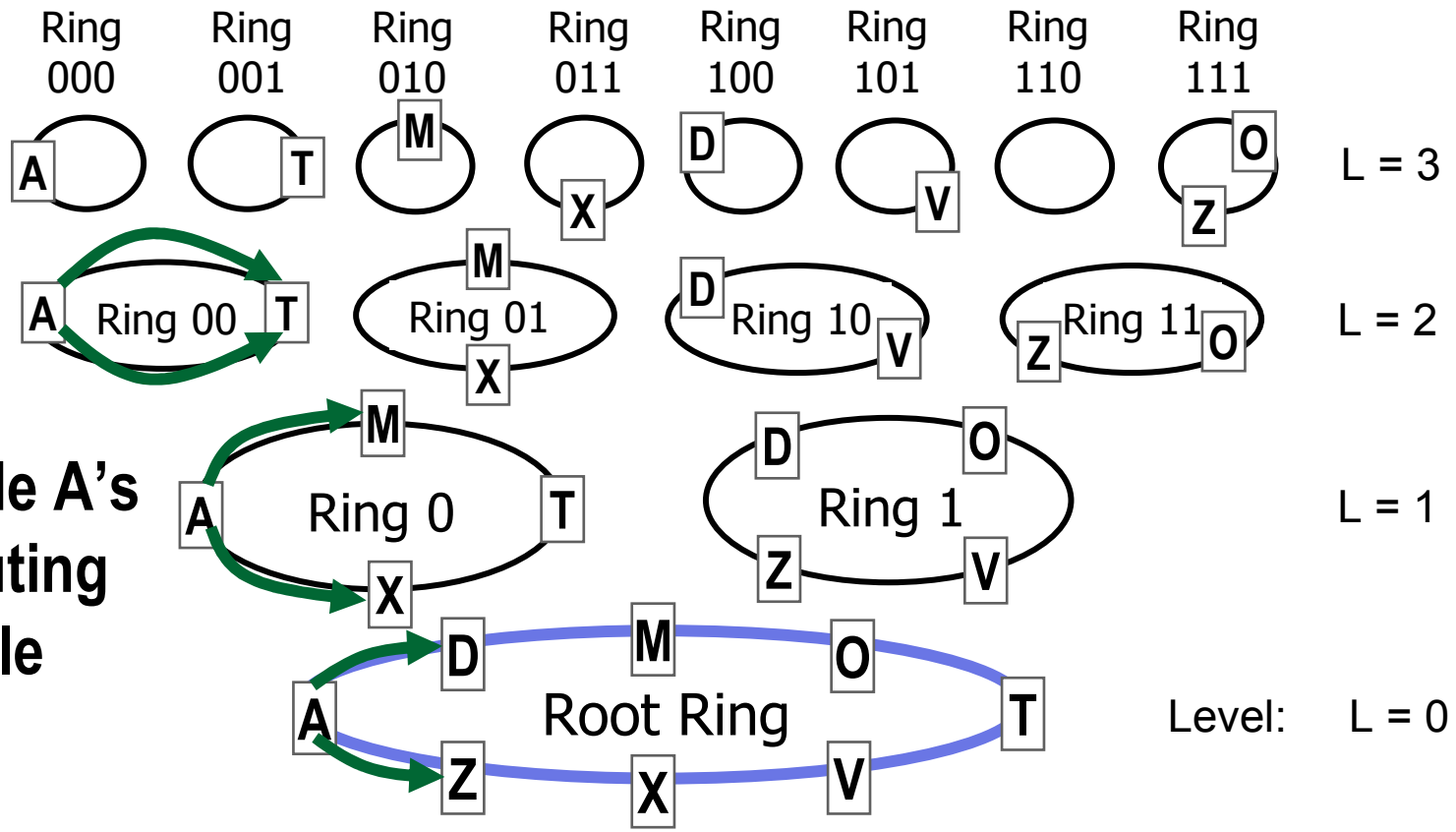
How it Works

- What counts is the Root Ring, the rest are simply shortcuts
  - Notice: Root Ring is sorted by name ID



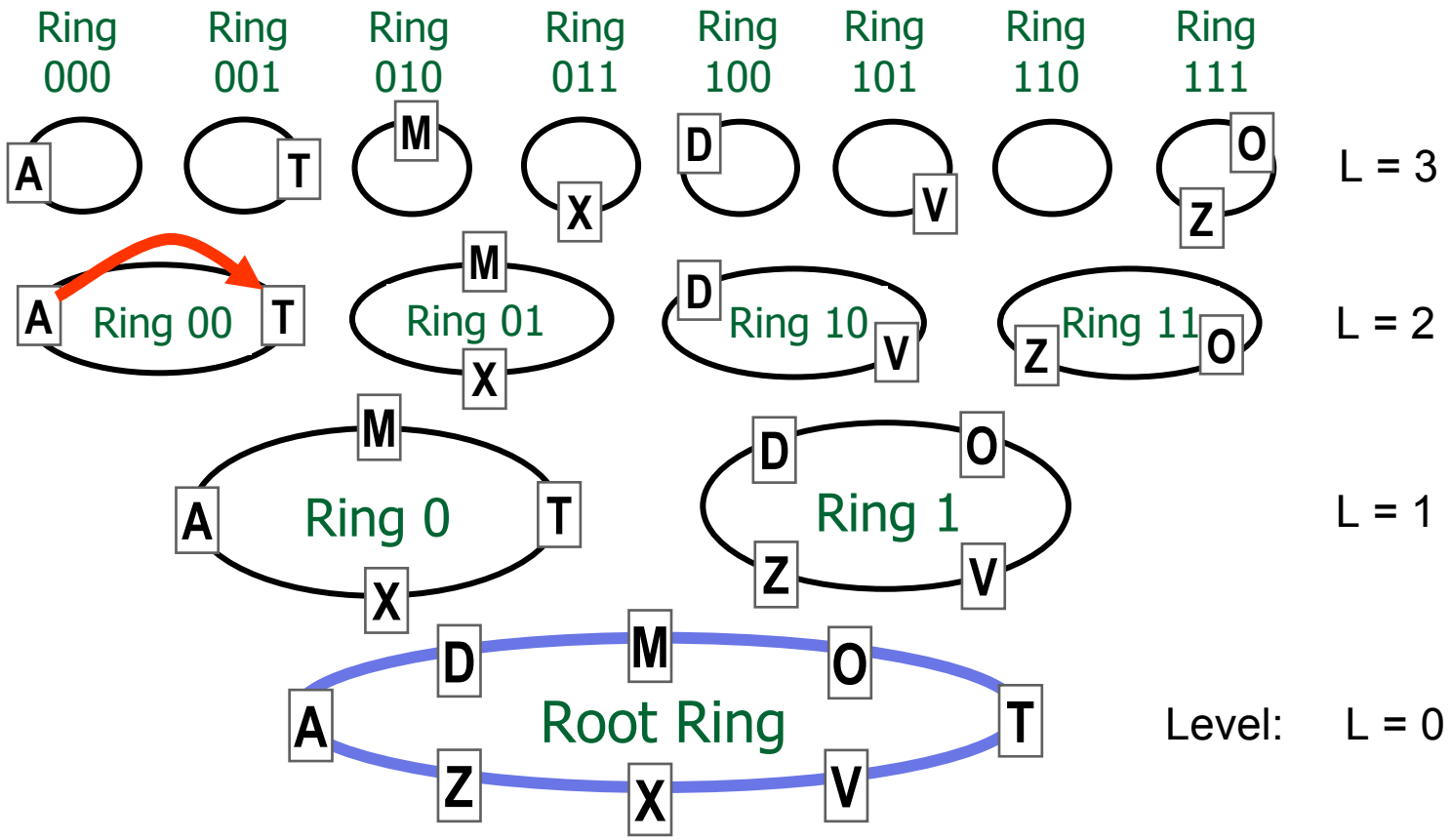
Level: L = 0

Routing by Name ID



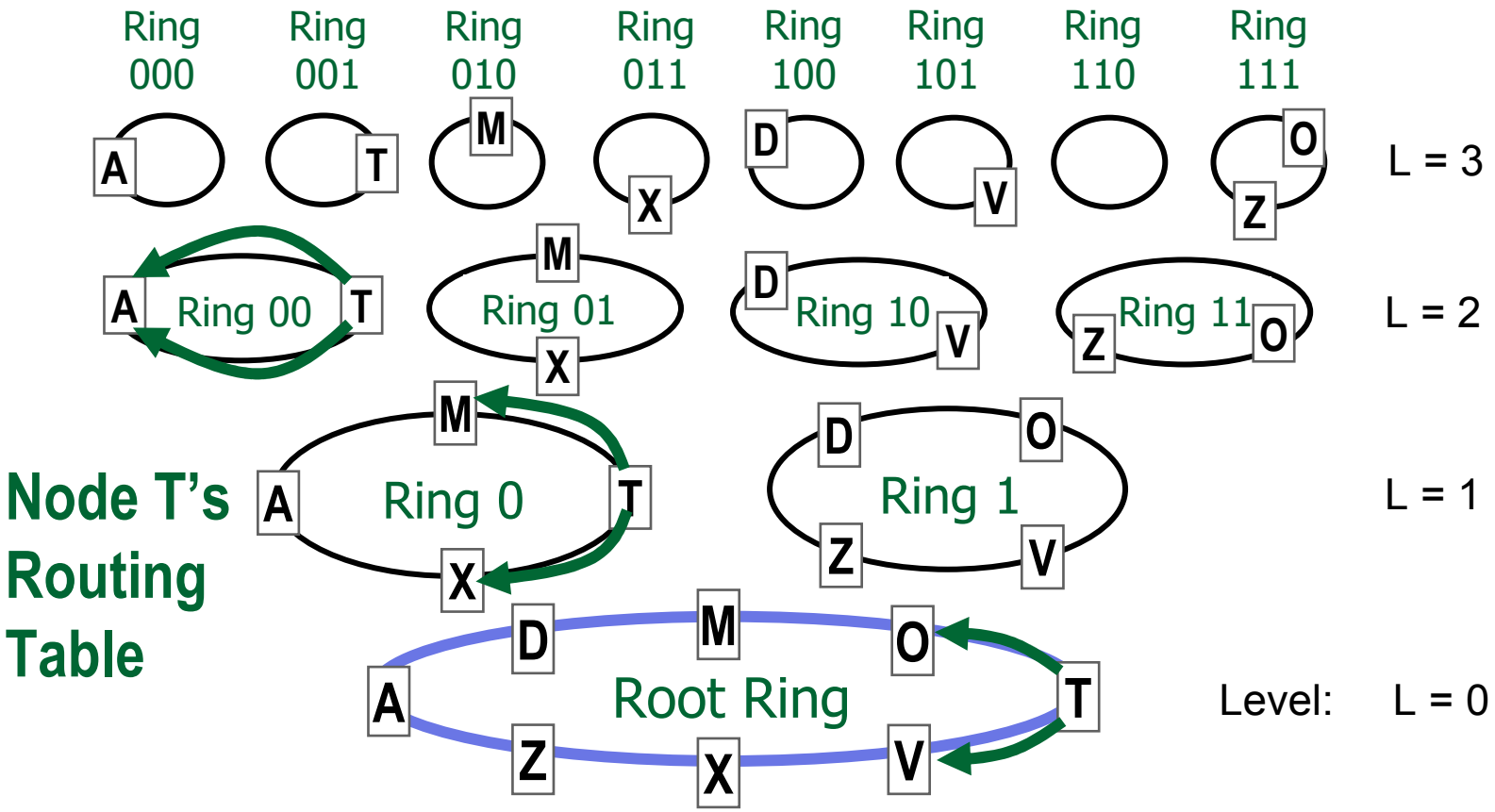
- Example: route from A to V
- Simple Rule: Forward the message to node that is closest to dest, without going too far.

Routing by Name ID



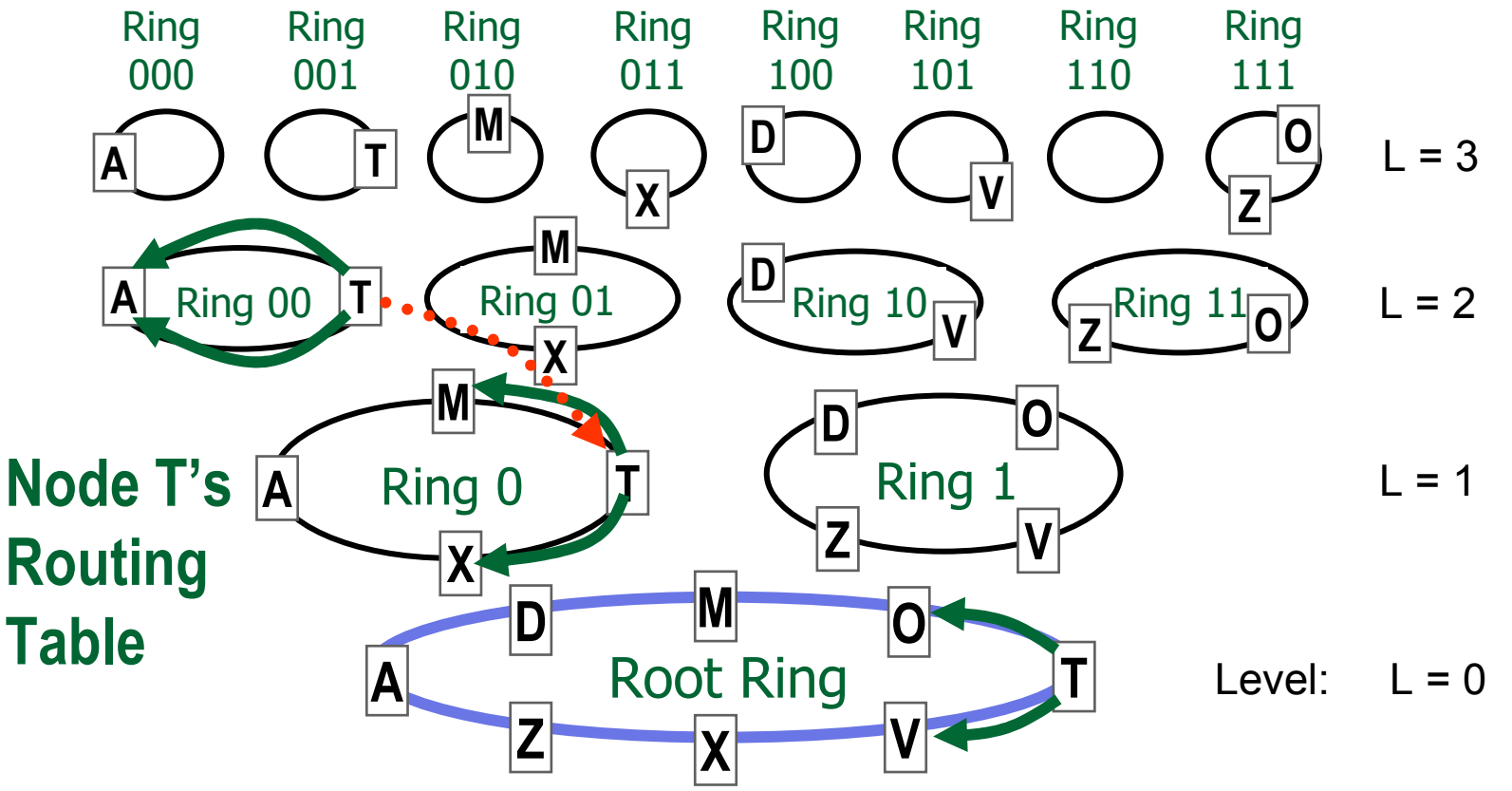
- Example: route from A to V
- Simple Rule: Forward the message to node that is closest to dest, without going too far.

Routing by Name ID



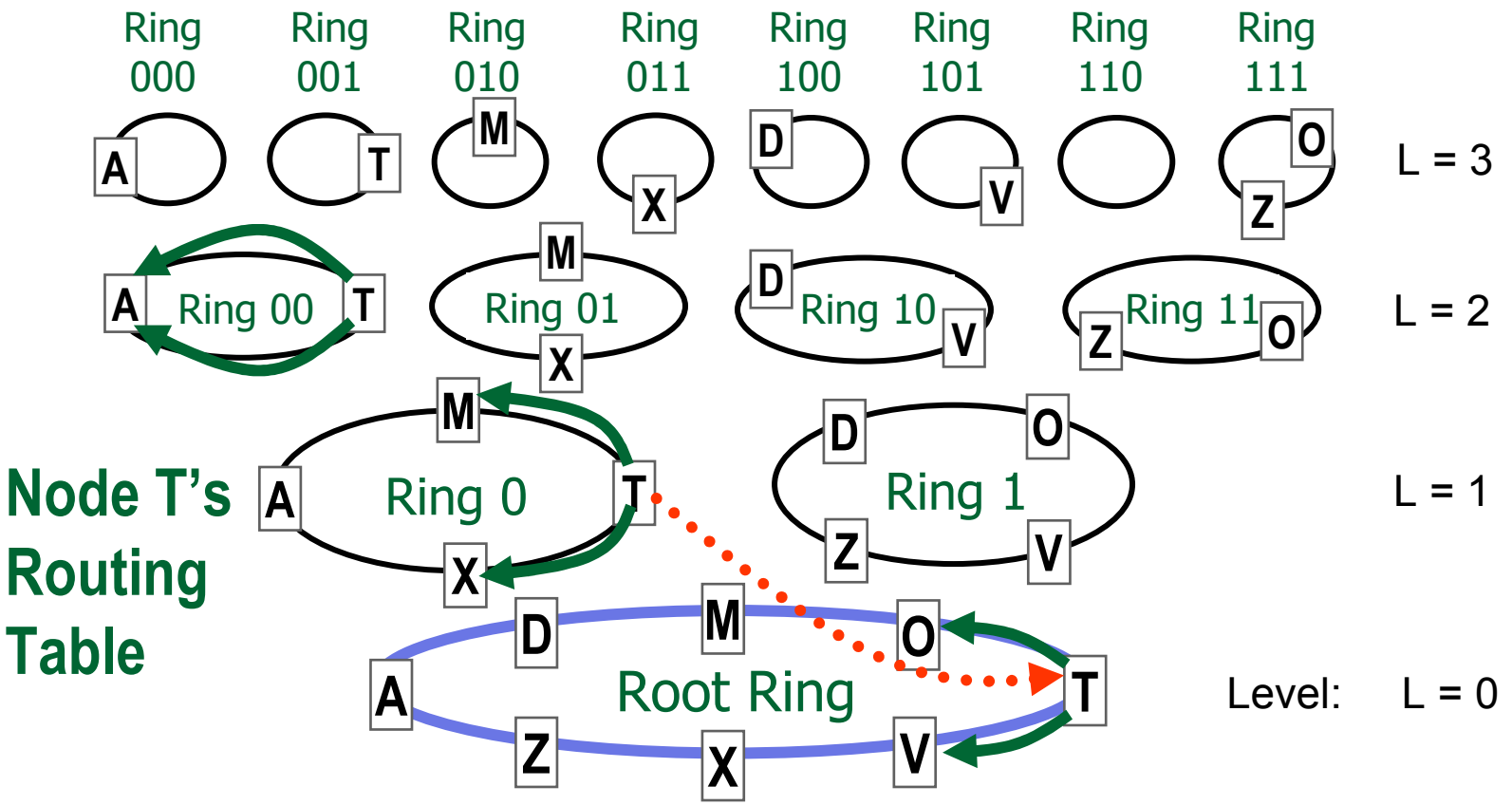
- Example: route from A to V
- Simple Rule: Forward the message to node that is closest to dest, without going too far.

Routing by Name ID



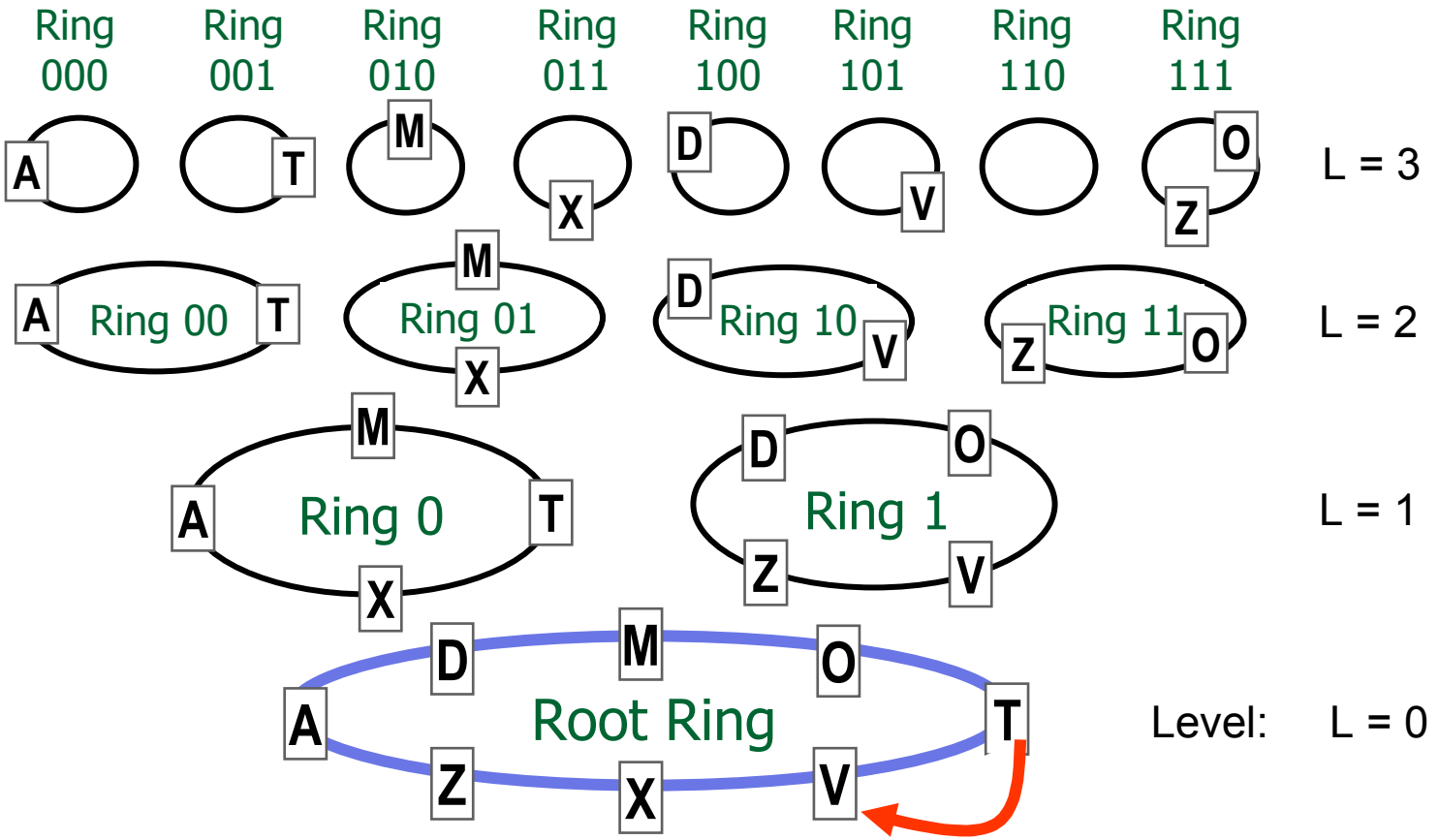
- Example: route from A to V
- Simple Rule: Forward the message to node that is closest to dest, without going too far.

Routing by Name ID



- Example: route from A to V
- Simple Rule: Forward the message to node that is closest to dest, without going too far.

Routing by Name ID

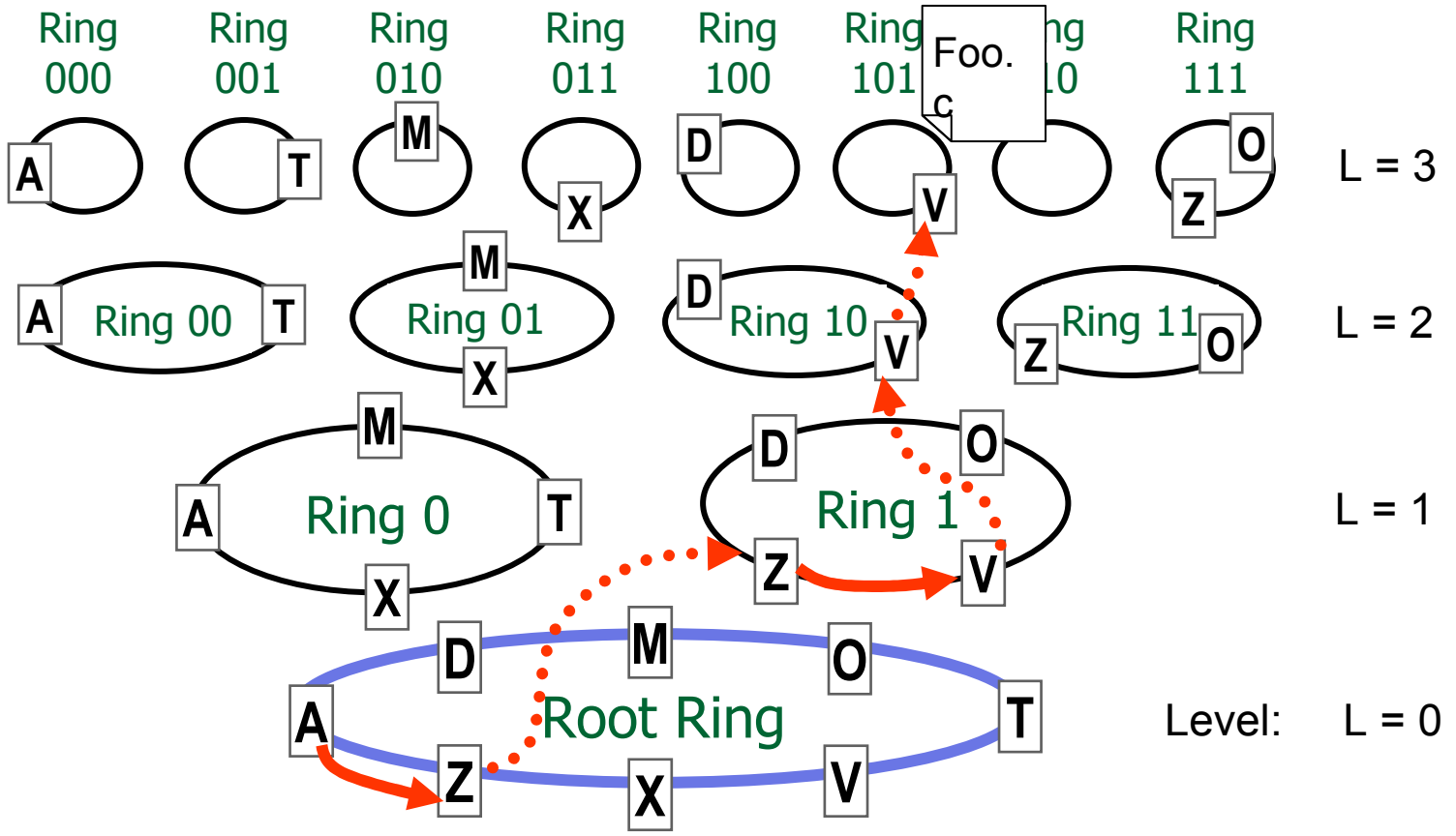


- Example: route from A to V
- Simple Rule: Forward the message to node that is closest to dest, without going too far.

Routing by Numeric ID

- Provides the basic DHT primitive
- To store file “Foo.c”
  - Hash(“Foo.c”) → a random numeric ID
  - Find highest ring matching that numeric ID
  - Store file on node in that ring
- O(logN) routing efficiency

Routing by Numeric ID



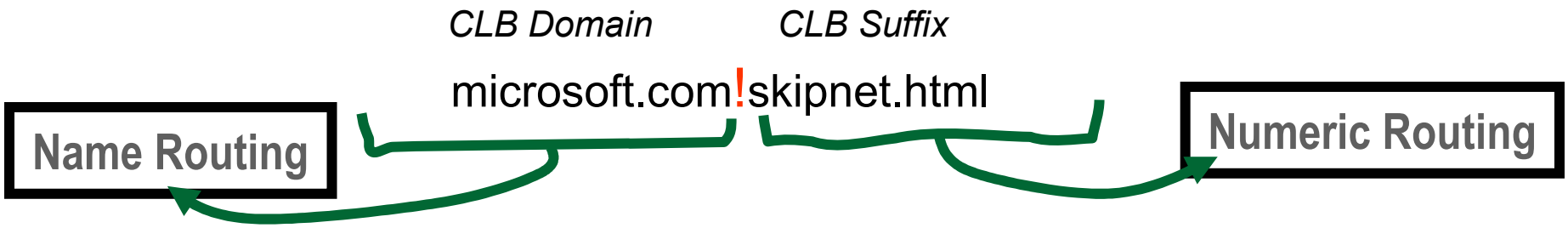
- Store file “Foo.c” from node A
  - Hash(“Foo.c”) = 101…
- Route from A to V in *numeric* space

Outline

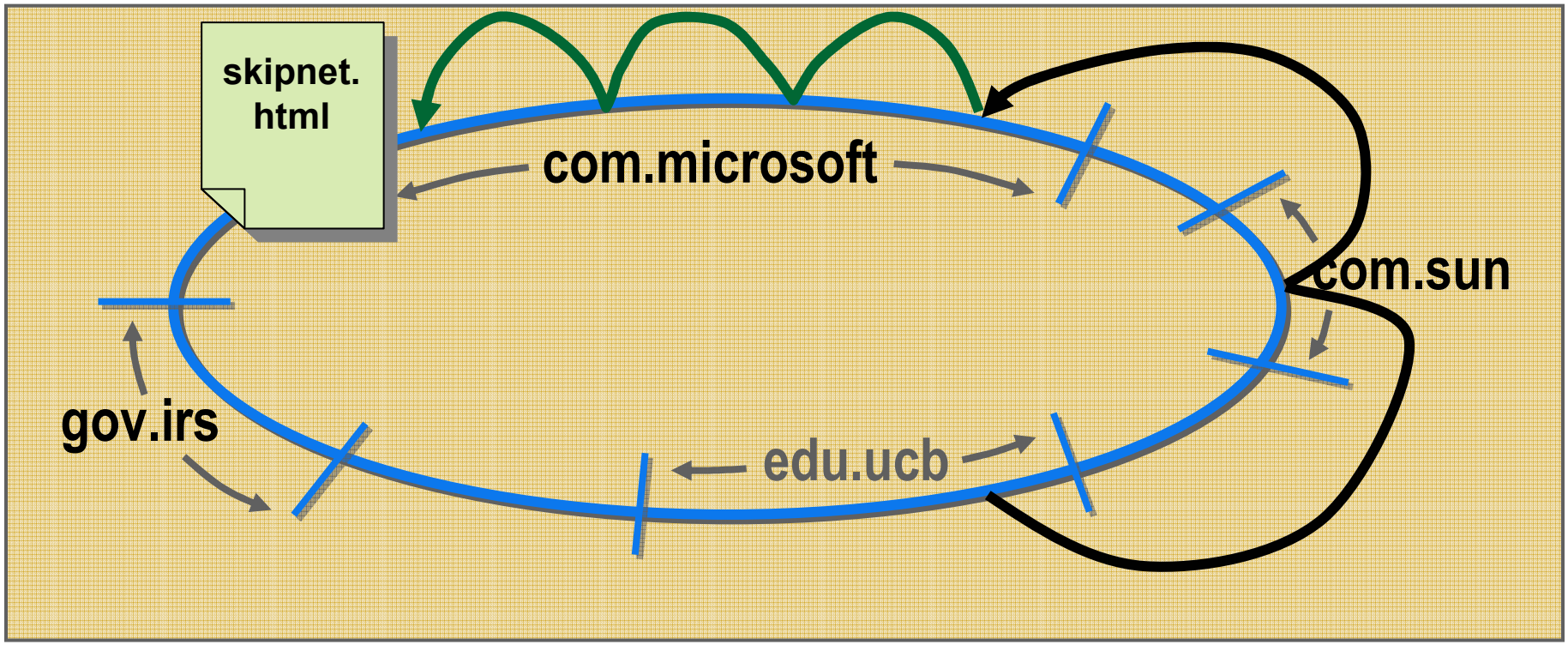
- Contributions
- How it Works
- **LOCALITY PROPERTIES**
- Performance
- Beyond SkipNet

Constrained Load Balancing (CLB)

- A result of the ability to route in both address spaces
- Divide data object names into 2 parts using the '!' special character

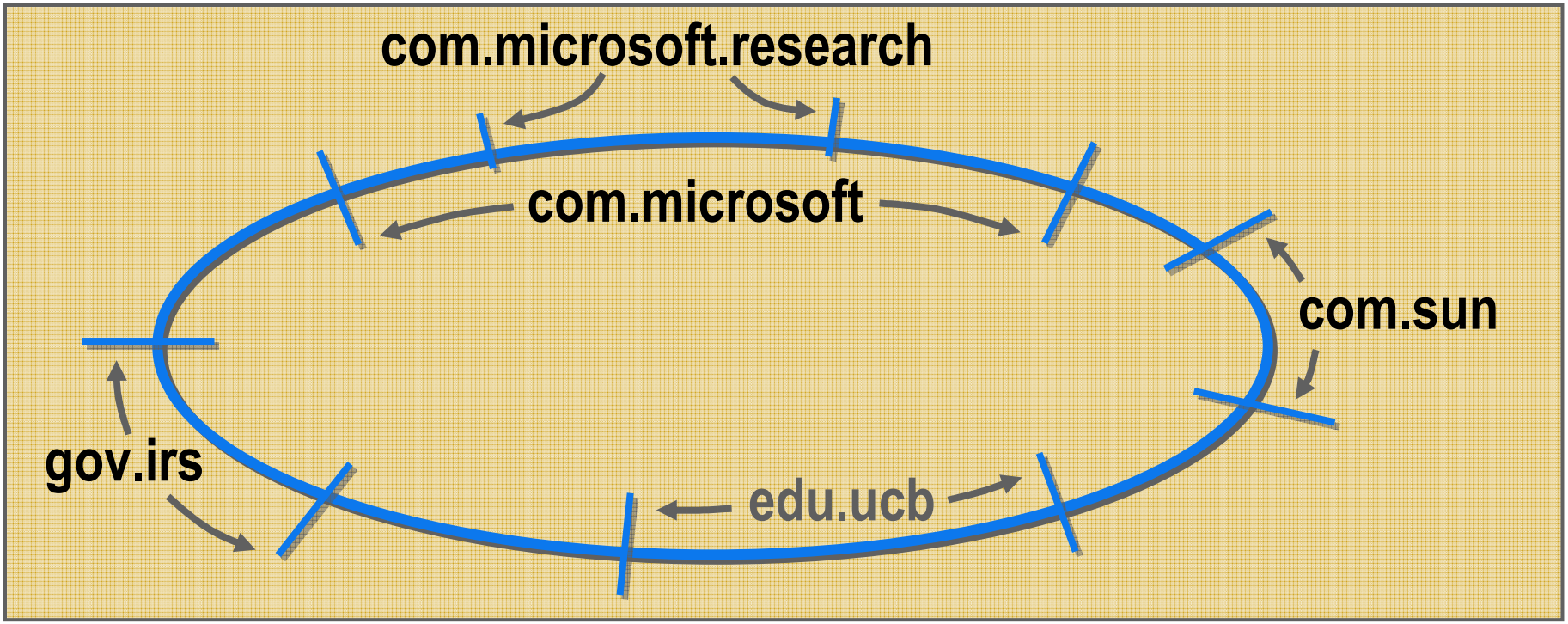


CLB Example



- To read file “com.microsoft!skipnet.html”
  - Route by name ID to “com.microsoft”
  - Route by numeric ID to Hash(“skipnet.html”) within the “com.microsoft” constraint

Path Locality

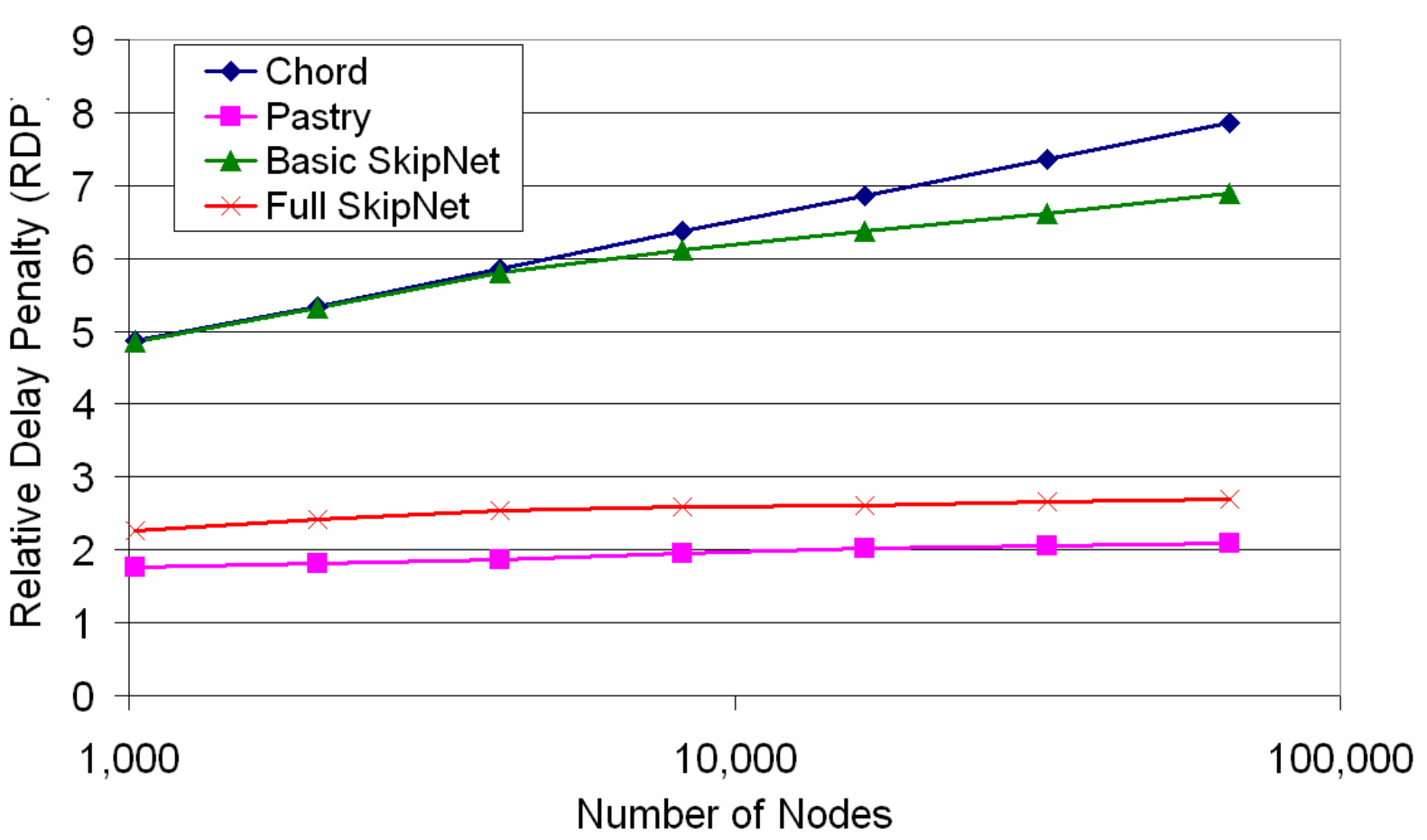


- Organizations correspond to contiguous SkipNet segments
  - Internal routing by NameID remains internal
- Nodes have left / right pointers

Outline

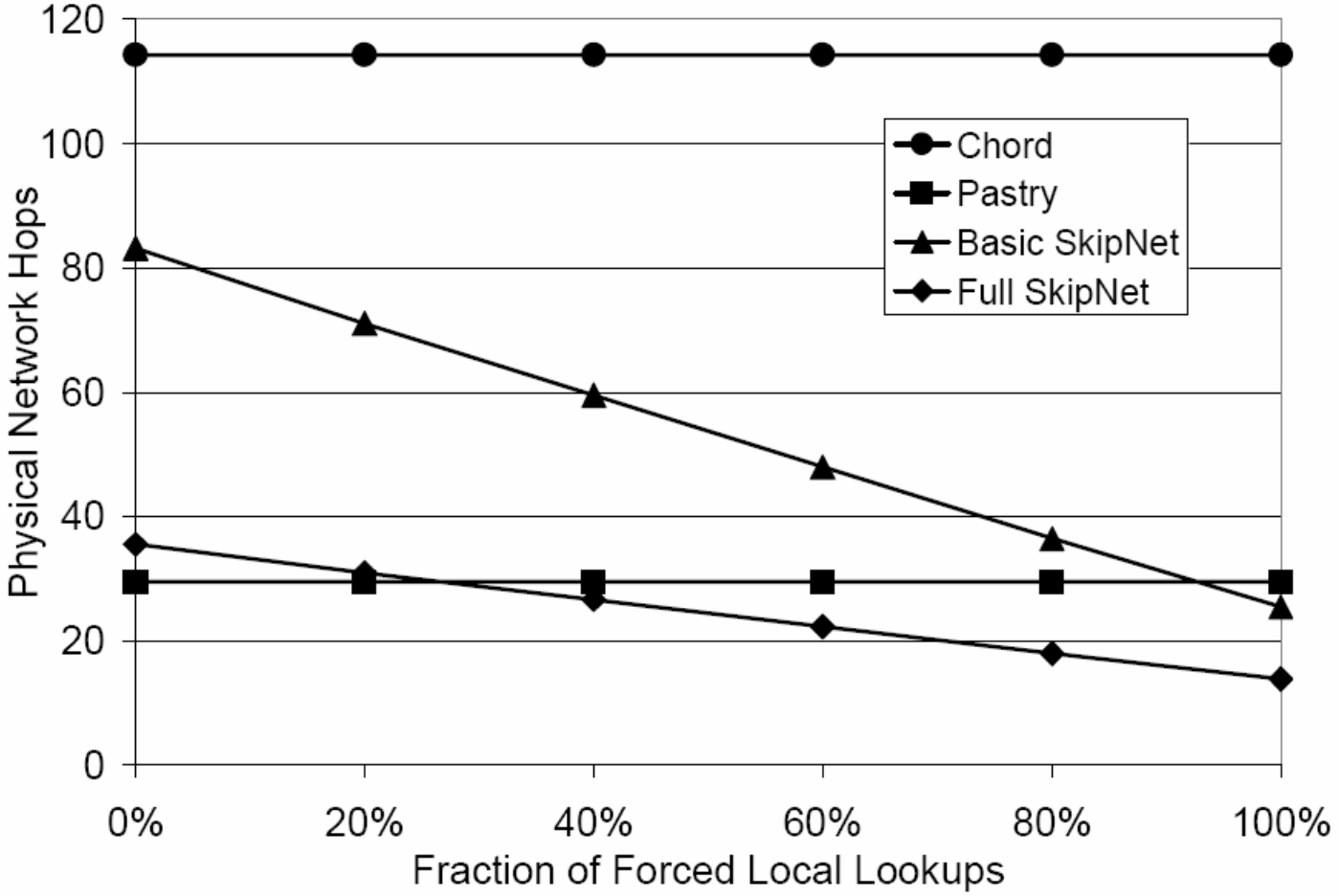
- Contributions
- How it Works
- Locality Properties
- **PERFORMANCE**
- Beyond SkipNet

Routing by Name ID Performance



Benefits come at no extra cost

Local Lookup Performance



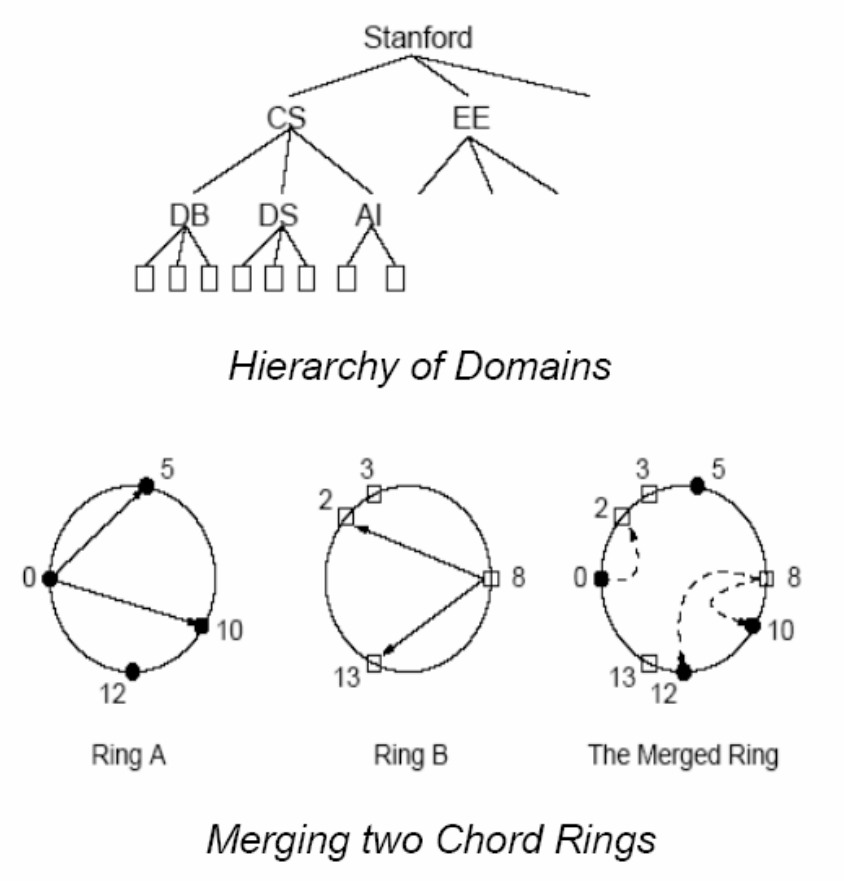
- Full SkipNet outperforms others

Outline

- Contributions
- How it Works
- Locality Properties
- Performance
- **BEYOND SKIPNET**

Alternatives to SkipNet

- Hierarchical DHTs (Canon) -- 2004
  - E.g. Hierarchical Chord (Crescendo)
  - Many smaller rings merged together



Alternatives to SkipNet

- Hierarchical DHTs (Canon)
  - Adapts to Physical Network
  - Efficient Caching
  - Efficient Multicast
  - Exhibits same content locality properties as SkipNet
    - Content and path locality
    - Local administrative domains
    - Fault Isolation
- SkipNet requires modifying the key to ensure locality
  - Canon allows arbitrary storage domains w/o key modification

Chord vs SkipNet: What's the difference?

- Very similar routing structure: ring with shortcuts
- SkipNet data ordering and placement is based on user-chosen name IDs, Chord is random
  - SkipNet numeric IDs are random/arbitrary, Chord is tied to hash
- SkipNet uses bidirectional pointers

Context of SkipNet: Big Questions

- How does this fit in?
- Are the contributions significant?
  - Addresses Chord's two big problems
    - Awareness of underlying topology
    - Content placed close to the users
- Is this a good approach?
  - P-Table and C-Table approaches to speedup routing seem hacky and unclear
- Comments

References

- P. Ganesan, K. Gummadi, and H. Garcia -Molina, “Canon in G major: designing DHTs with hierarchical structure”, Proc. of the 24th IEEE International Conference on Distributed Computing Systems (IEEE ICDCS’04), 2004.
- http://theory.csail.mit.edu/~nickh/Publications/SkipNet/usits.ppt