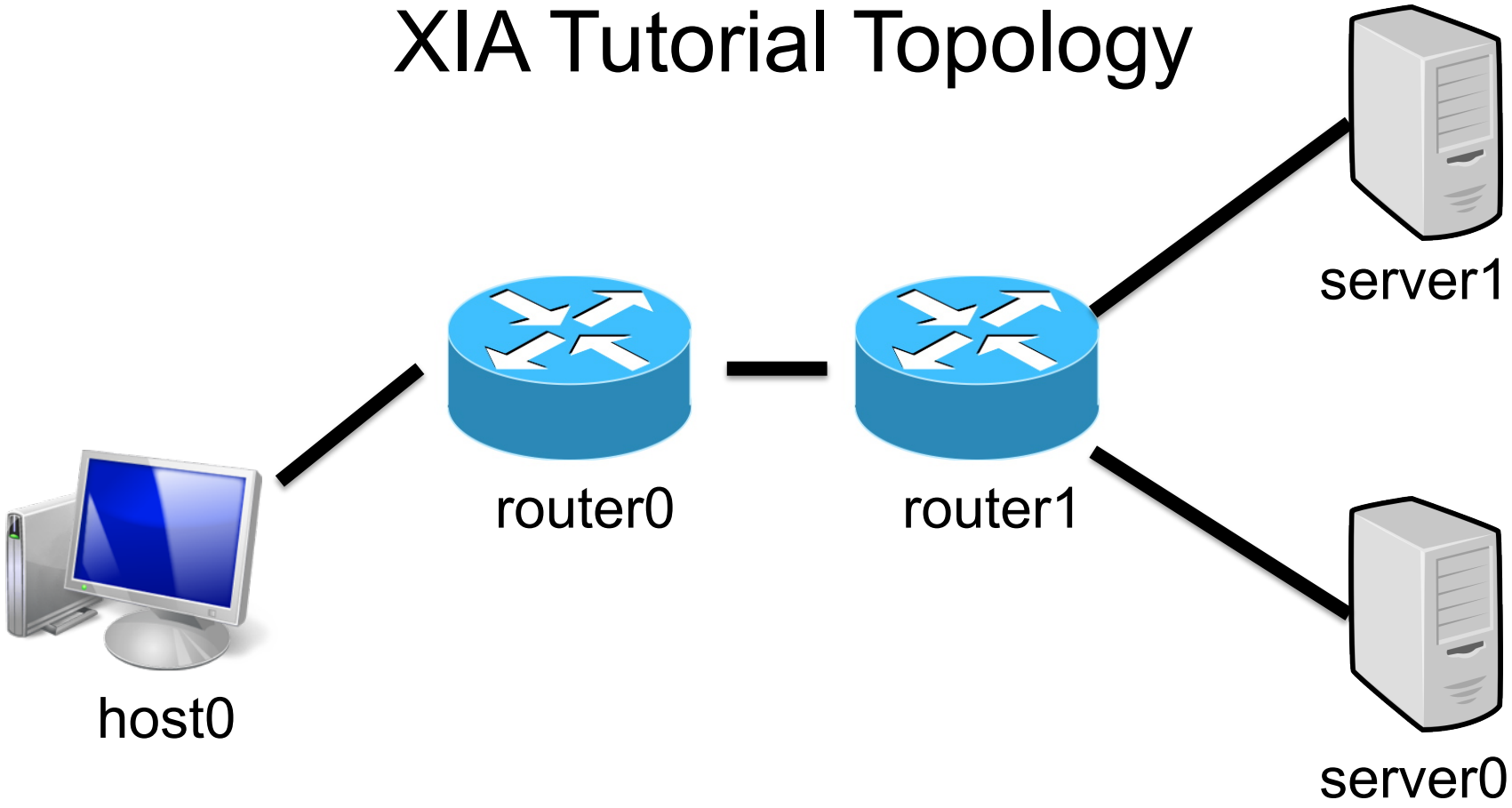
Carnegie Mellon University

eXpressive Internet Architecture on GENI

Srinivasan Seshan, Matt Mukerjee, Yuchen Wu, Dan Barrett, …

XIA Tutorial Topology



The eXpressive Internet Architecture (XIA) Team

Peter Steenkiste, Dave Andersen, David Eckhardt, Sara Kiesler, Jon Peha, Adrian Perrig, Srini Seshan, Marvin Sirbu, Hui Zhang

Aditya Akella

John Byers

Bruce Maggs

A History of Internet Evolution

- Many success stories

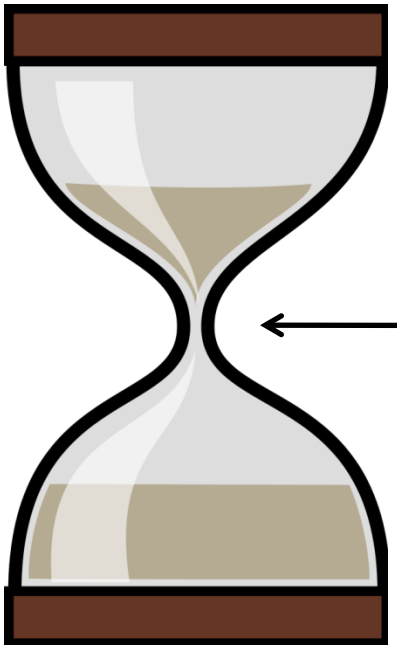
- 1983 → Flag day switch from NCP to IP
- 1988 → /etc/hosts to DNS
- 1996 → TCP SACK
- 1989-1994 → EGP to BGP [1..4]

A History of Internet Evolution

- But also many failures

- 1986 → IP Multicast
- 1997 → IntServ
- 1998 → DiffServ
- 1995 → IPv6
- Internet capabilities, traceback schemes, explicit congestion control, content-oriented processing, …

A History of Internet Evolution – A Summary



Applications

IP

Technology

Innovation both above and below IP

- **Hard to change IP**
  - …especially after 1990

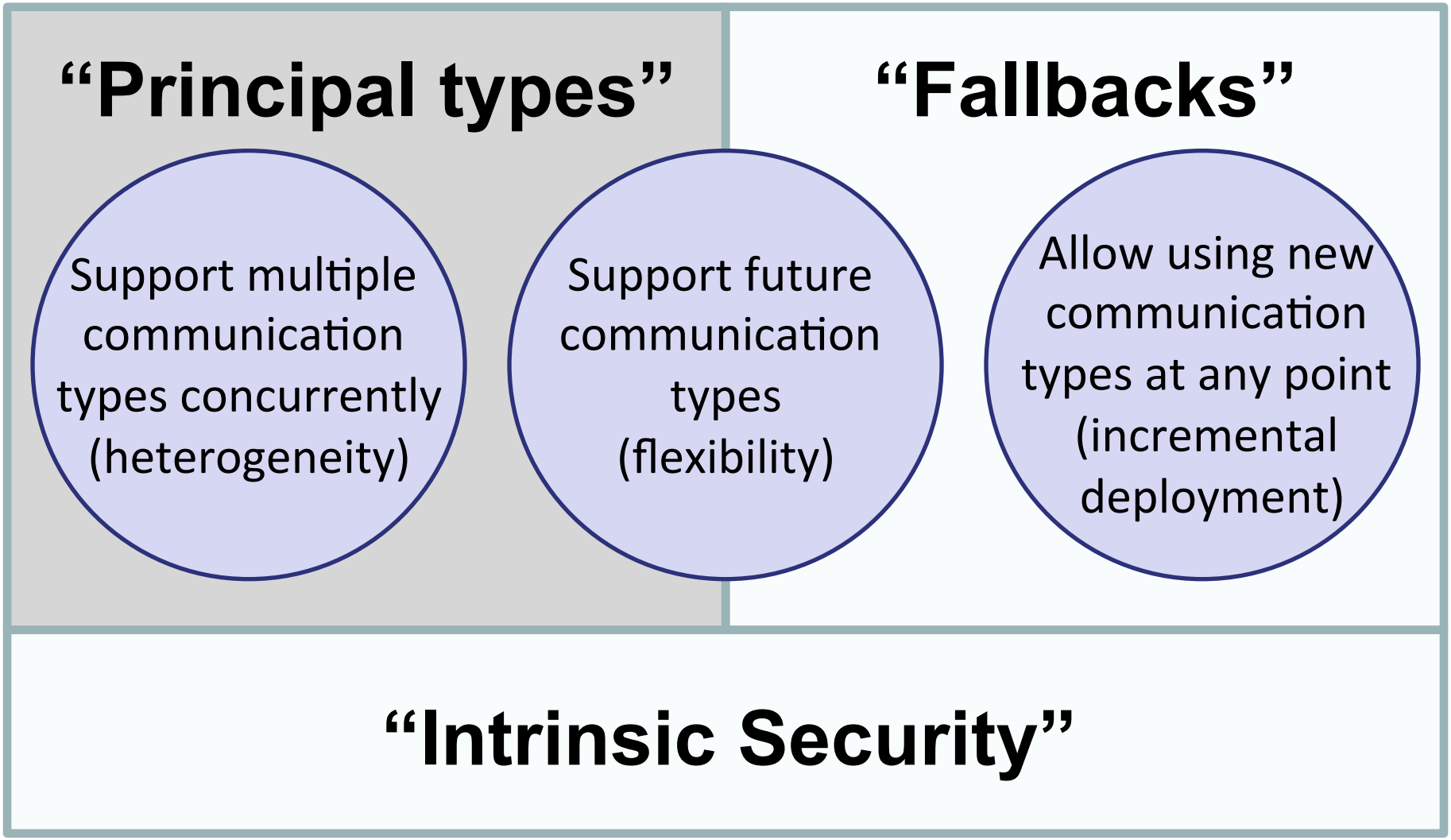
A Perfect Match…or Redundant?

- GENI goals: “Overcoming the Internet Impasse Through Virtualization” [Hotnets 2004]
  - Provide a virtual laboratory for networking and distributed systems research and education.
  - Explore new networking techniques at scale
- XIA goals: “An Architecture for an Evolvable and Trustworthy Internet” [Hotnets 2011]
  - Design a network architecture that simplifies the use and introduction of new network functionality

Outline

- XIA overview
  - Architecture review
- Using XIA as a research platform
  - Adding new functionality to XIA

XIA's Goals and Design Pillars



Principal Types

Define your own communication model

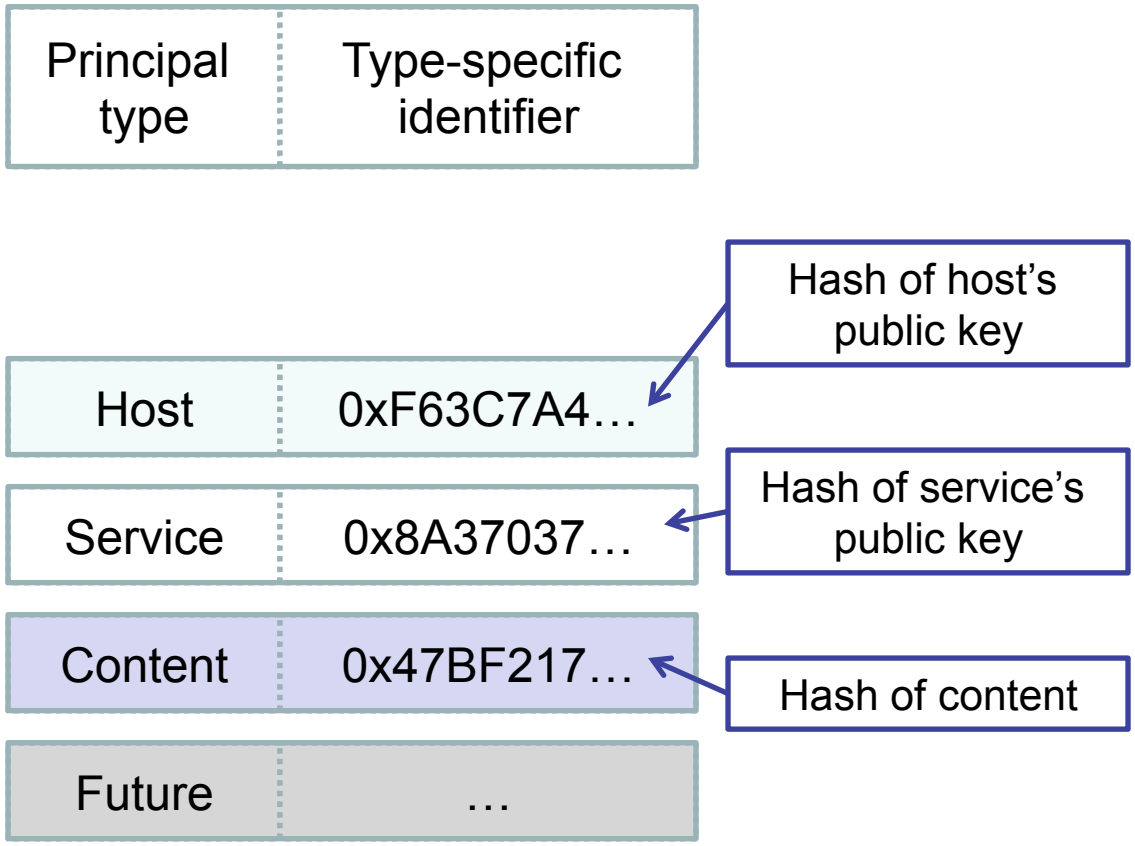
Principals

Current Internet

IP address

128.2.10.162

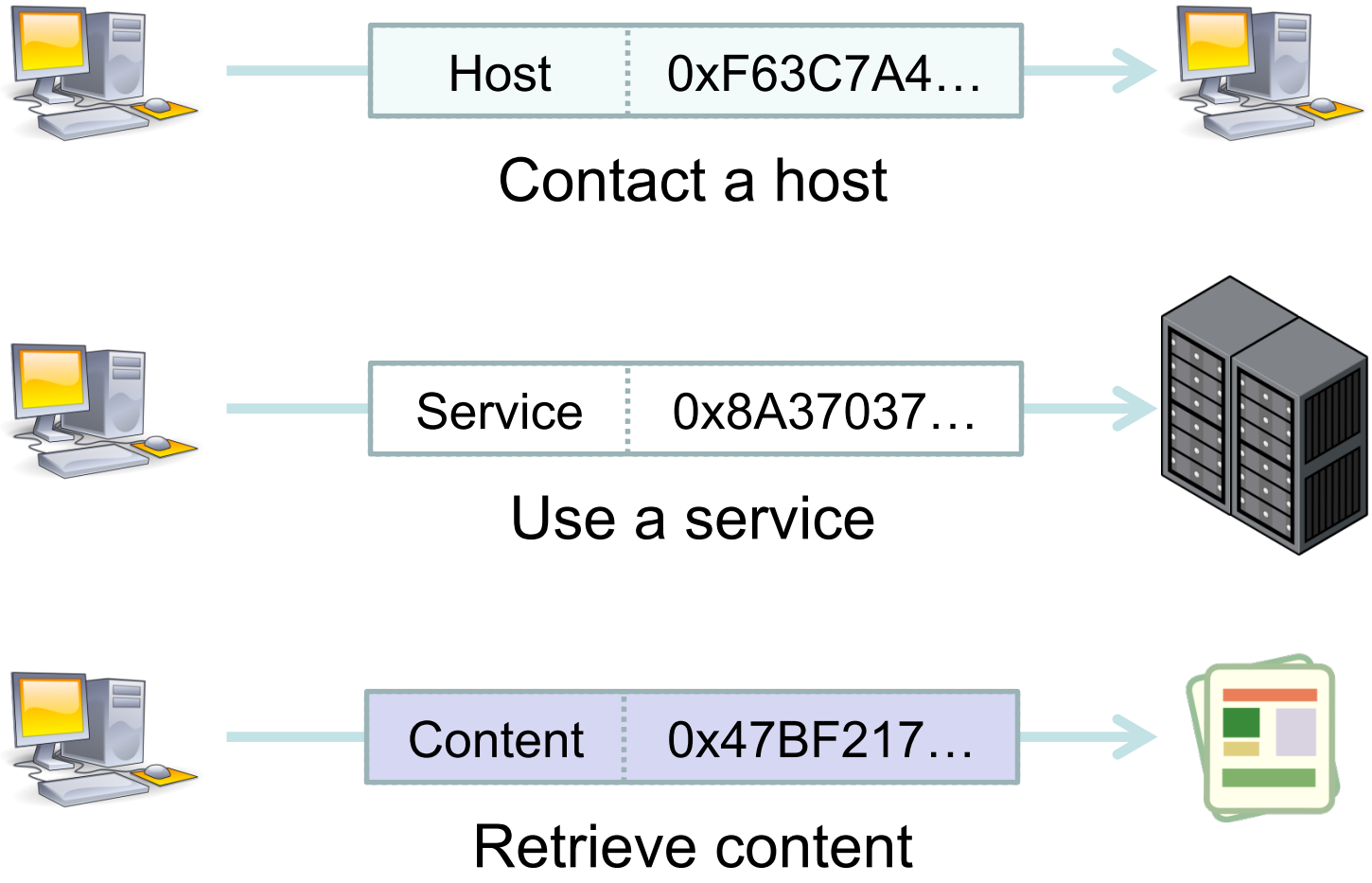
XIA



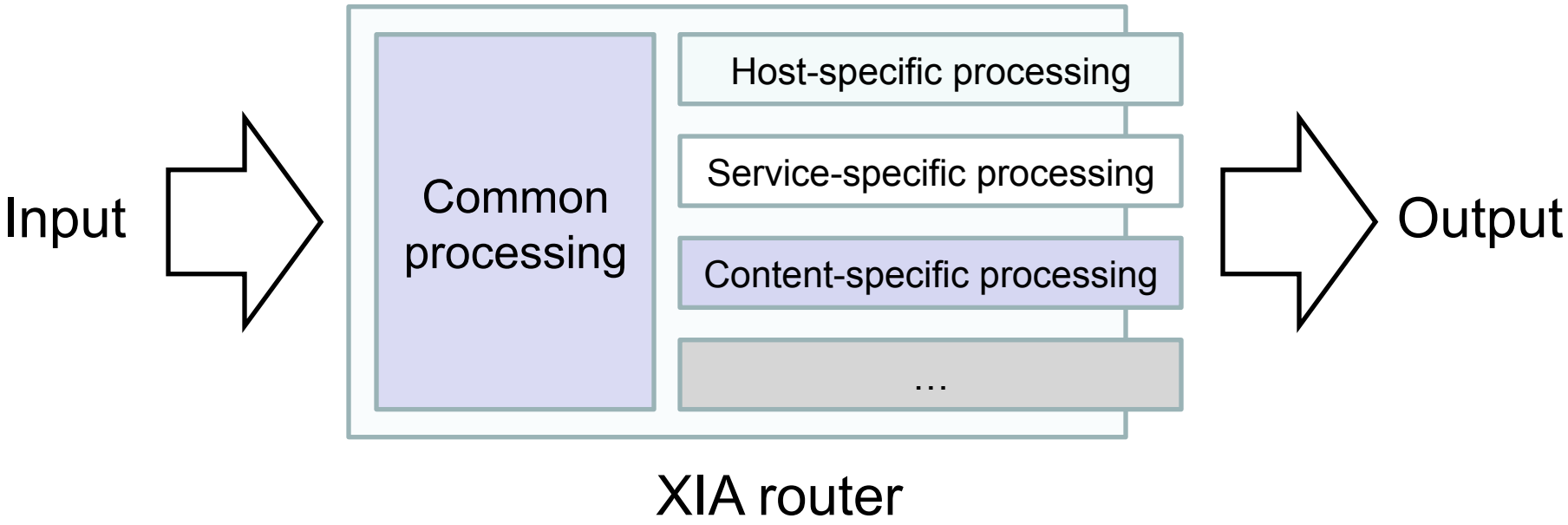
Principal Definition 1: Address Allocation and Intrinsic Security

- XIA uses self-certifying identifiers that guarantee security properties for communication operation
  - Host ID is a hash of its public key – accountability (AIP)
  - Content ID is a hash of the content – correctness
  - Does not rely on external configuration
- Intrinsic security is specific to the principal type
- Example: retrieve content using …
  - Content XID: content is correct
  - Service XID: the right service provided content
  - Host XID: content was delivered from right host

Principal Definition 2: Type-Specific Semantics



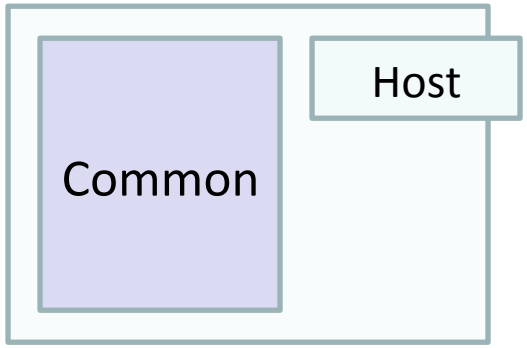
Principal Definition 3: Type-Specific Processing



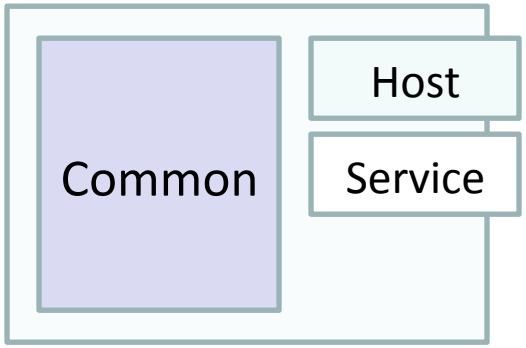
- Type-specific processing examples
  - Service: load balancing or service migration
  - Content: content caching

Routers with Different Capabilities

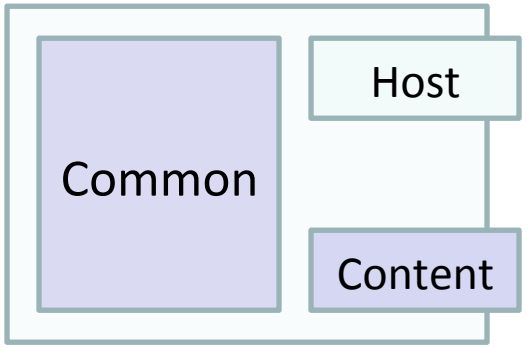
- Routers are **not** required to support every principal type
  - The only requirement: Host-based communication



Host-only router



Service-enabled router



Content-enabled router

Flexible Addressing

Tomorrow's communication types… today!

Fallbacks: Alternative Ways for Routers to Fulfill Intent of Packet

Intent: Retrieve Content

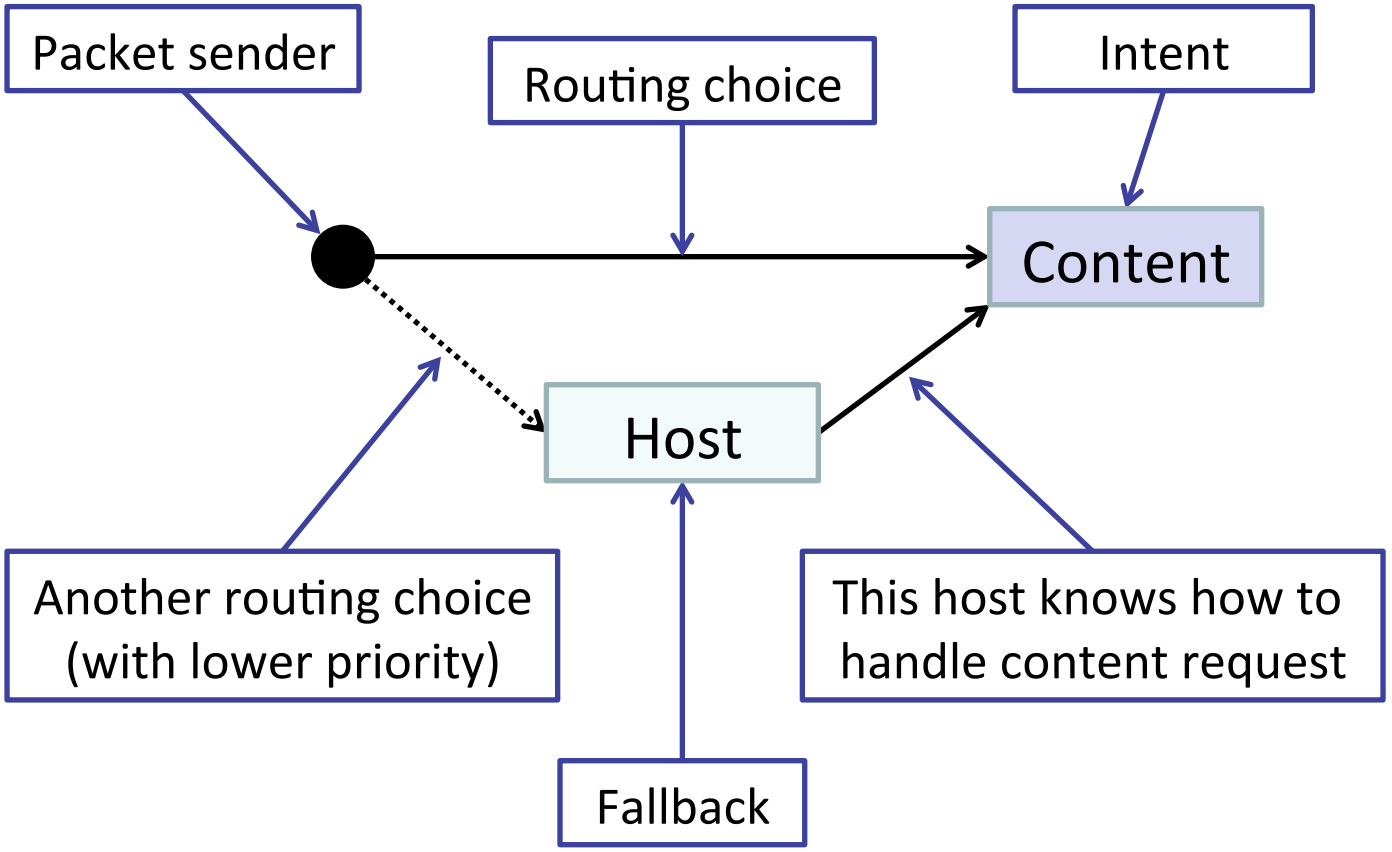
Fallback: Contact Host ,

who understands Content request

What the network does:

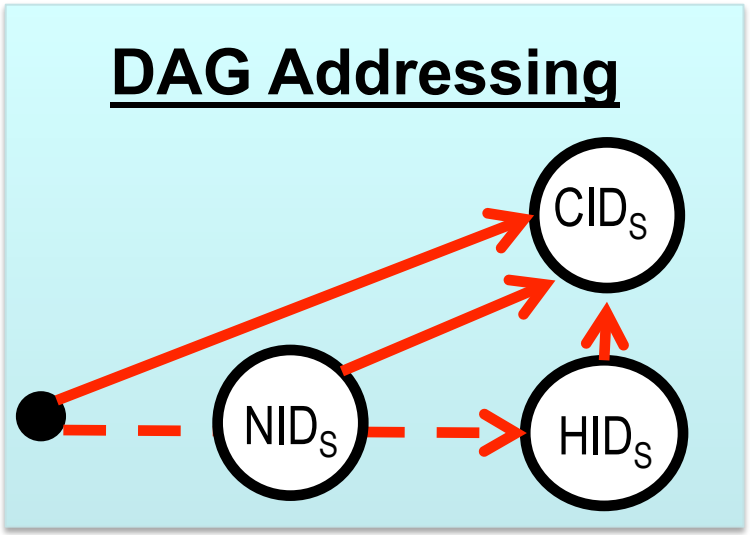
- With content-enabled routers, use Content for routing
- Otherwise, use Host for routing (always succeeds)

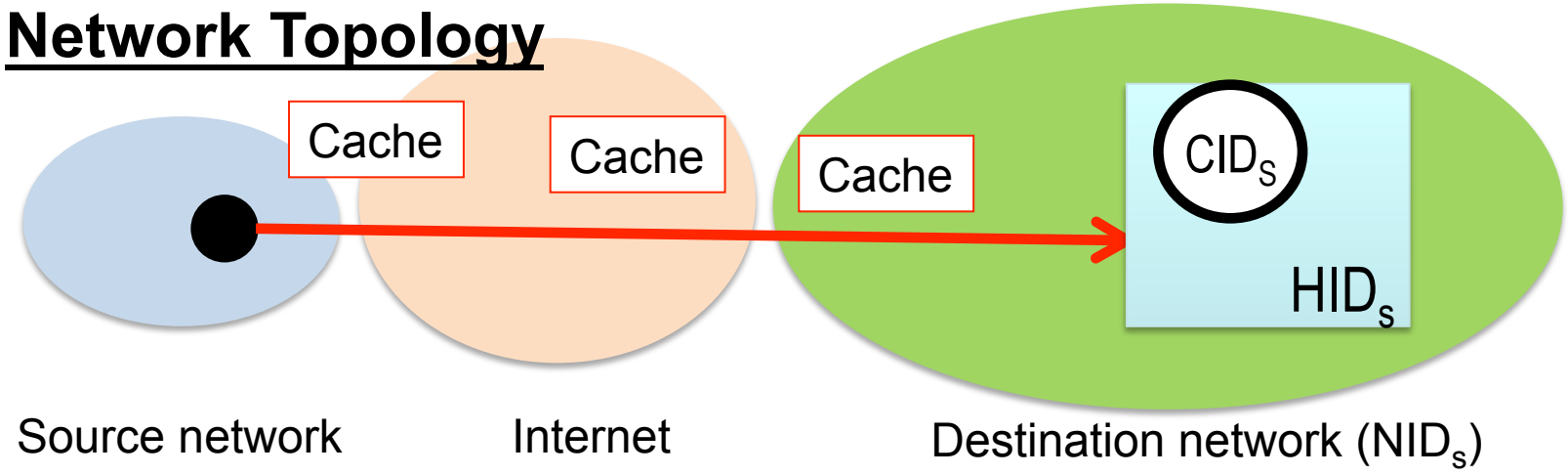
DAG (Direct Acyclic Graph)-Based Addressing Enables Fallbacks



Scoping Using DAGs

- Combining intent and fallback address offers flexibility for network in completing request
  - Set of principal types can evolve
  - Also supports scoping
  - Implemented as DAGs





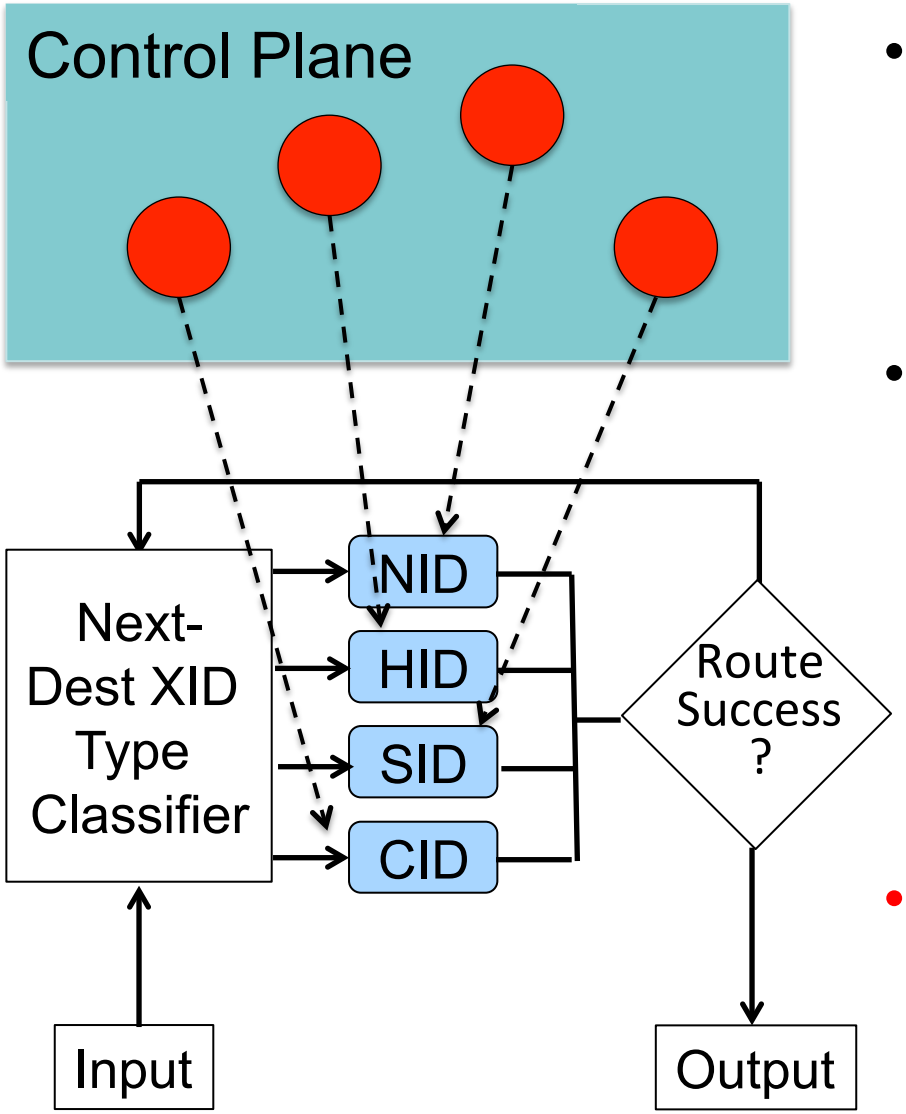
Outline

- XIA overview
  - Architecture review
- Using XIA as a research platform
  - Adding new functionality to XIA

GENI and XIA: Complementary

- GENI → a great testbed for deployment of new ideas
- XIA → a great framework for designing/ implementing new ideas
- Adding new XIDs:
  - Data plane: define per hop processing
  - Control plane:
  - API/intrinsic security

XIA Packet Processing Pipeline



- Principal-independent processing defines how to interpret the DAG
  - Core architecture
- Principal-dependent processing realizes forwarding semantics for each XID type
  - Logically: one forwarding table per XID type
  - Reality: anything goes, e.g., no forwarding table
- **Control plane sets up forwarding for each principal type**

XIA API

- XSockets Library
  - Designed to be similar to Berkley sockets
  - New socket family (`AF_XIA`) and `sockaddr_x` structure
  - New content centric APIs
  - No support for `getXbyY` functions calls (`Xgetaddrinfo`)
- Compatibility Library
  - Catches standard socket calls and remaps them to XIA specific calls
  - Easier to port existing application, or create multi-network applications

Building and Using an XIA Network

- Looking at various network challenges and they can be addressed within XIA
  - Internet congestion control
  - Multicast and mobility
- Deploying and managing XIA networks
  - Multihoming, multipath
  - Service discovery, binding and routing
  - Fast XIA
  - Establishing and controlling session

XIA Resources

- XIA Home Page
  http://www.cs.cmu.edu/~xia
- XIA Wiki
  https://github.com/XIA-Project/xia-core/wiki
- XIA on Github
  https://github.com/XIA-Project/xia-core

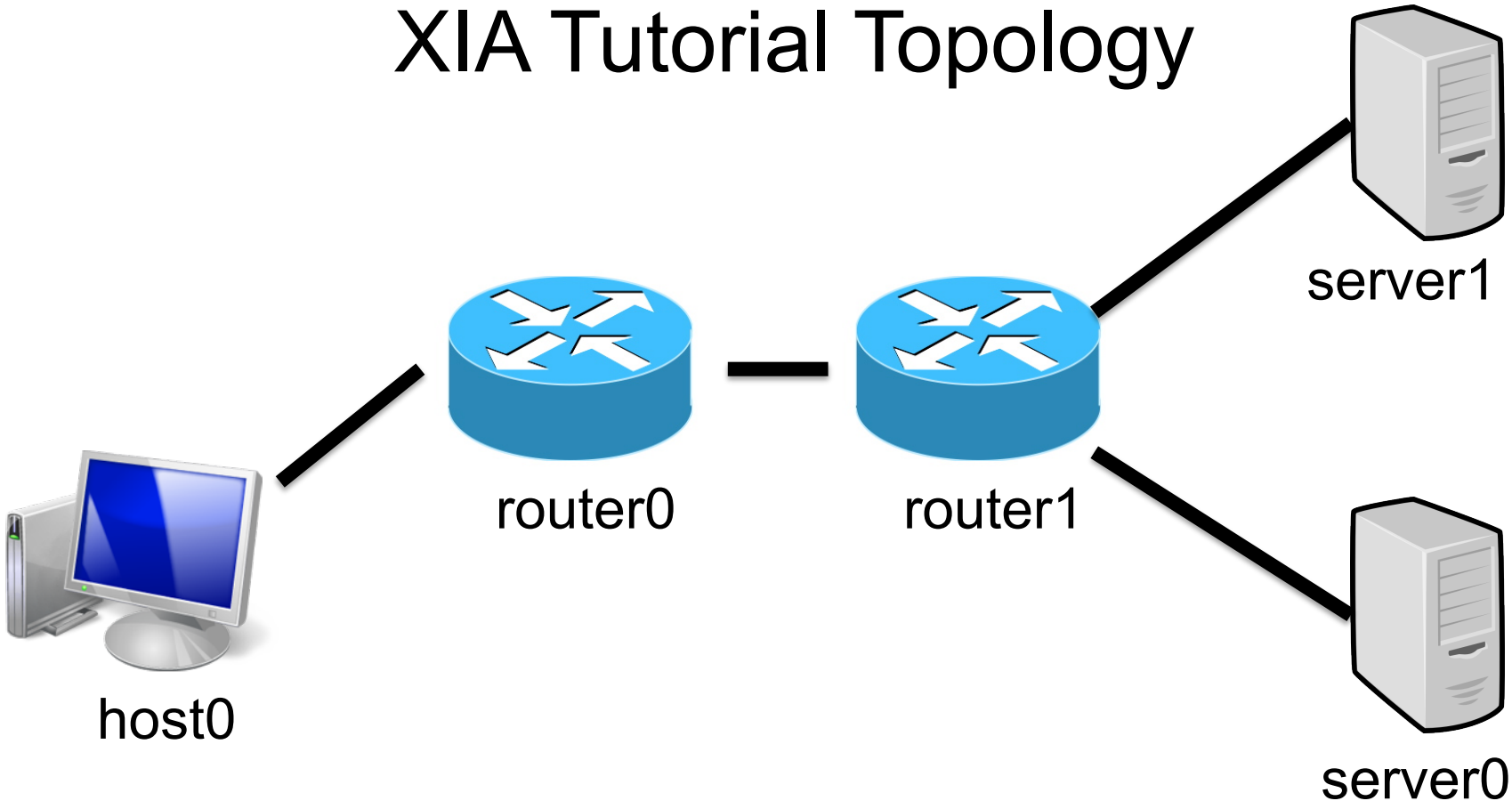
- Email
  - Support
    xia-users-help@cs.cmu.edu
  - XIA-Users Mailing List
    https://mailman.srv.cs.cmu.edu/mailman/listinfo/xia-users
  - XIA Announcements Mailing List
    https://mailman.srv.cs.cmu.edu/mailman/listinfo/xia-announce

What we're going to do today

1. Running XIA over GENI.
   - Do a simple file transfer using content chunks
   - Examine how opportunistic caching works

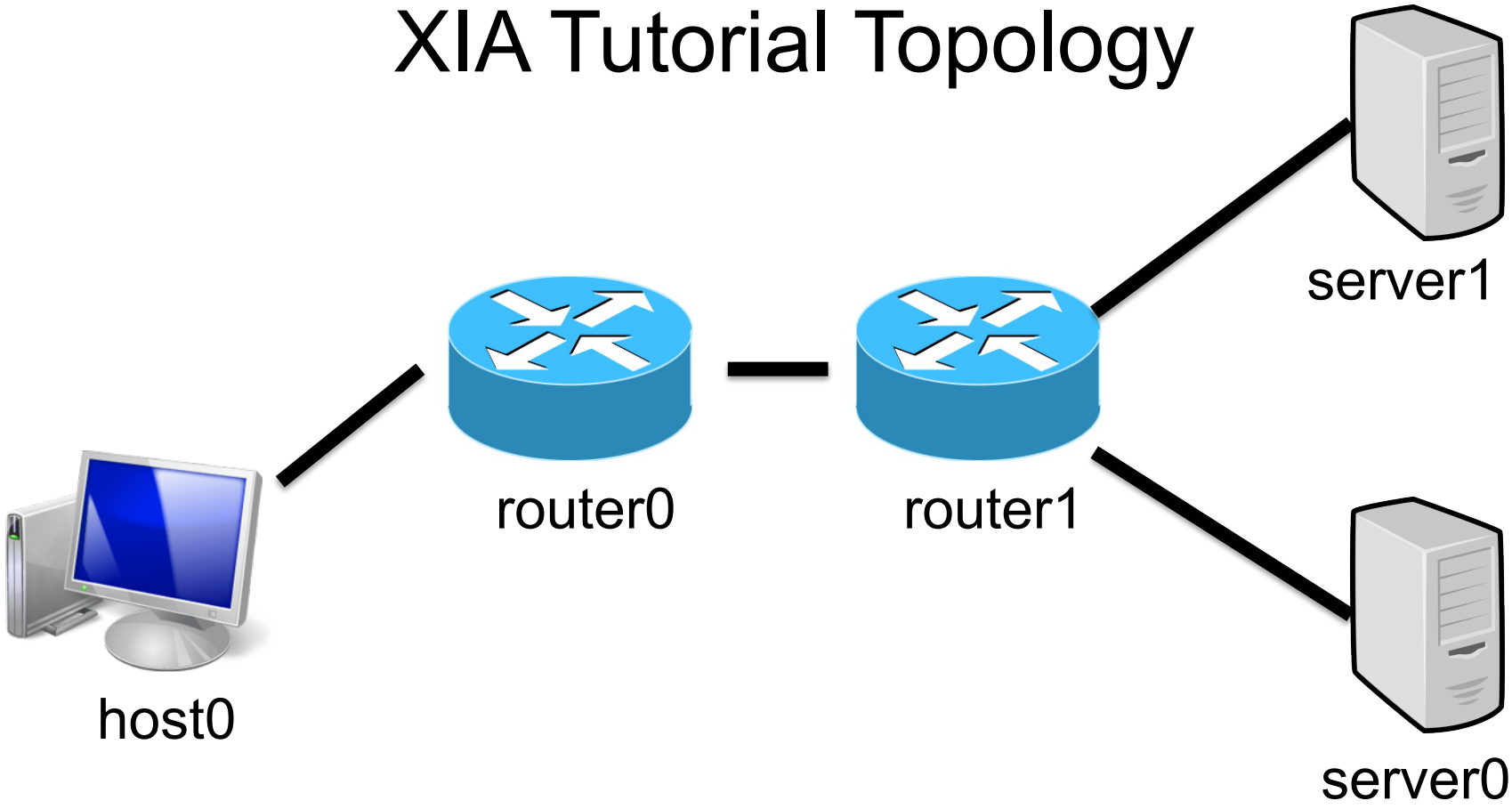
2. Adding a New Principal Type.
   - Add new functionality to XIA
   - New principal type that can do load balancing directly in the network.

XIA Tutorial Topology



Let’s boot the network!

XIA Tutorial Topology



router1's forwarding table

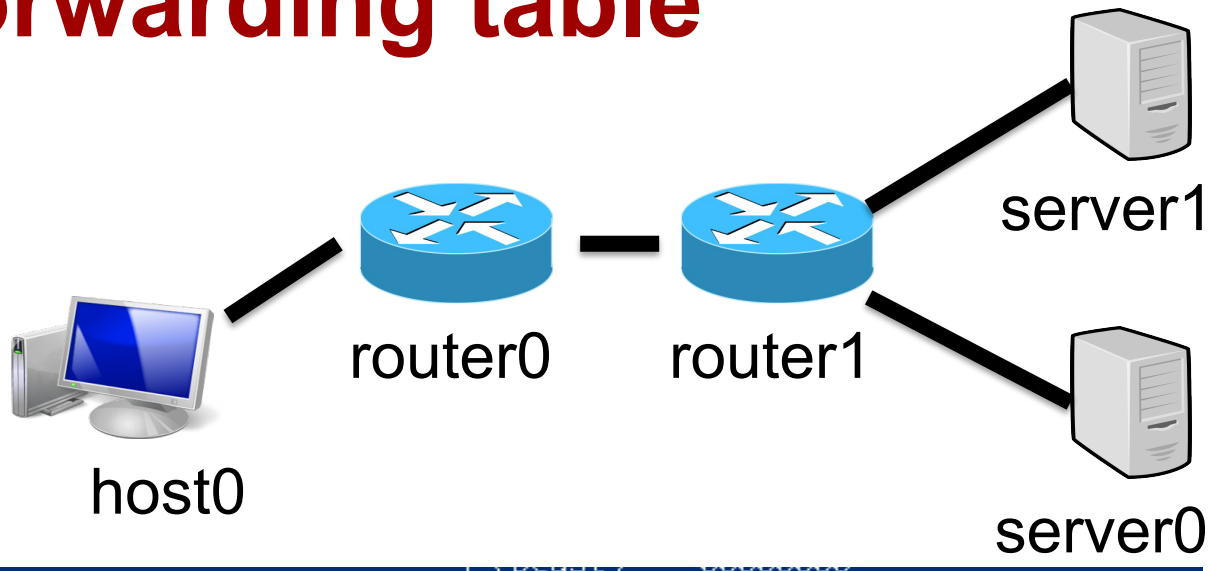
```
XIA Route version 0.9
router1
TYPE  XID                                                  PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                            -7 (fallback) 00000000
AD    AD_NAMESERVER                                         2            0000ffff HID_ROUTER0
AD    AD_SERVERS                                           -2 (self)     00000000
HID   (default)                                            -7 (fallback) 00000000
HID   BHID                                                 -4 (bcast)    00000000
HID   HID_ROUTER1                                          -2 (self)     00000000
HID   HID_ROUTER0                                           2            00000000 HID_ROUTER0
HID   HID_SERVER1                                           1            0000ffff HID_SERVER1
HID   HID_SERVER0                                           0            0000ffff HID_SERVER0
SID   (default)                                            -7 (fallback) 00000000
SID   sid:30000ff00000000000000000000000007b305534        -2 (self)     00000000
SID   SID_XROUTE                                           -2 (self)     00000000
CID   (default)                                            -7 (fallback) 00000000
IP    (default)                                            -7 (fallback) 00000000
```

router1's forwarding table

```
XIA Route version 0.9
router1
TYPE  XID                                                PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                          -7 (fallback) 00000000
AD    AD_NAMESERVER                                       2            0000ffff HID_ROUTER0
AD    AD_SERVERS                                         -2 (self)     00000000
HID   (default)                                          -7 (fallback) 00000000
HID   BHID                                               -4 (bcast)    00000000
HID   HID_ROUTER1                                        -2 (self)     00000000
HID   HID_ROUTER0                                        2             00000000 HID_ROUTER0
HID   HID_SERVER1                                        1             0000ffff HID_SERVER1
HID   HID_SERVER0                                        0             0000ffff HID_SERVER0
SID   (default)                                          -7 (fallback) 00000000
SID   sid:30000ff0000000000000000000000007b305534       -2 (self)     00000000
SID   SID_XROUTE                                         -2 (self)     00000000
CID   (default)                                          -7 (fallback) 00000000
IP    (default)                                          -7 (fallback) 00000000
```

Routes for all connected devices

router1's forwarding table




```
XIA Route version 0.9
router1
TYPE  XID
------------------------------
AD    (default)
AD    AD_NAMESERVER
AD    AD_SERVERS
HID   (default)
HID   BHID
HID   HID_ROUTER1                                  -2 (self)     00000000
HID   HID_ROUTER0                                   2            00000000 HID_ROUTER0
HID   HID_SERVER1                                   1            0000ffff HID_SERVER1
HID   HID_SERVER0                                   0            0000ffff HID_SERVER0
SID   (default)                                                  00000000
SID   sid:30000ff0000000000000000000000007b305534  -2 (self)     00000000
SID   SID_XROUTE                                   -2 (self)     00000000
CID   (default)                                    -7 (fallback) 00000000
IP    (default)                                    -7 (fallback) 00000000
```

Routes for all connected devices

router1's forwarding table

```
XIA Route version 0.9
router1
TYPE  XID                                                PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                          -7 (fallback) 00000000
AD    AD_NAMESERVER                                       2            0000ffff HID_ROUTER0
AD    AD_SERVERS                                         -2 (self)     00000000
HID   (default)                                          -7 (fallback) 00000000
HID   BHID                                               -4 (bcast)    00000000
HID   HID_ROUTER1                                        -2 (self)     00000000
HID   HID_ROUTER0                                         2            00000000 HID_ROUTER0
HID   HID_SERVER1                                         1            0000ffff HID_SERVER1
HID   HID_SERVER0                                         0            0000ffff HID_SERVER0
SID   (default)                                          -7 (fallback) 00000000
SID   sid:30000ff0000000000000000000000007b305534       -2 (self)     00000000
SID   SID_XROUTE                                         -2 (self)     00000000
CID   (default)                                          -7 (fallback) 00000000
IP    (default)                                          -7 (fallback) 00000000
```

Some routes are physical ports

router1's forwarding table

```
XIA Route version 0.9
router1
TYPE  XID                                                PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                          -7 (fallback) 00000000
AD    AD_NAMESERVER                                       2            0000ffff HID_ROUTER0
AD    AD_SERVERS                                         -2 (self)     00000000
HID   (default)                                          -7 (fallback) 00000000
HID   BHID                                               -4 (bcast)    00000000
HID   HID_ROUTER1                                        -2 (self)     00000000
HID   HID_ROUTER0                                         2            00000000 HID_ROUTER0
HID   HID_SERVER1                                                      0000ffff HID_SERVER1
HID   HID_SERVER0                                                      0000ffff HID_SERVER0
SID   (default)                                          -7 (fallback) 00000000
SID   sid:30000ff000000000000000000000007b305534        -2 (self)     00000000
SID   SID_XROUTE                                         -2 (self)     00000000
CID   (default)                                          -7 (fallback) 00000000
IP    (default)                                          -7 (fallback) 00000000
```

Some routes are "control"
(e.g., *self* == this host)

router1's forwarding table

```
XIA Route version 0.9
router1
TYPE  XID                                                  PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                            -7 (fallback) 00000000
AD    AD_NAMESERVER                                         2            0000ffff HID_ROUTER0
AD    AD_SERVERS                                           -2 (self)     00000000
HID   (default)                                            -7 (fallback) 00000000
HID   BHID                                                 -4 (bcast)    00000000
HID   HID_ROUTER1                                          -2 (self)     00000000
HID   HID_ROUTER0                                           2            00000000 HID_ROUTER0
HID   HID_SERVER1                                           1            0000ffff HID_SERVER1
HID   HID_SERVER0                                           0            0000ffff HID_SERVER0
SID   (default)                                            -7 (fallback) 00000000
SID   sid:30000ff0000000000000000000000007b305534          -2 (self)     00000000
SID   SID_XROUTE                                           -2 (self)     00000000
CID   (default)                                            -7 (fallback) 00000000
IP    (default)                                            -7 (fallback) 00000000
```

Some routes are default routes

router1's forwarding table

```
XIA Route version 0.9
router1
TYPE  XID                                                PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                          -7 (fallback) 00000000
AD    AD_NAMESERVER                                       2            0000ffff HID_ROUTER0
AD    AD_SERVERS                                         -2 (self)     00000000
HID   (default)                                          -7 (fallback) 00000000
HID   BHID                                               -4 (bcast)    00000000
HID   HID_ROUTER1                                        -2 (self)     00000000
HID   HID_ROUTER0                                         2            00000000 HID_ROUTER0
HID   HID_SERVER1                                         1            0000ffff HID_SERVER1
HID   HID_SERVER0                                         0            0000ffff HID_SERVER0
SID   (default)                                          -7 (fallback) 00000000
SID   sid:30000ff0000000000000000000000007b305534       -2 (self)     00000000
SID   SID_XROUTE                                         -2 (self)     00000000
CID   (default)                                          -7 (fallback) 00000000
IP    (default)                                          -7 (fallback) 00000000
```

All five principal types are listed

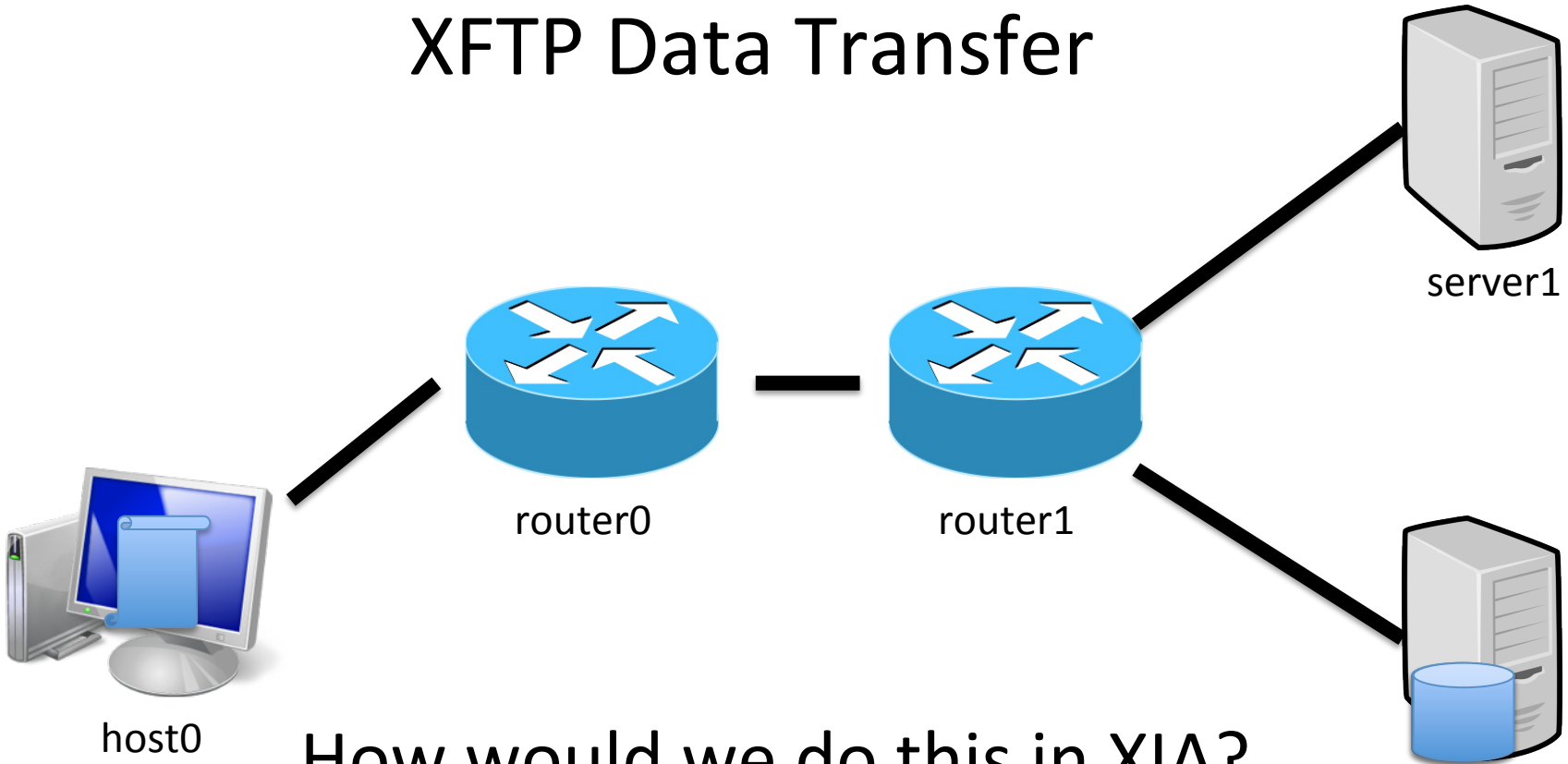
router1's forwarding table

```
XIA Route version 0.9
router1
TYPE  XID                                                    PORT          FLAGS    NEXT HOP
-------------------------------------------------------------------------------------------
AD    (default)                                              -7 (fallback) 00000000
AD    AD:1000000000000000000000000000000000000000          2             0000ffff HID:555555555
AD    AD:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb          -2 (self)     00000000
HID   (default)                                              -7 (fallback) 00000000
HID   HID:ffffffffffffffffffffffffffffffffffffffff         -4 (bcast)    00000000
HID   HID:6666666666666666666666666666666666666666         -2 (self)     00000000
HID   HID:5555555555555555555555555555555555555555          2             00000000 HID:555555555
HID   HID:4444444444444444444444444444444444444444          1             0000ffff HID:444444444
HID   HID:3333333333333333333333333333333333333333          0             0000ffff HID:333333333
SID   (default)                                              -7 (fallback) 00000000
SID   SID:30000ff0000000000000000000000007b305534          -2 (self)     00000000
SID   SID:1110000000000000000000000000000000001112          -2 (self)     00000000
CID   (default)                                              -7 (fallback) 00000000
IP    (default)                                              -7 (fallback) 00000000
```

Raw XIDs

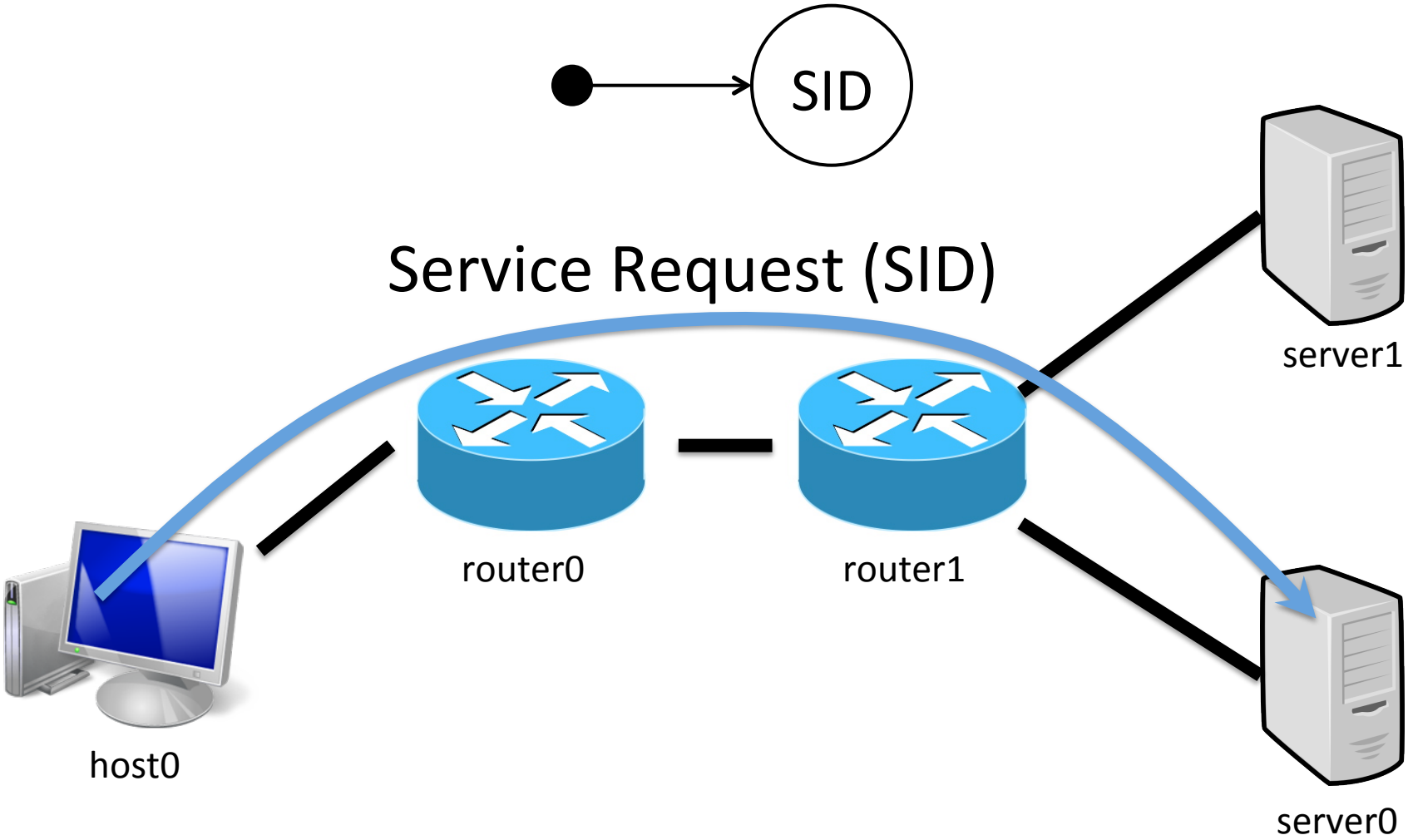
$bin/xroute –v

XFTP Data Transfer



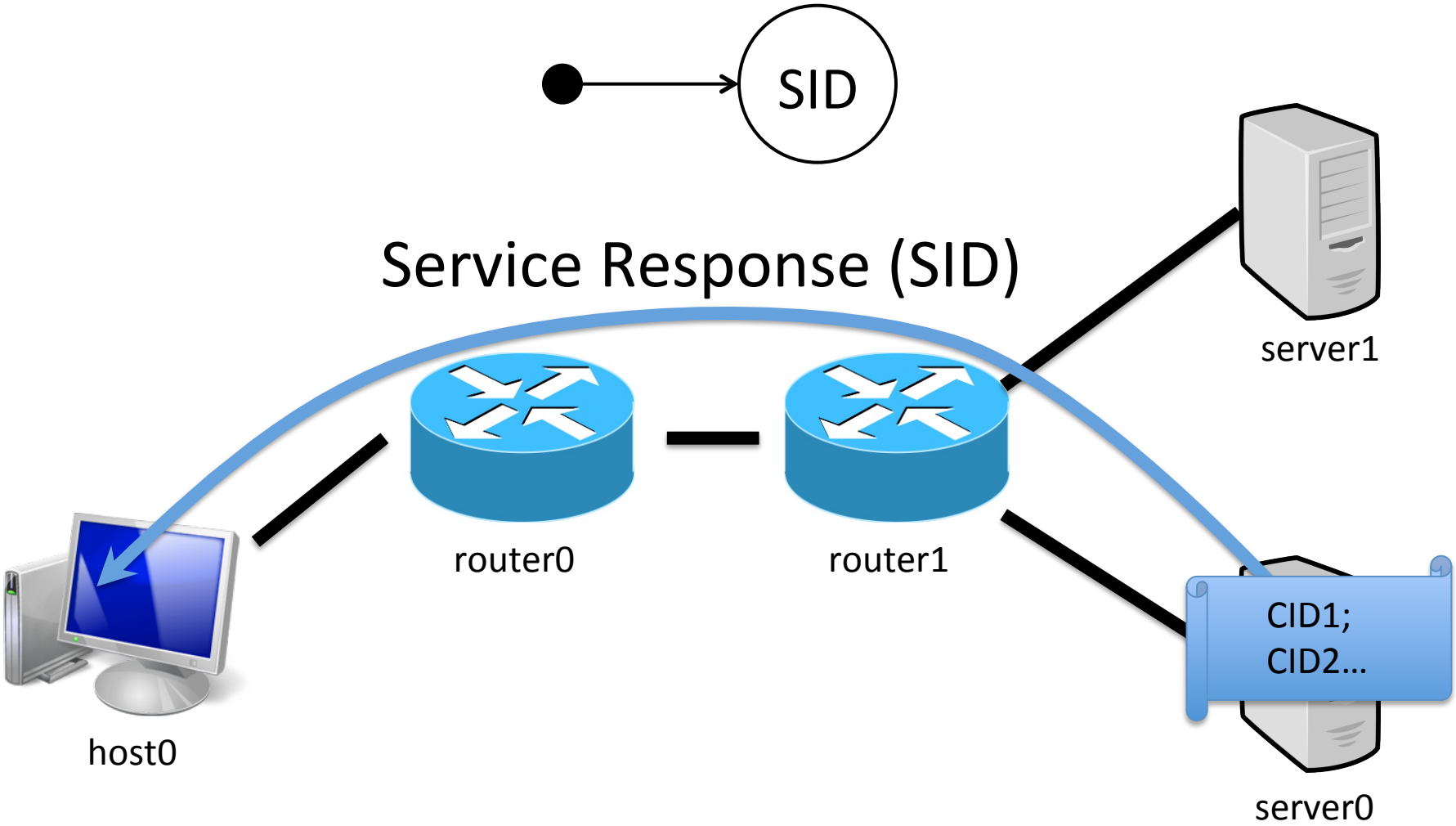
How would we do this in XIA?
(i.e., with services and content ID)

Request DAG (address)



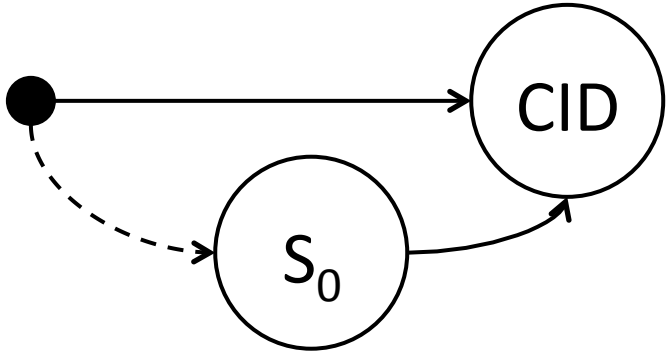
get sample.txt sample.txt.1

Request DAG (address)

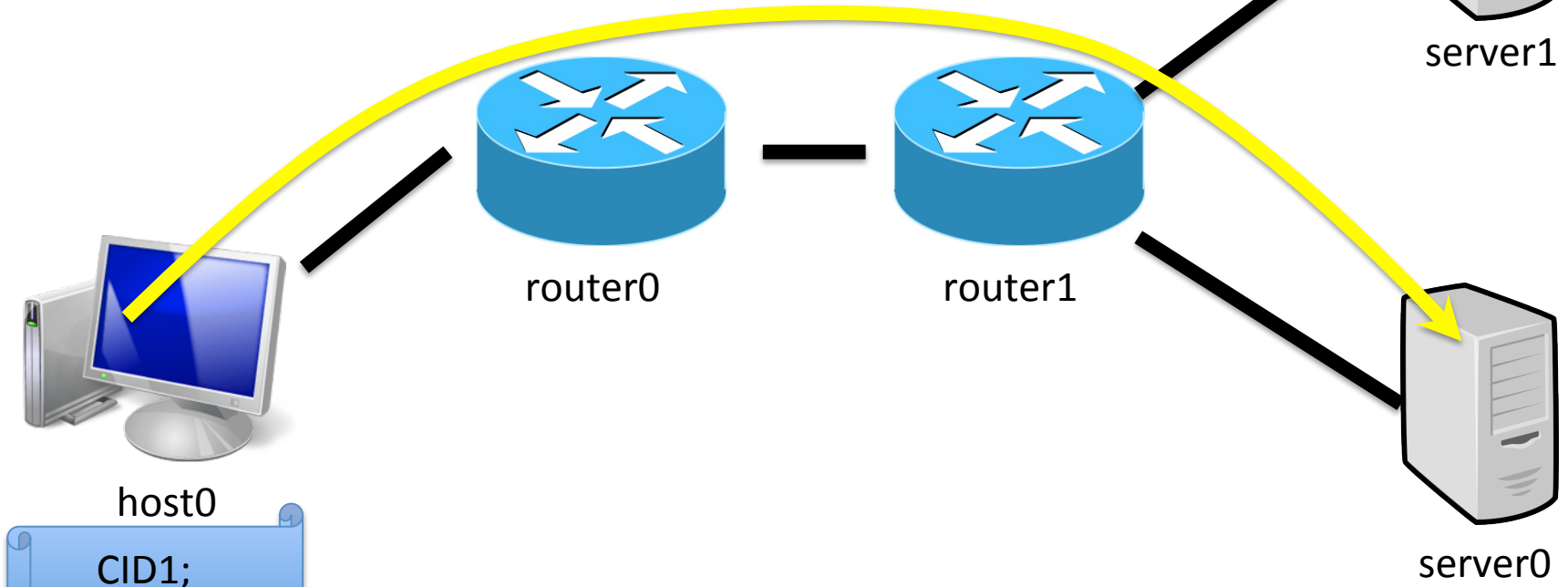


get sample.txt sample.txt.1

Request DAG (address)



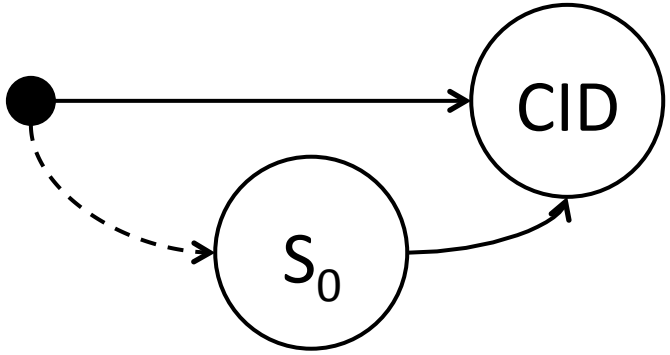
Content Requests (CIDs)



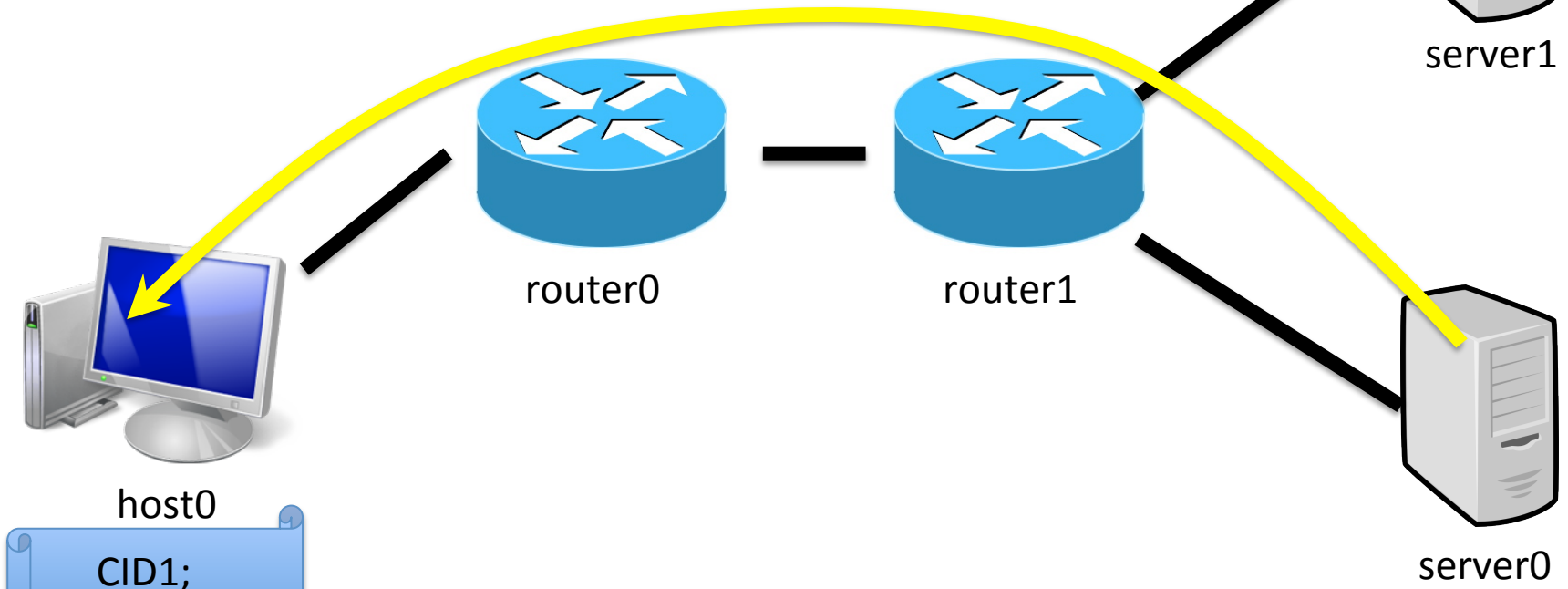
CID1;
CID2…

get sample.txt sample.txt.1

Request DAG (address)

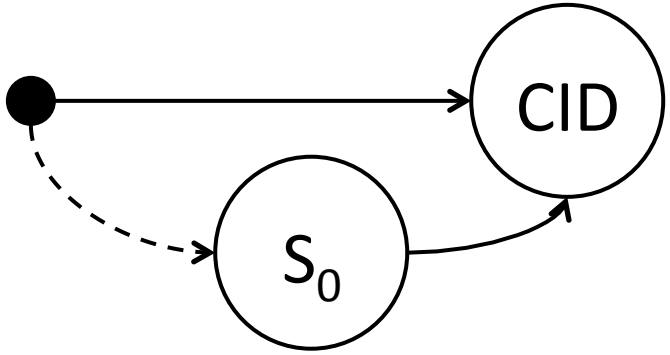


Content Response (CIDs)

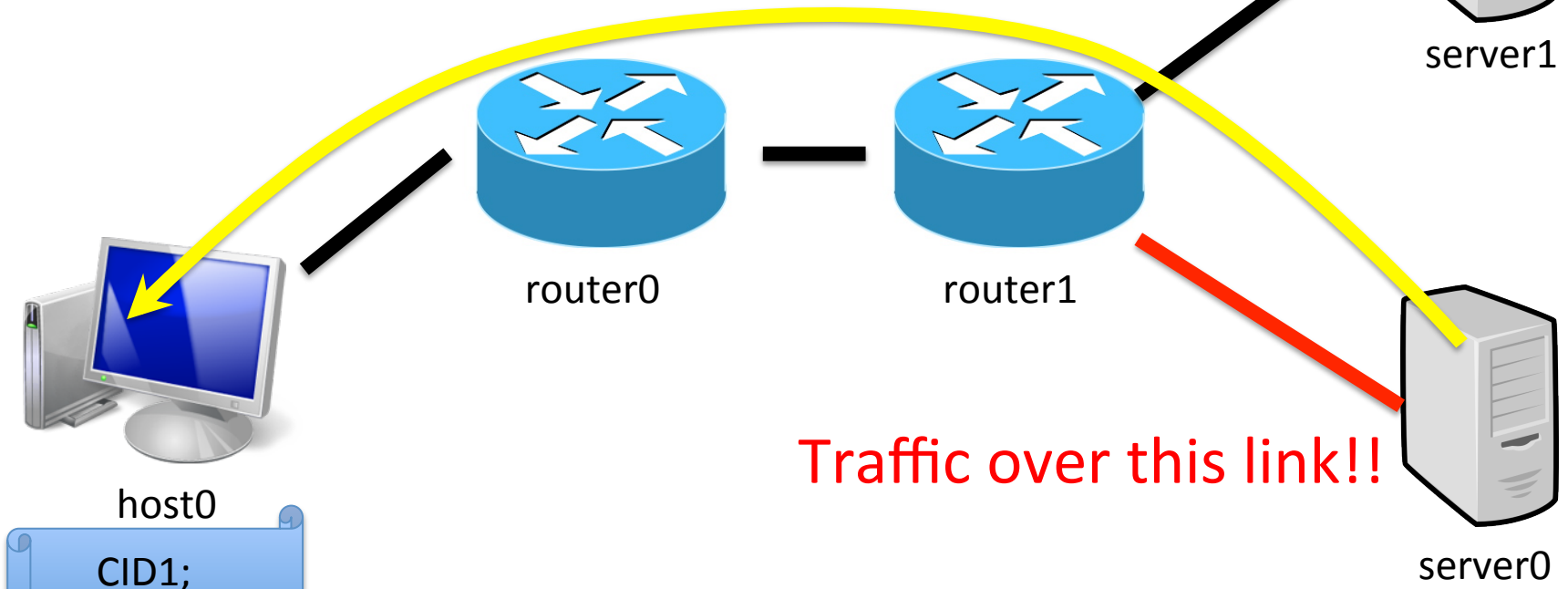


get sample.txt sample.txt.1

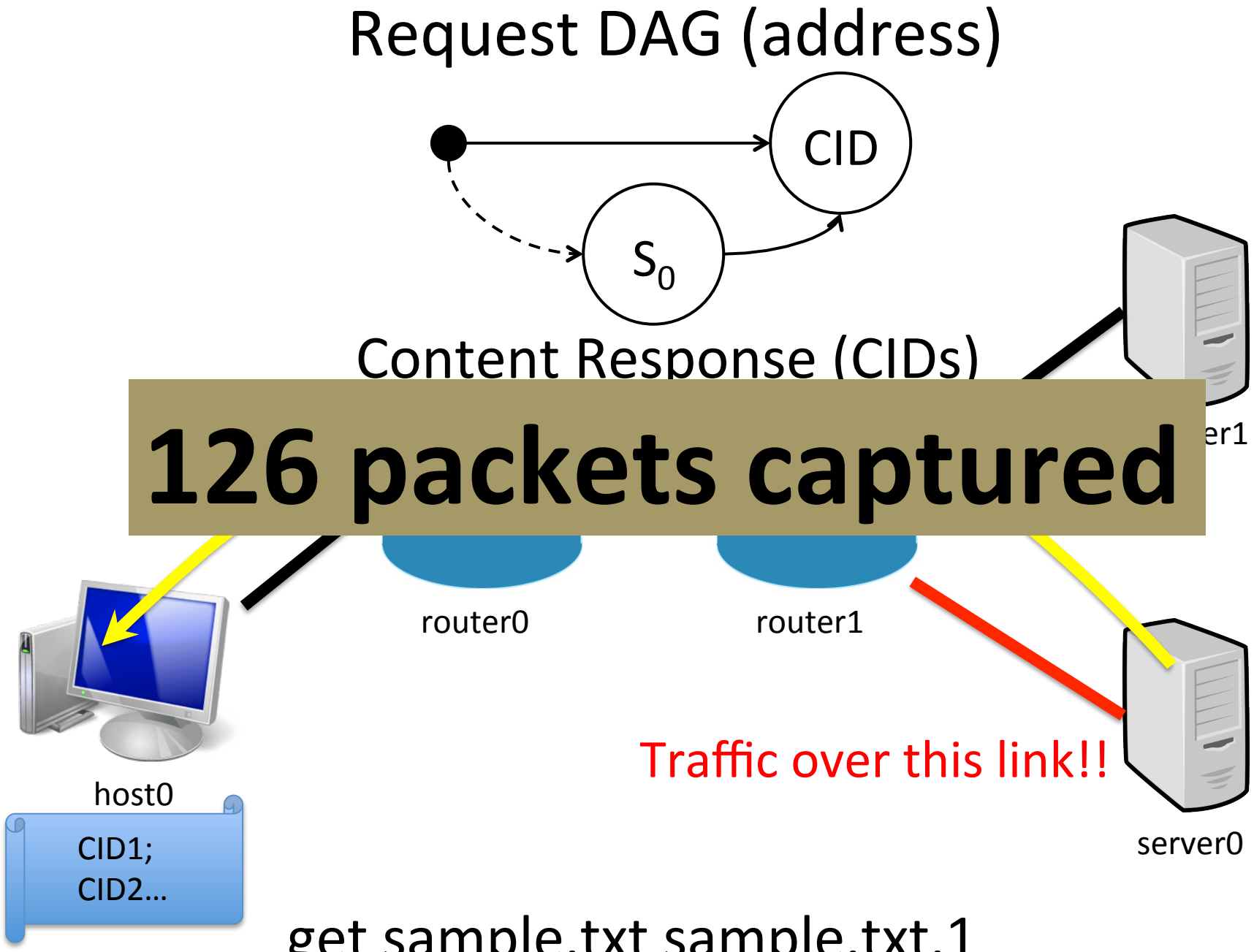
Request DAG (address)



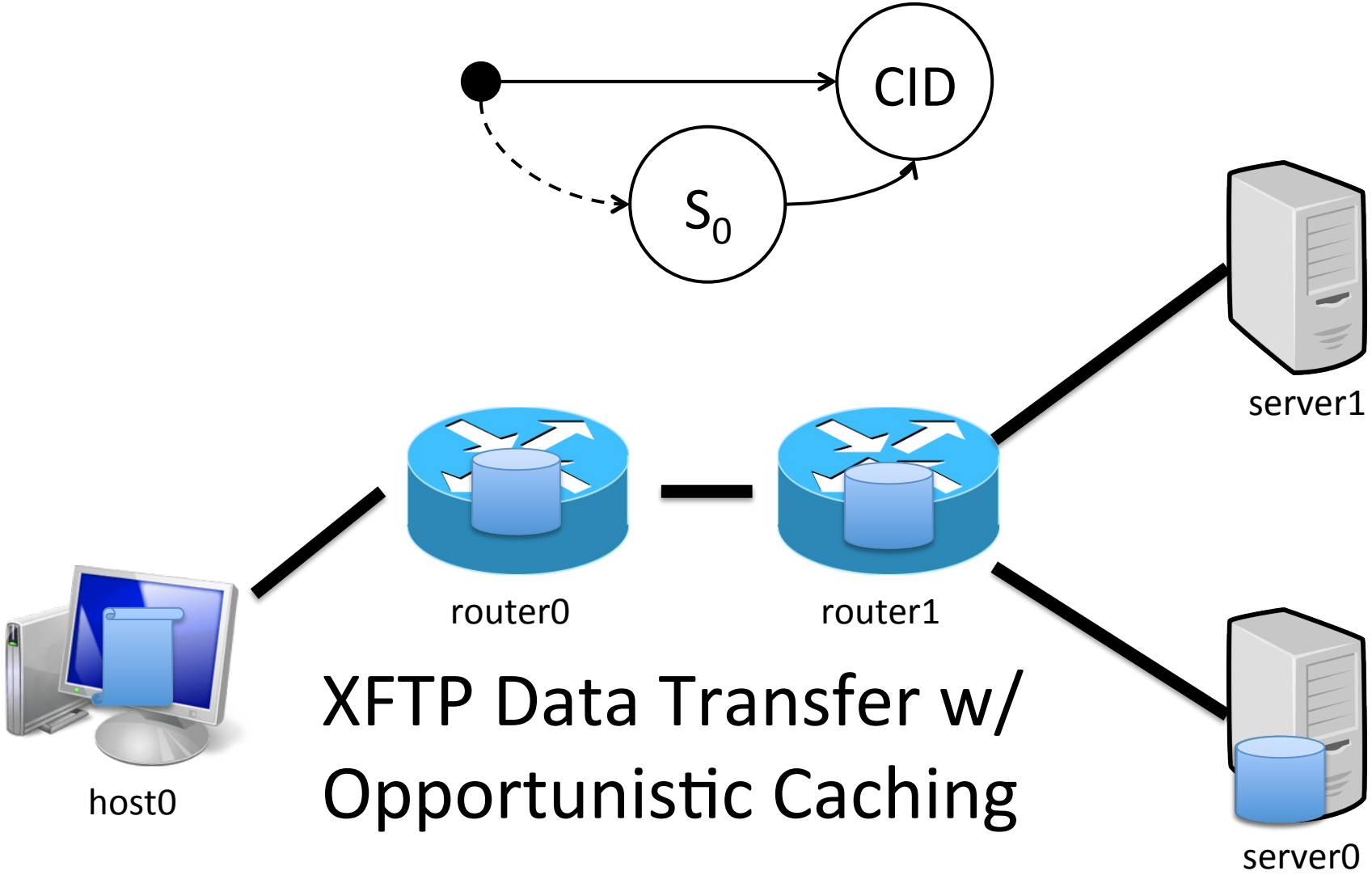
Content Response (CIDs)



get sample.txt sample.txt.1


Request DAG (address)
CID
S_0
Content Response (CIDs)
126 packets captured
router0
router1
Traffic over this link!!
host0
CID1;
CID2…
server0
get sample.txt sample.txt.1

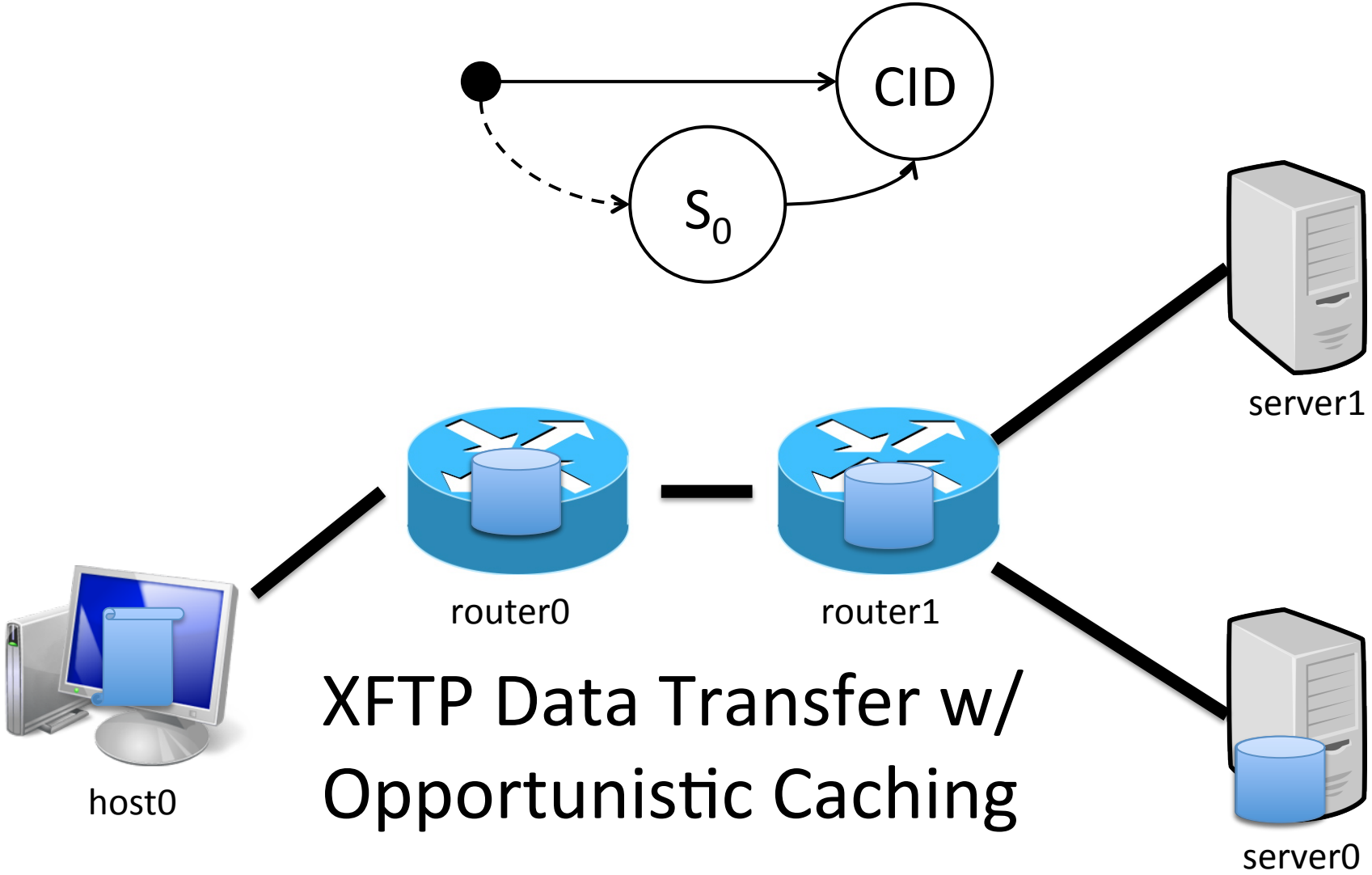
Request DAG (address)



XFTP Data Transfer w/ Opportunistic Caching

get sample.txt sample.txt.2

Request DAG (address)



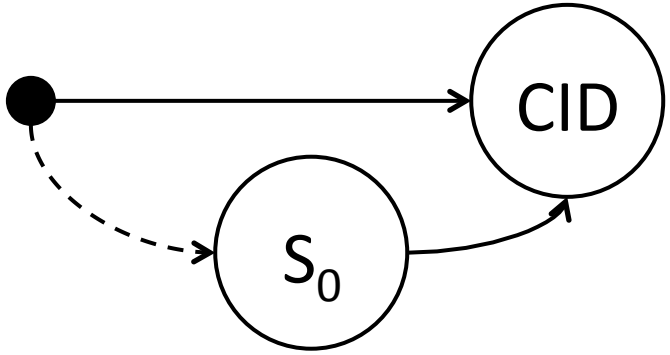
XFTP Data Transfer w/ Opportunistic Caching

get sample.txt sample.txt.2

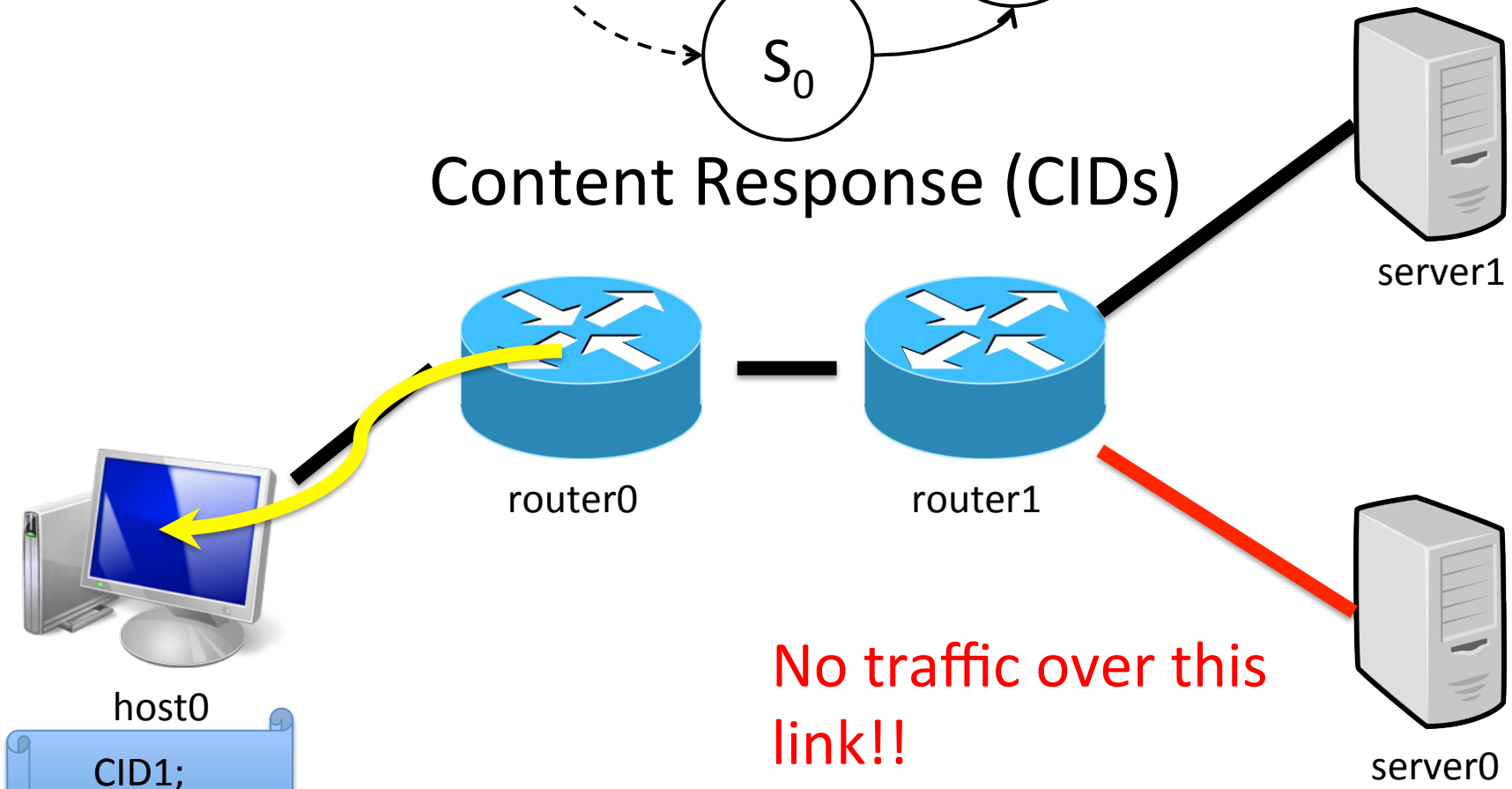
router1's populated cache

Cached content

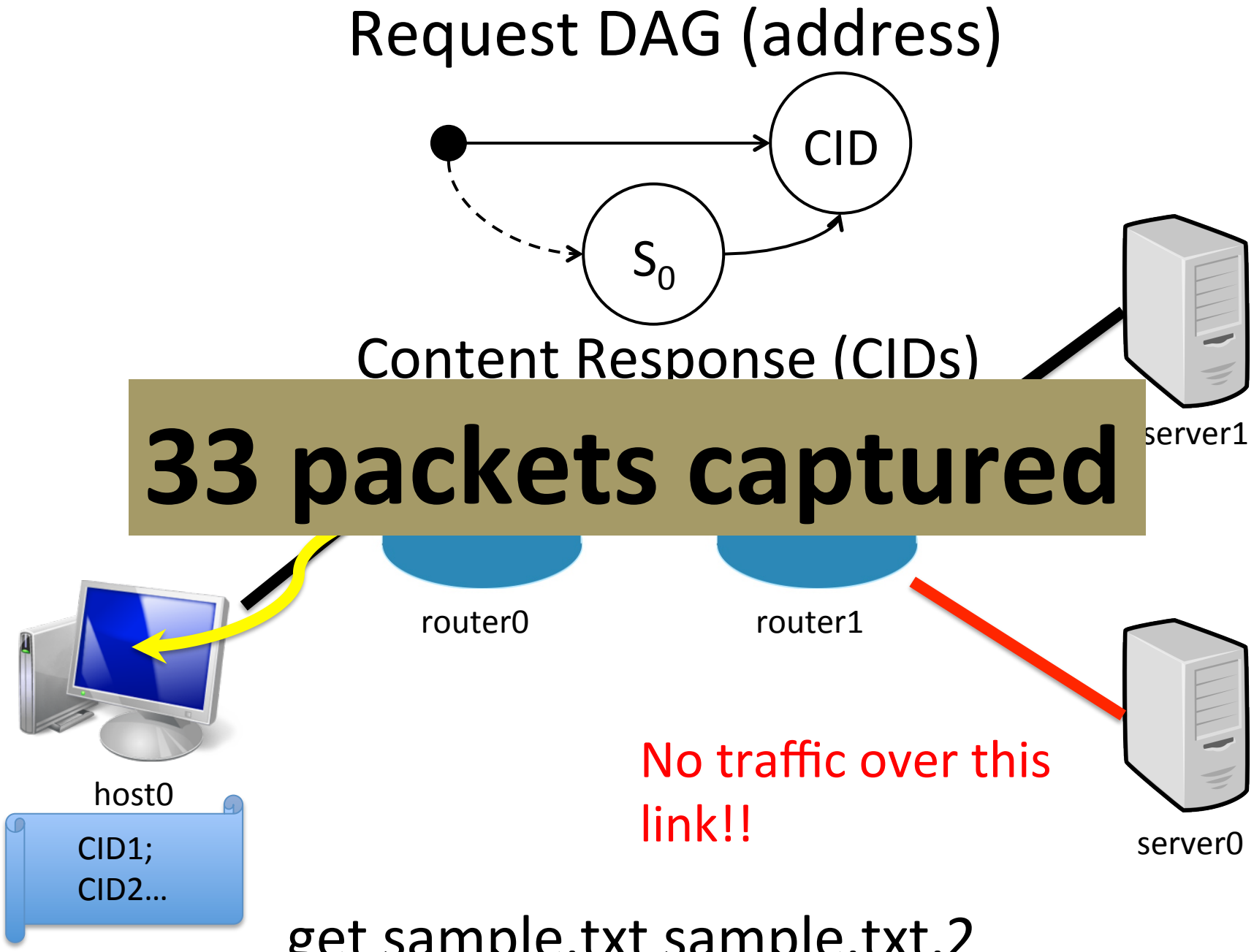
Request DAG (address)



Content Response (CIDs)



get sample.txt sample.txt.2


Request DAG (address)
CID
S_0
Content Response (CIDs)
33 packets captured
server1
router0
router1
host0
CID1;
CID2…
No traffic over this link!!
server0
get sample.txt sample.txt.2

XIA as A Research Platform

XIA GENI Tutorial Part 2B

XIA Focuses on Evolvability

- XIA is design to embrace novel networking ideas
- It is simple and fast to implement and verify new designs on XIA
- With good backward-compatibility to existing components.

How to Extend XIA

- Define the **Principal**:
  Addressing and intrinsic security
- Define its **Control Plane**
  How to build the forwarding table
- Define its **Per-hop Behavior**
  How the routers treat this new principal
- Define **APIs**
  How end hosts treat this new principal

Optimizing Content Centric Networks

- The CID principal
  Hash value as a CID for a data chunk
- Control Plane:
  Algorithms to prefetch static data
  Protocol to schedule links and path for live video
- Per-Hop:
  Cache data on the fly
- APIs:
  XrequestChunk() XputChunk() …

Routing Protocols

Security Enhancement

Mobile Connectivities

New Ideas?

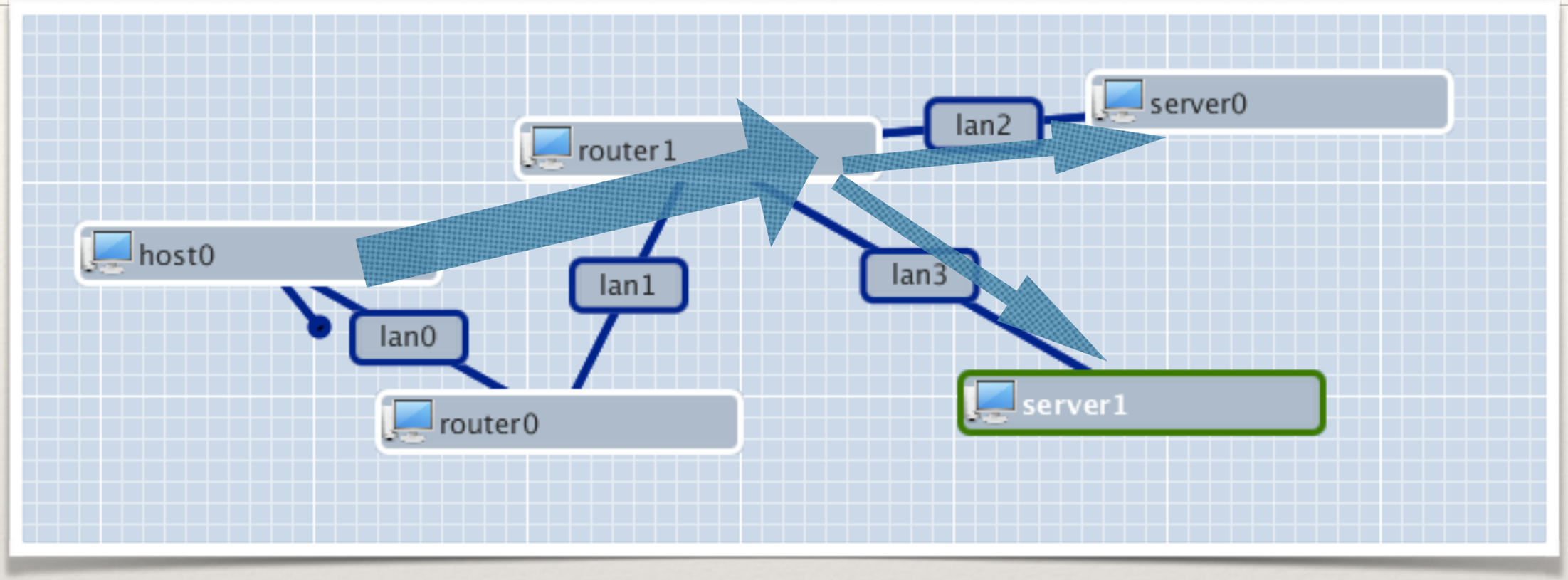
Forwarding Behaviors

Addressing Schemes

Possible Research Directions

Principal\Network Component	Per-hop	Ctl Plane	Addressing/ Intrinsic Security	APIs
CID	Yes	Yes	Yes	Yes
Multicast	Yes	Yes	Yes	-
Mobile	?	Yes	-	?
QoS	Yes	Yes	?	?
?	?	?	?	?
Load balancing	**Yes**	**Yes**	**?**	**-**

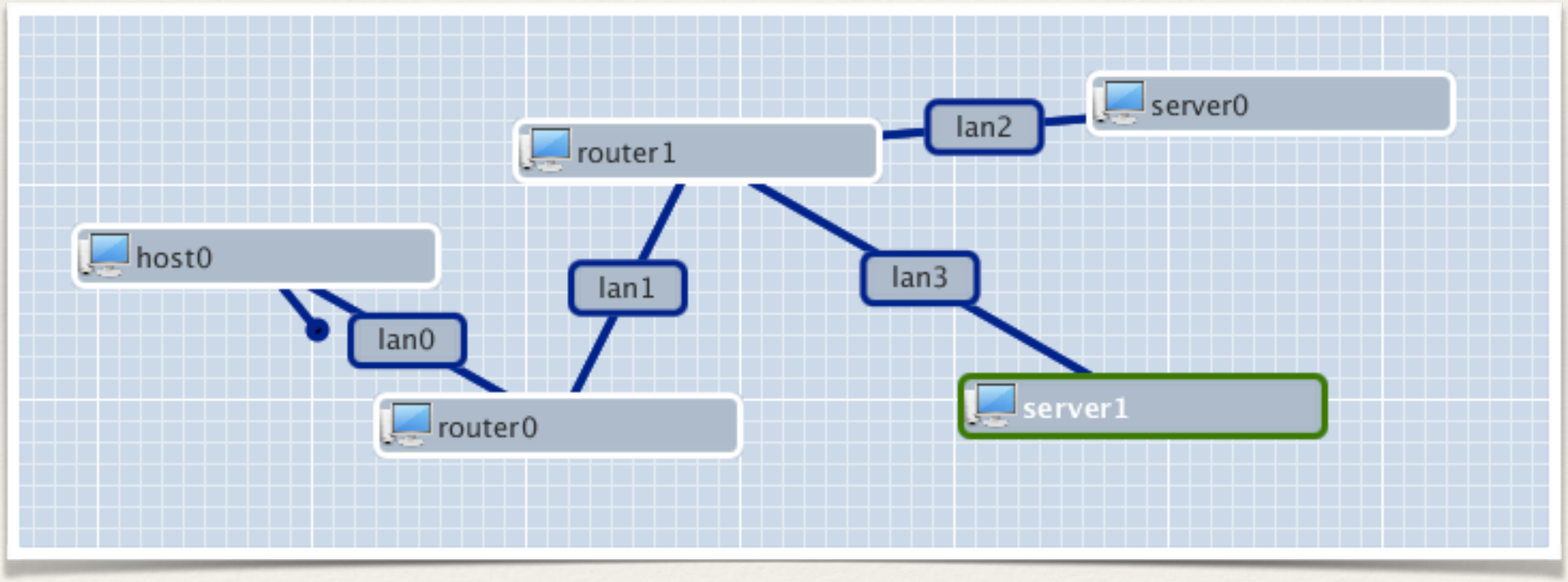
Add Load Balancing Principal



Goal:

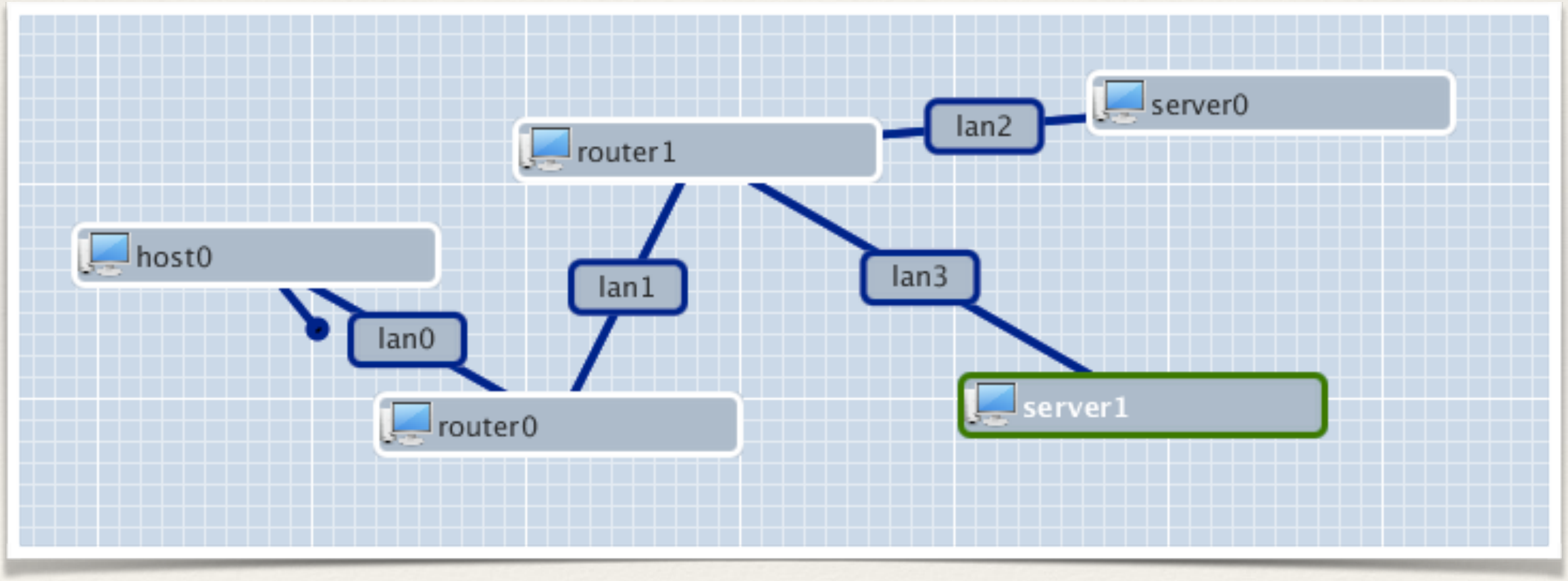
- Balance the load between two servers
- By only updating a small portion of the network
- Within 20 mins

Outline



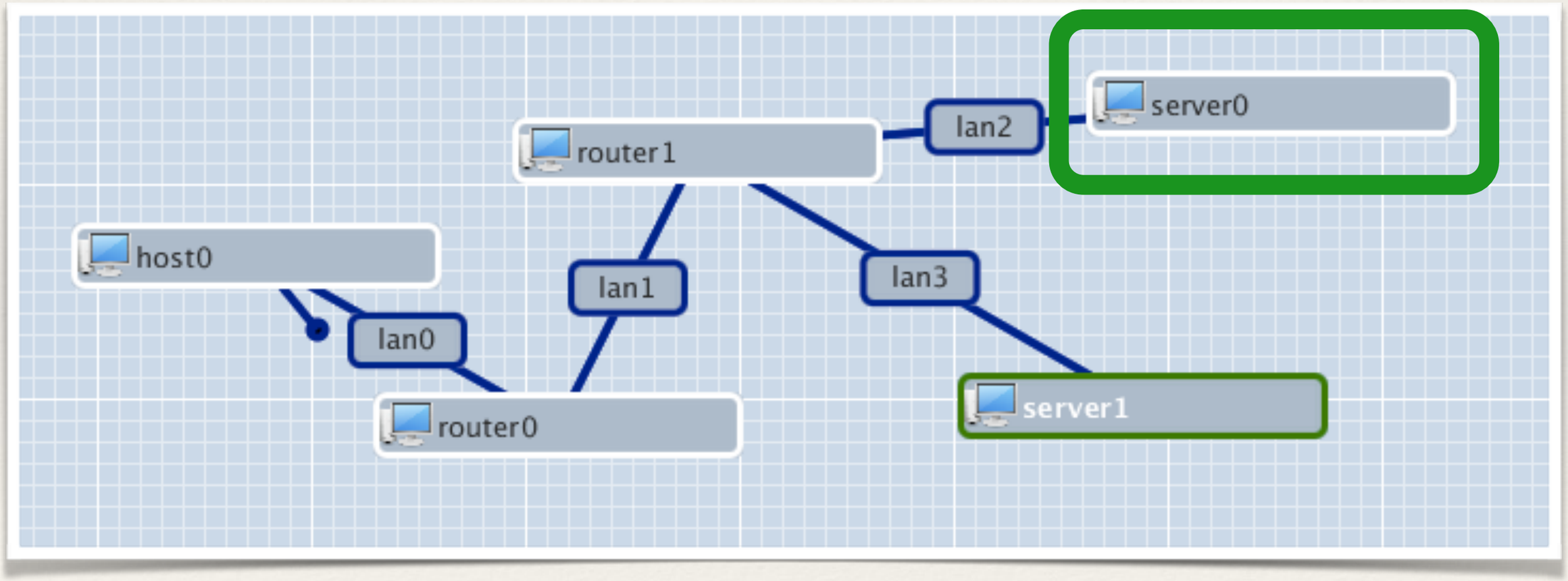
Outline

1. Bring up two service replicas



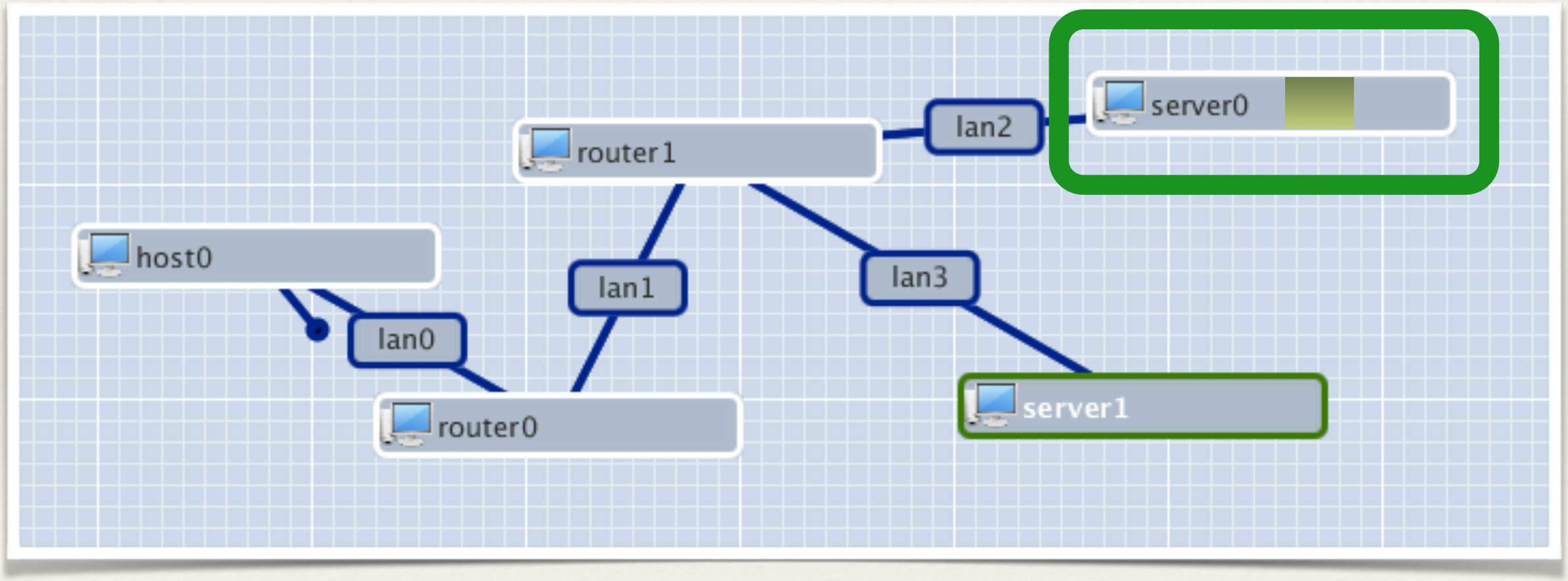
Outline

1. Bring up two service replicas



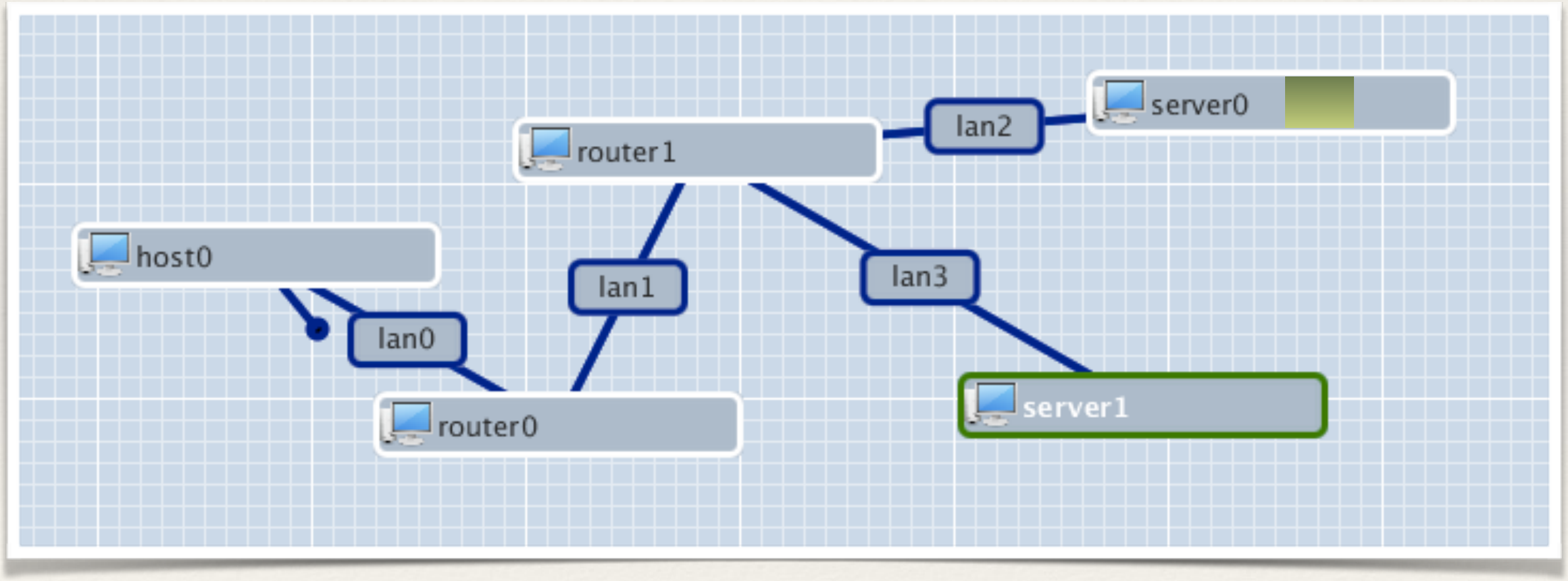
Outline

1. Bring up two service replicas



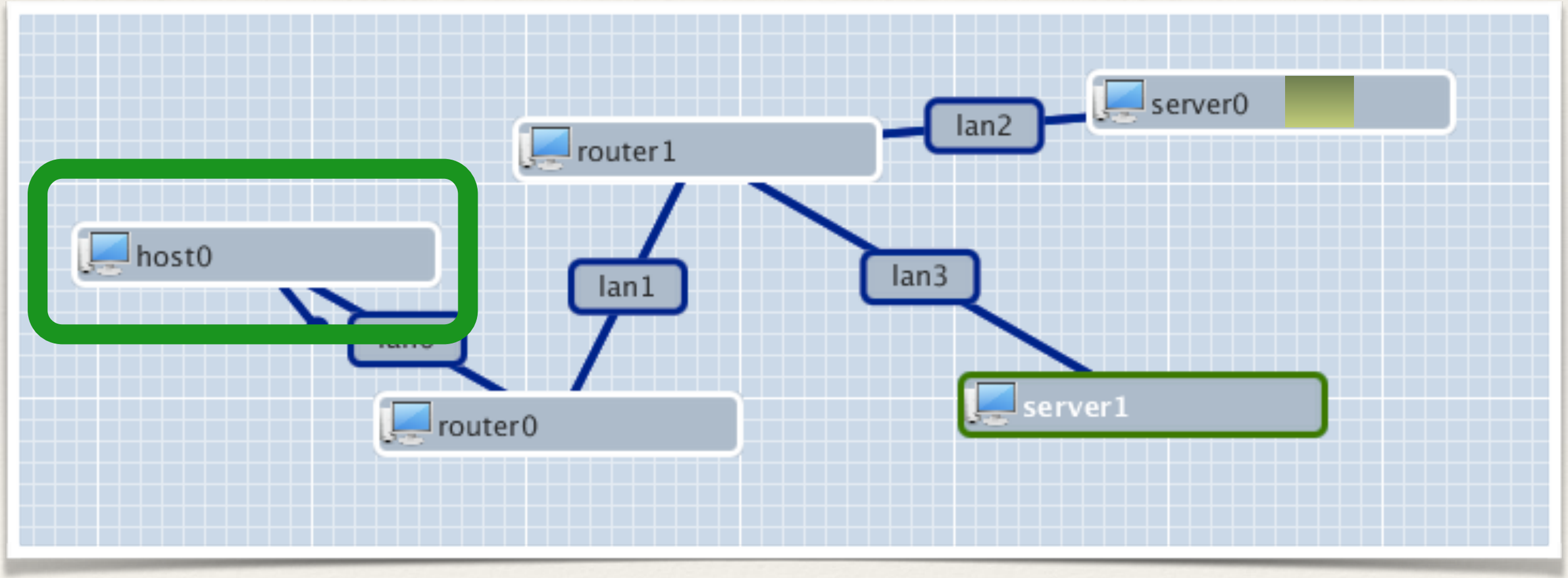
Outline

1. Bring up two service replicas



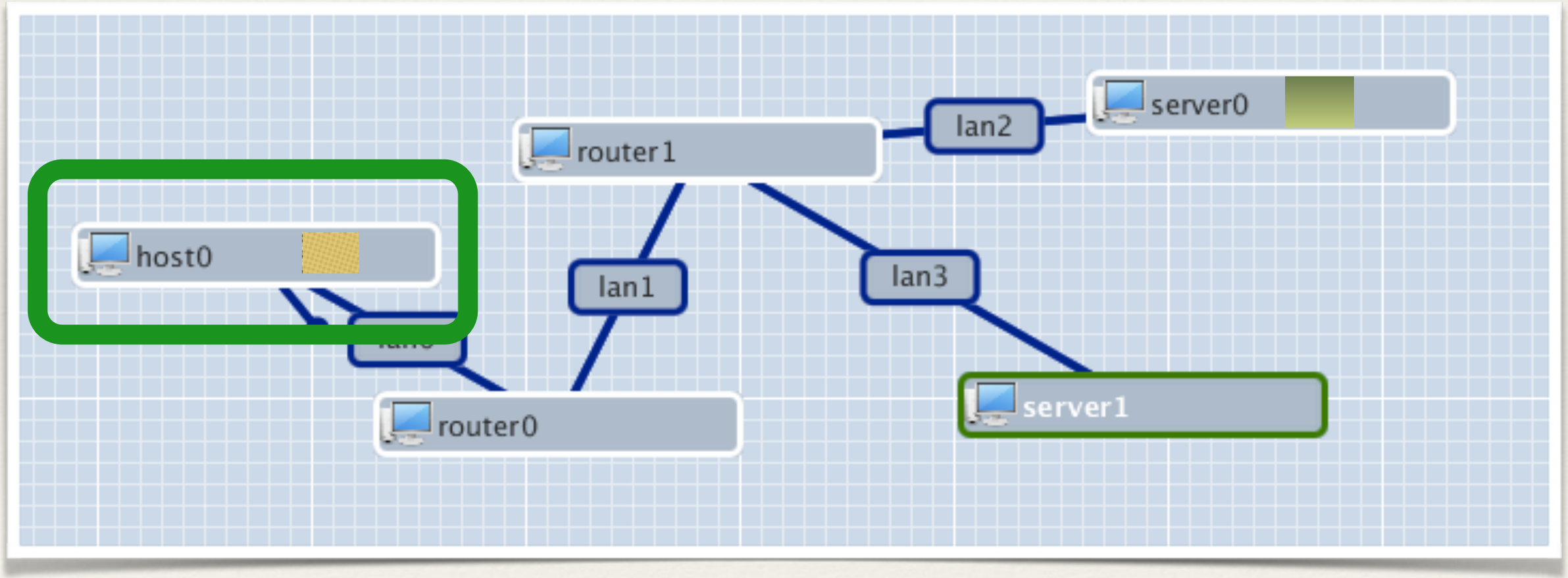
Outline

1. Bring up two service replicas



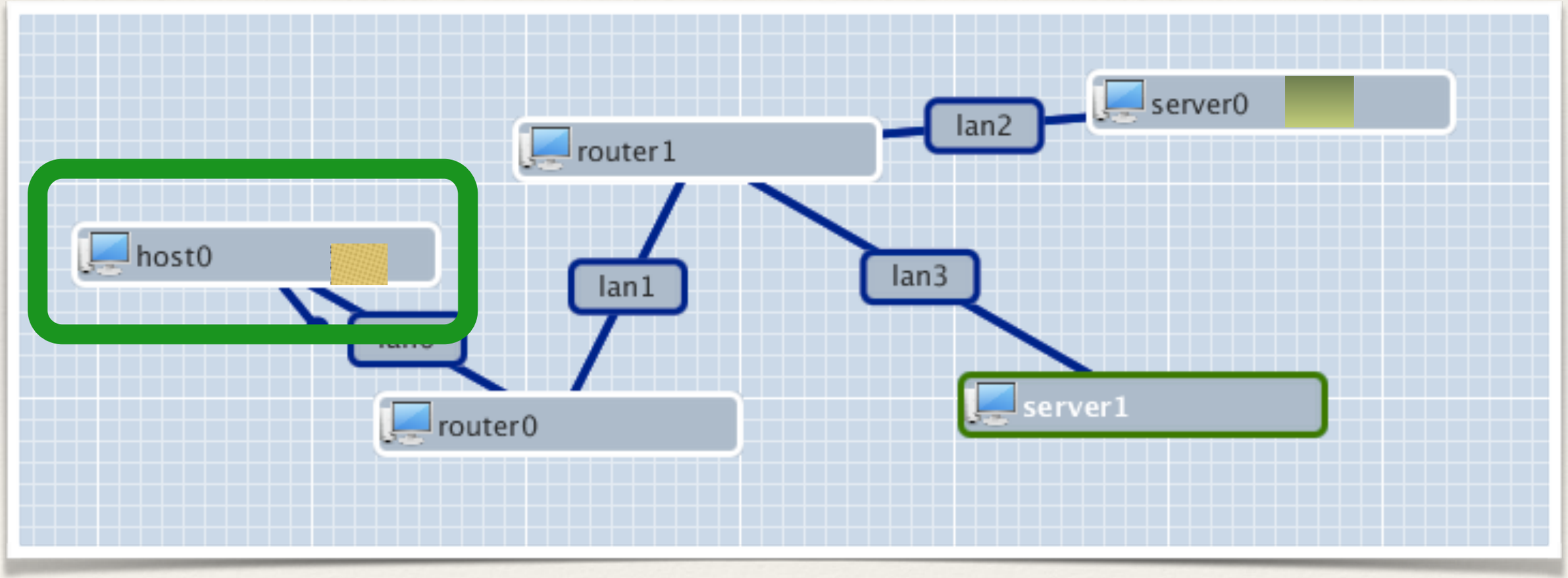
Outline

1. Bring up two service replicas



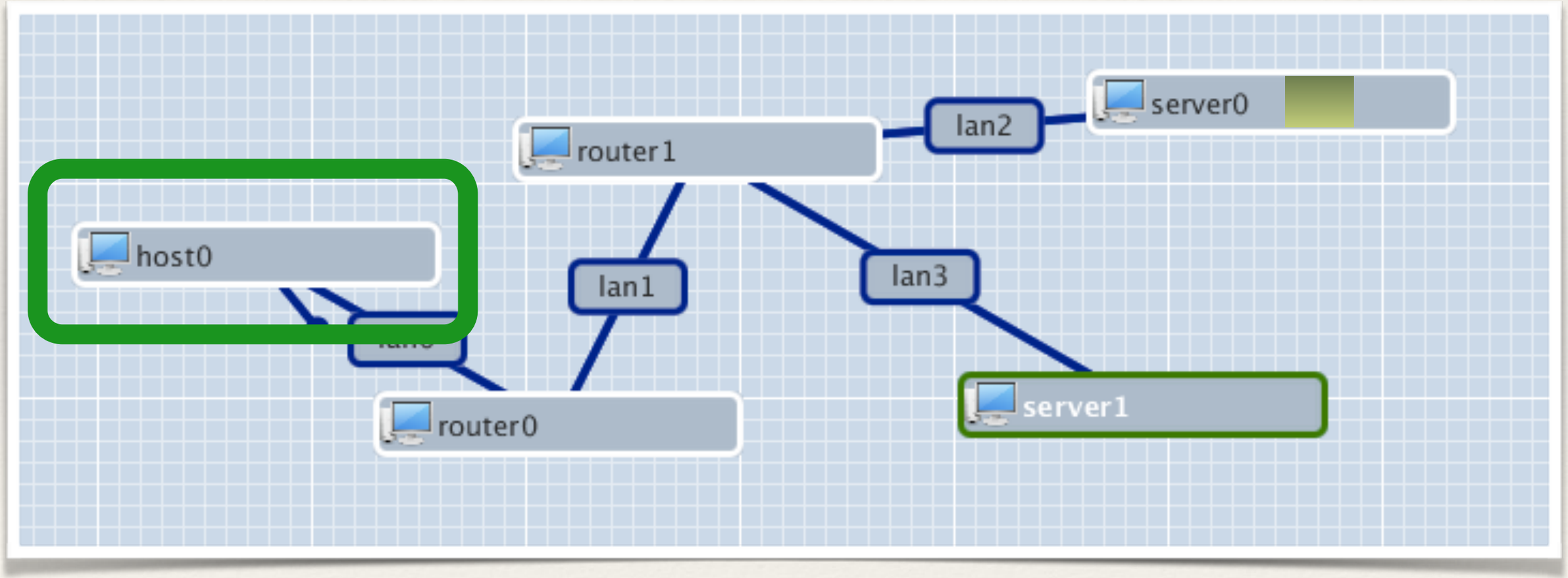
Outline

1. Bring up two service replicas



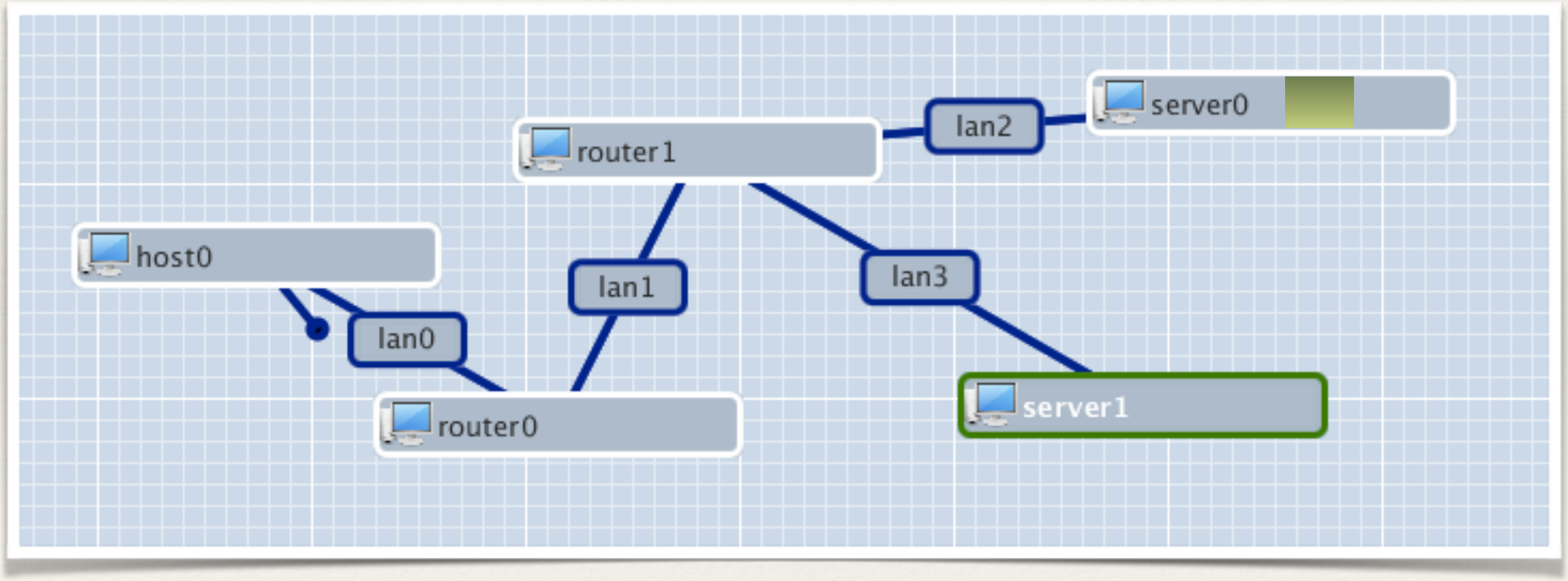
Outline

1. Bring up two service replicas



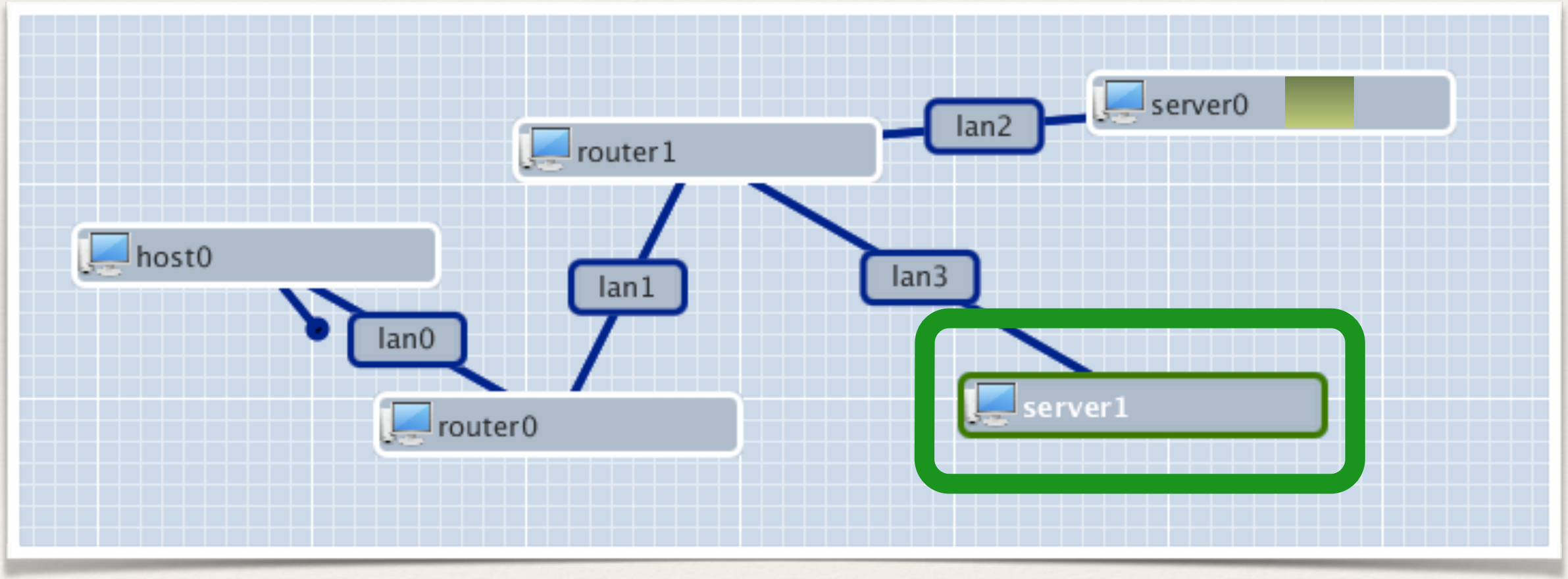
Outline

1. Bring up two service replicas



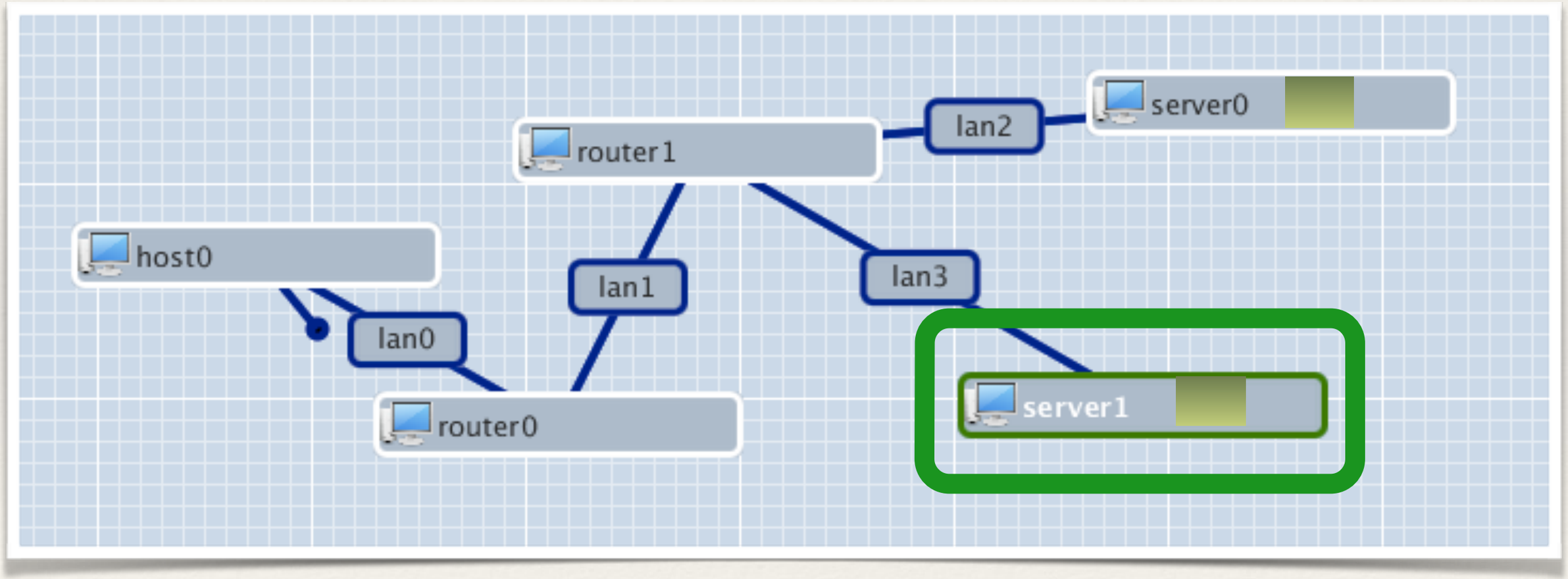
Outline

1. Bring up two service replicas



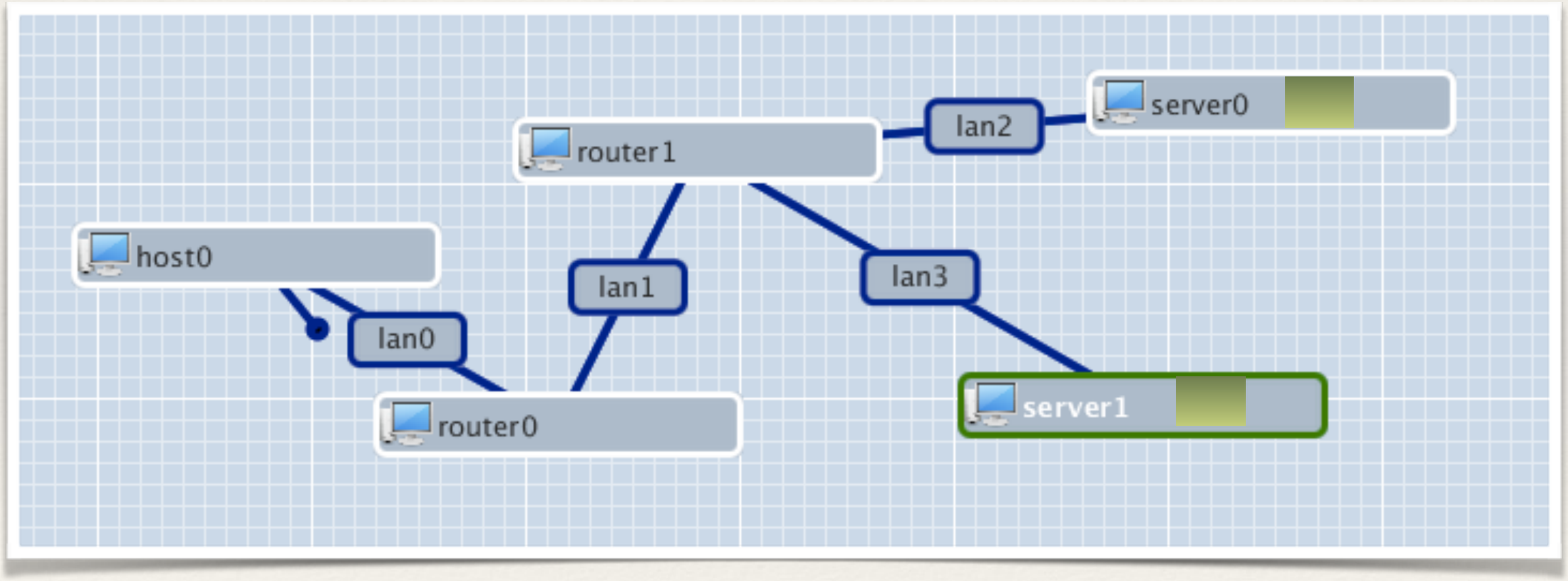
Outline

1. Bring up two service replicas



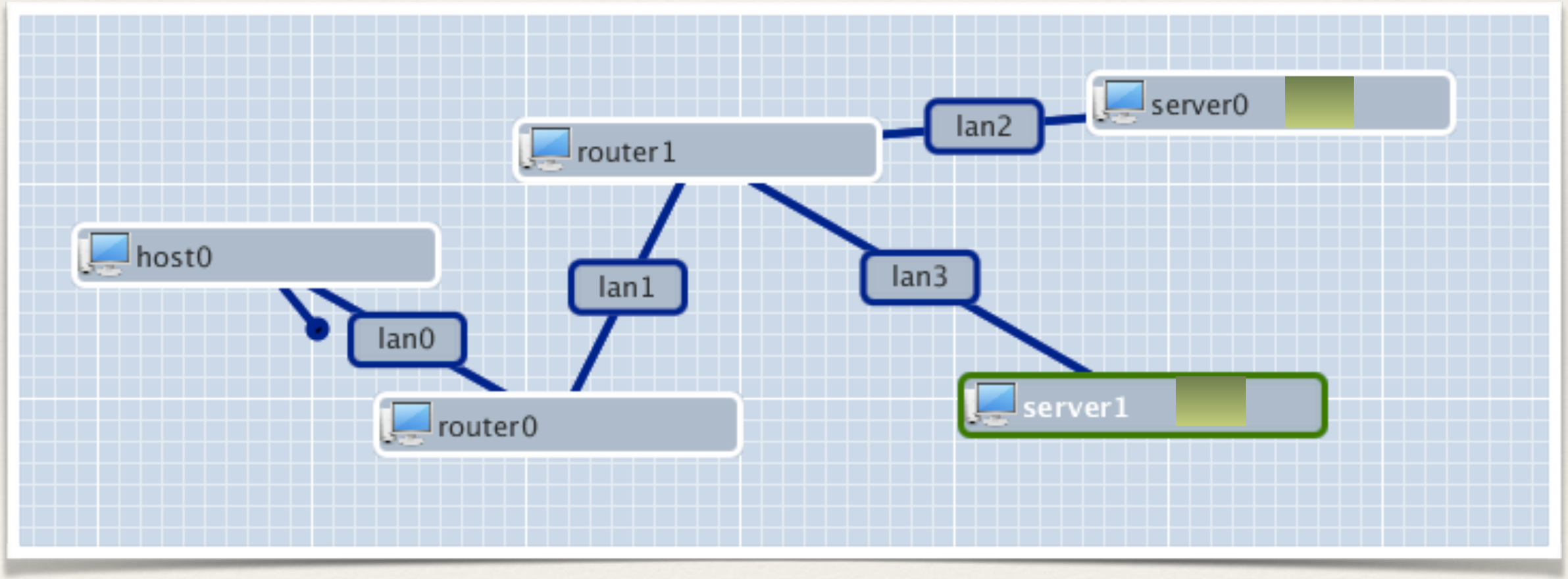
Outline

1. Bring up two service replicas



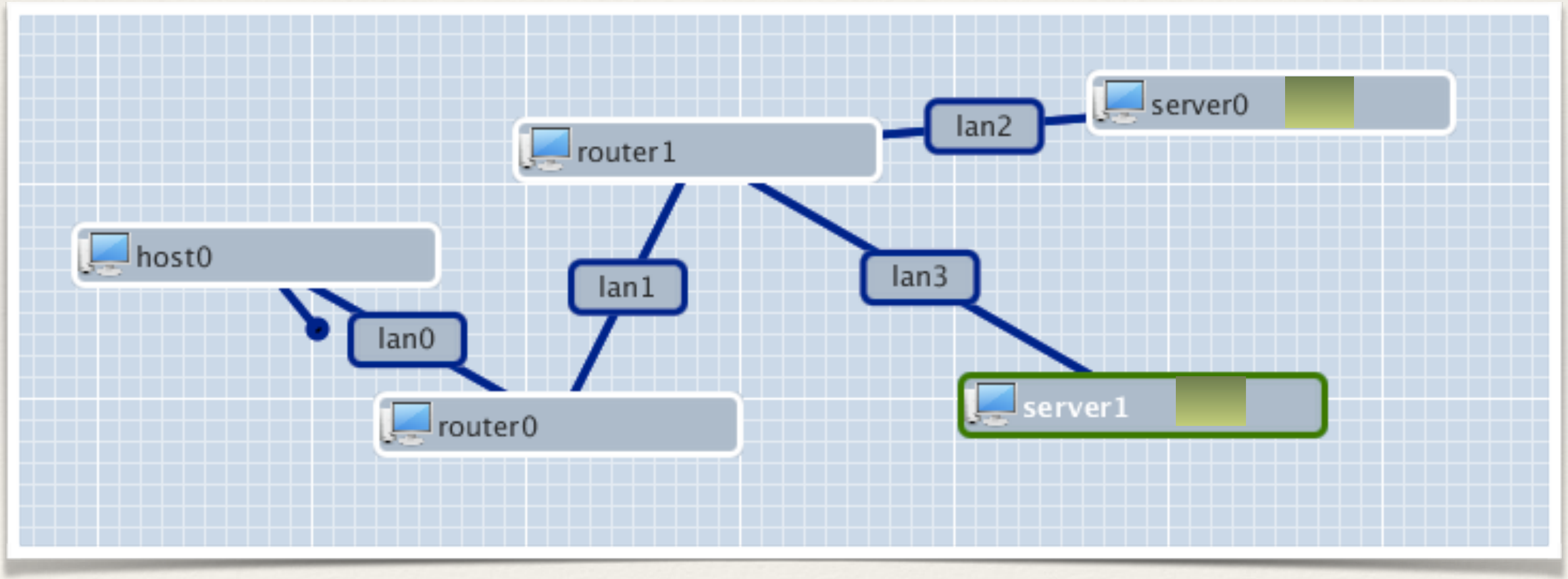
Outline

1. Bring up two service replicas
2. Add new principal type: LID



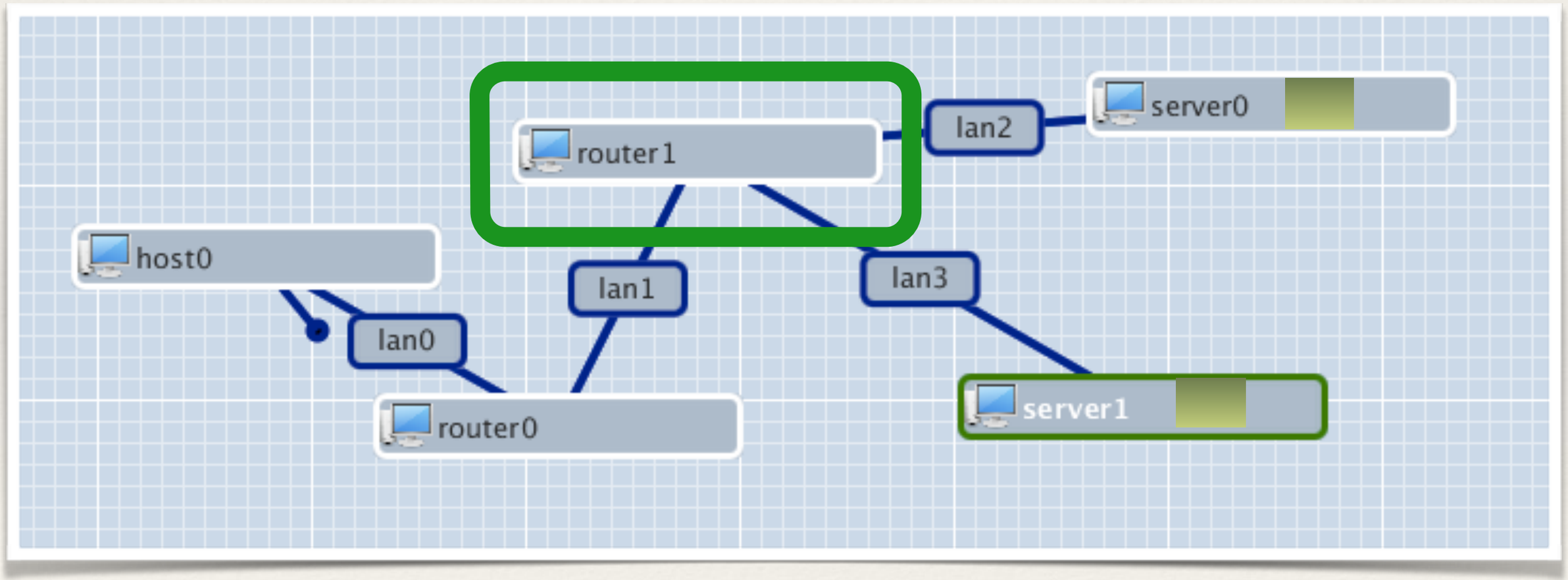
Outline

1. Bring up two service replicas
2. Add new principal type: LID
3. Add a new forwarding engine



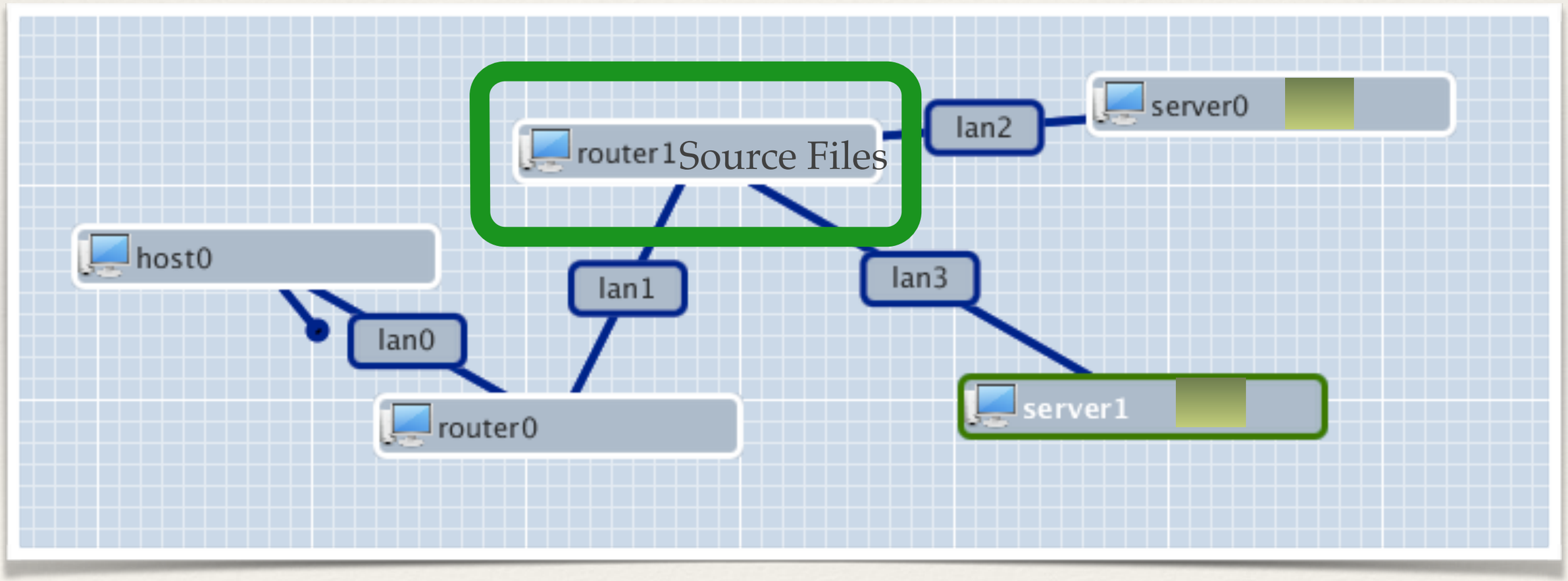
Outline

1. Bring up two service replicas
2. Add new principal type: LID
3. Add a new forwarding engine



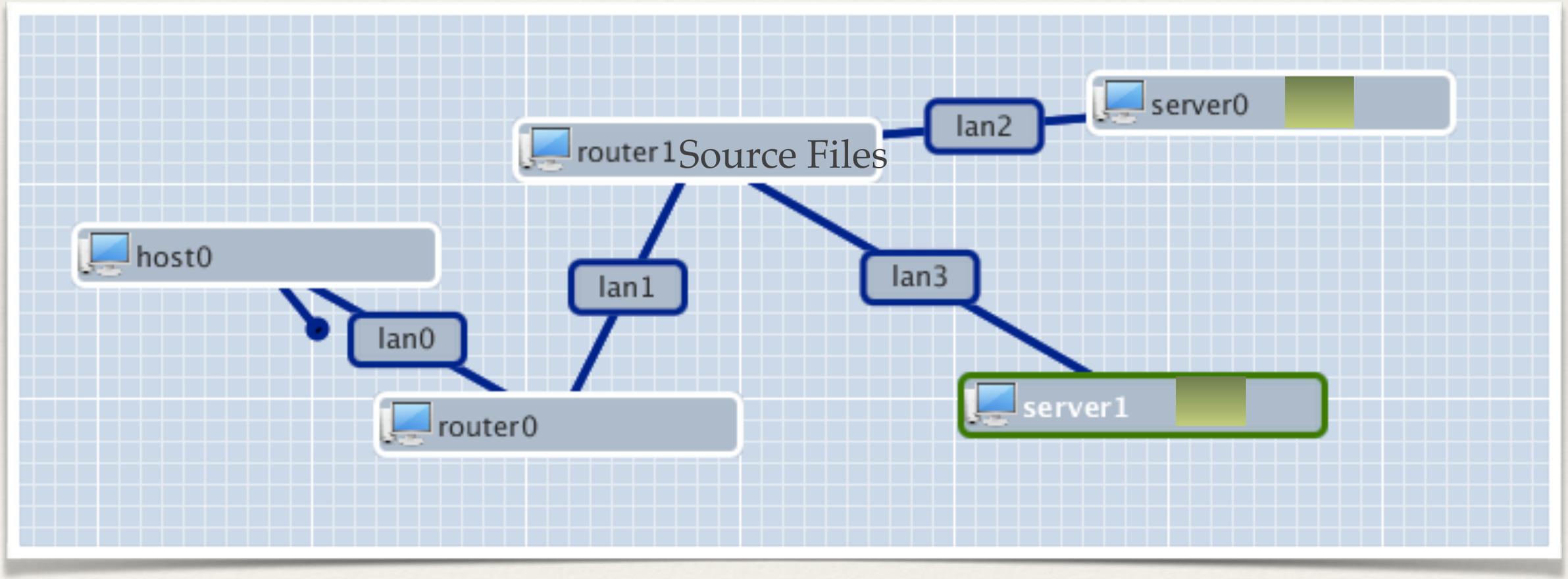
Outline

1. Bring up two service replicas
2. Add new principal type: LID
3. Add a new forwarding engine



Outline

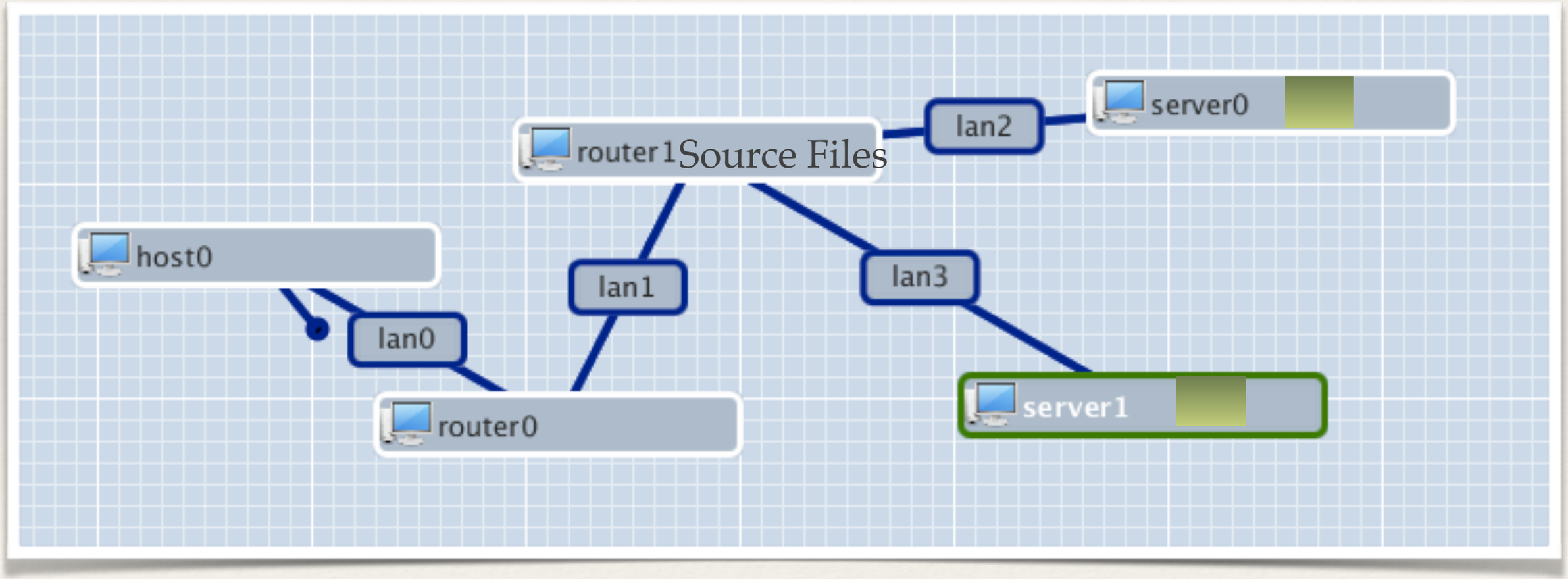
1. Bring up two service replicas
2. Add new principal type: LID
3. Add a new forwarding engine



Outline

1. Bring up two service replicas 2. Add new principal type: LID

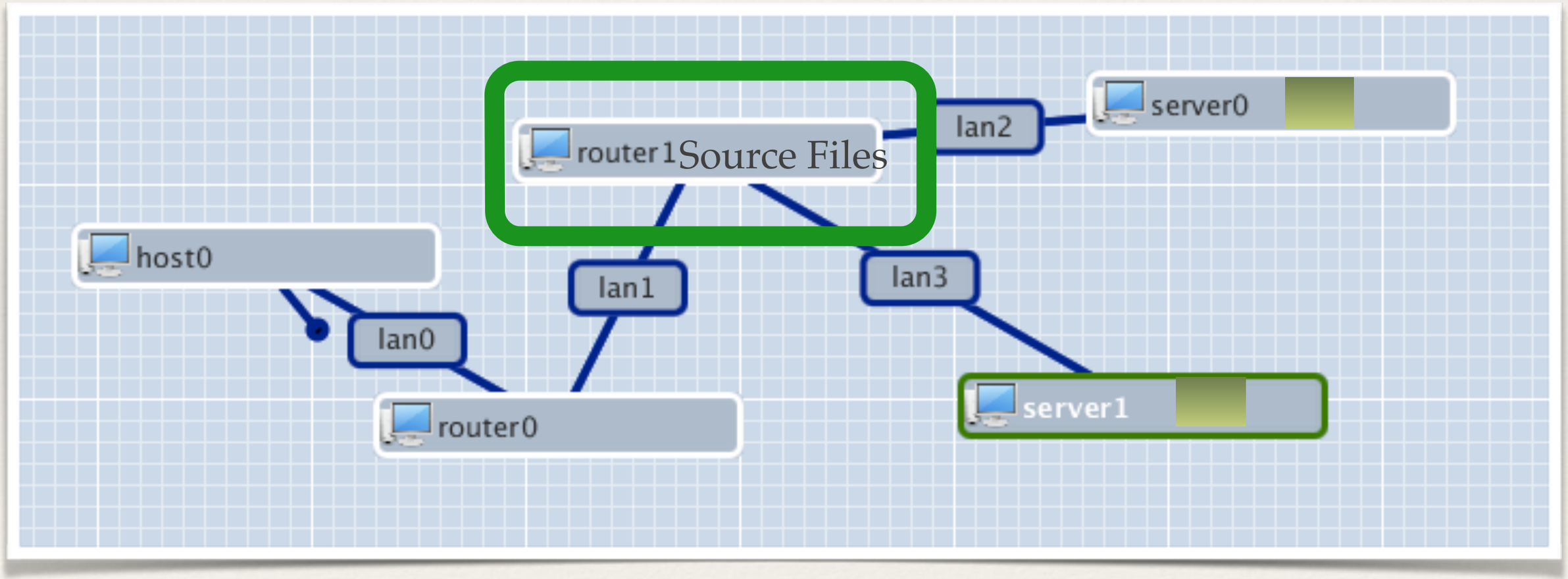
3. Add a new forwarding engine 4. Compile and reboot network 5. add routes



Outline

1. Bring up two service replicas 2. Add new principal type: LID

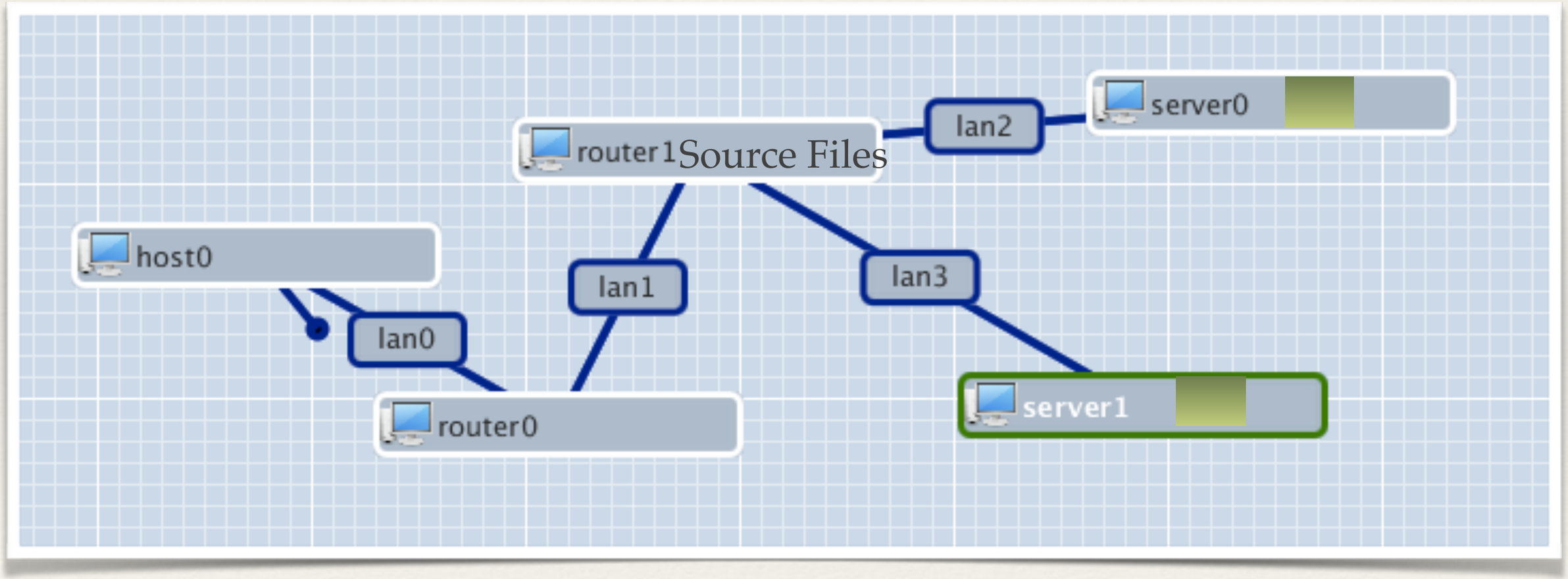
3. Add a new forwarding engine 4. Compile and reboot network 5. add routes



Outline

1. Bring up two service replicas 2. Add new principal type: LID

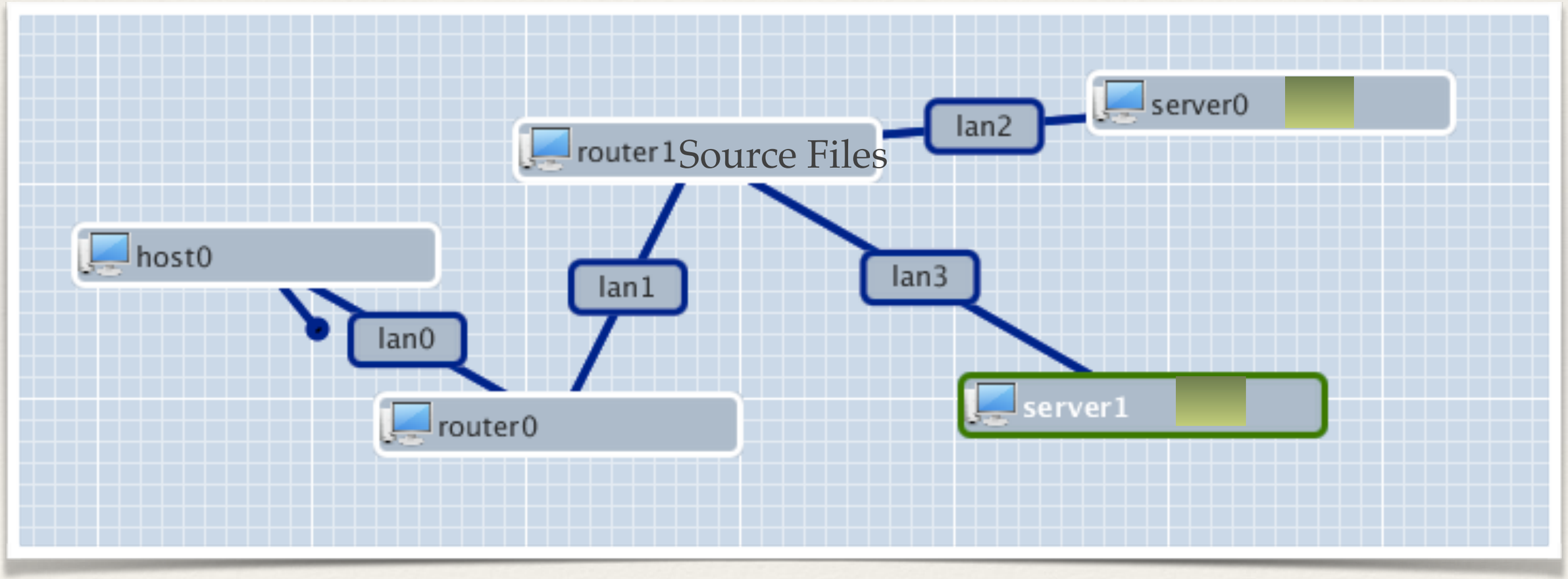
3. Add a new forwarding engine 4. Compile and reboot network 5. add routes



Outline

1. Bring up two service replicas 2. Add new principal type: LID

3. Add a new forwarding engine 4. Compile and reboot network 5. add routes

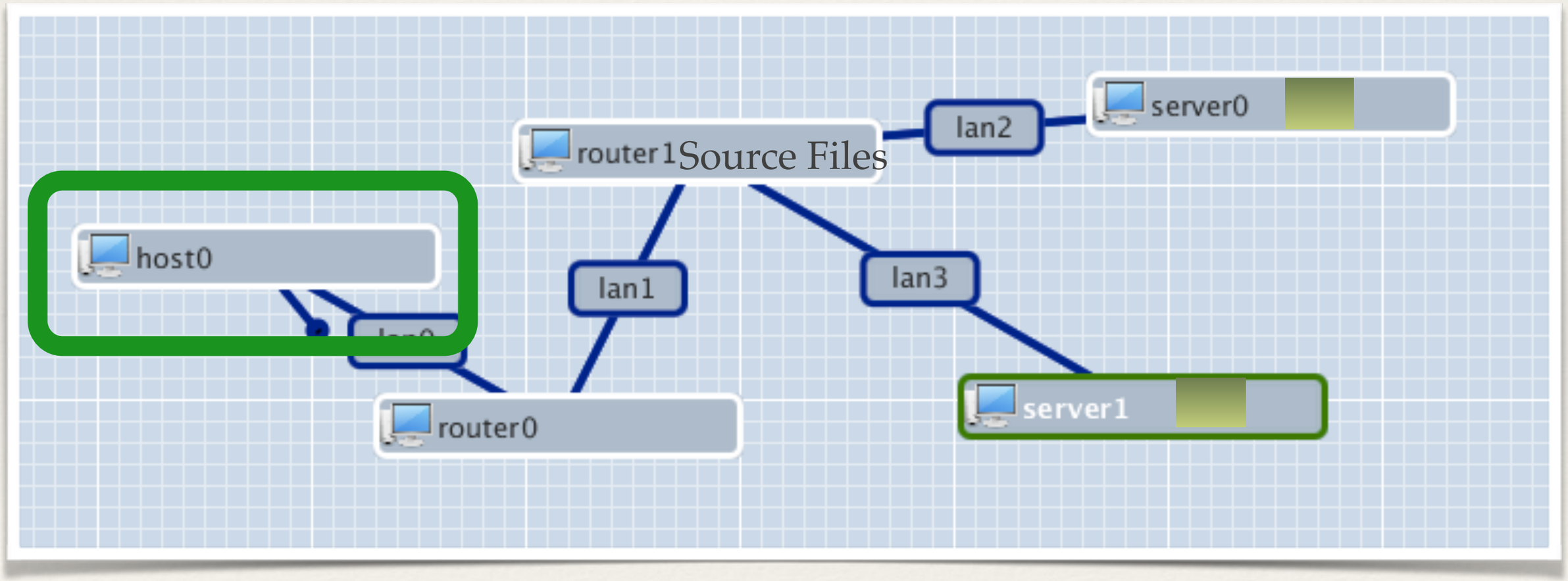


6. Verify the behavior of LID

Outline

1. Bring up two service replicas 2. Add new principal type: LID

3. Add a new forwarding engine 4. Compile and reboot network 5. add routes

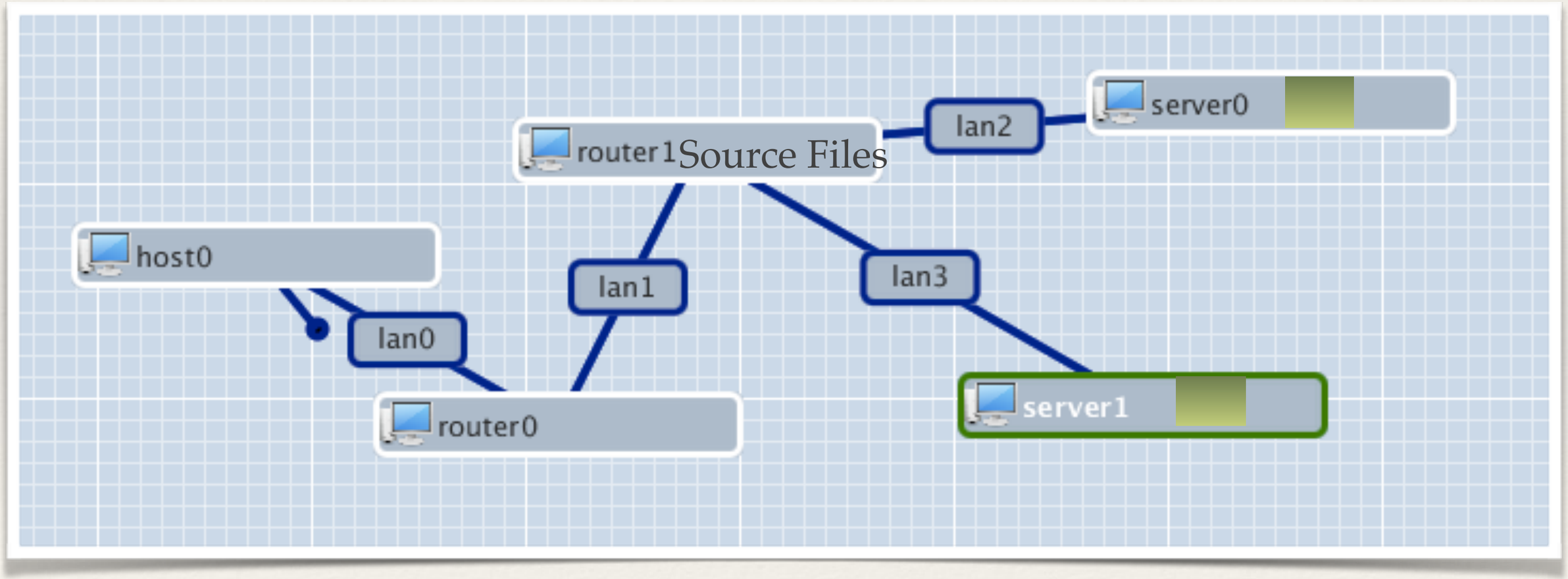


6. Verify the behavior of LID

Outline

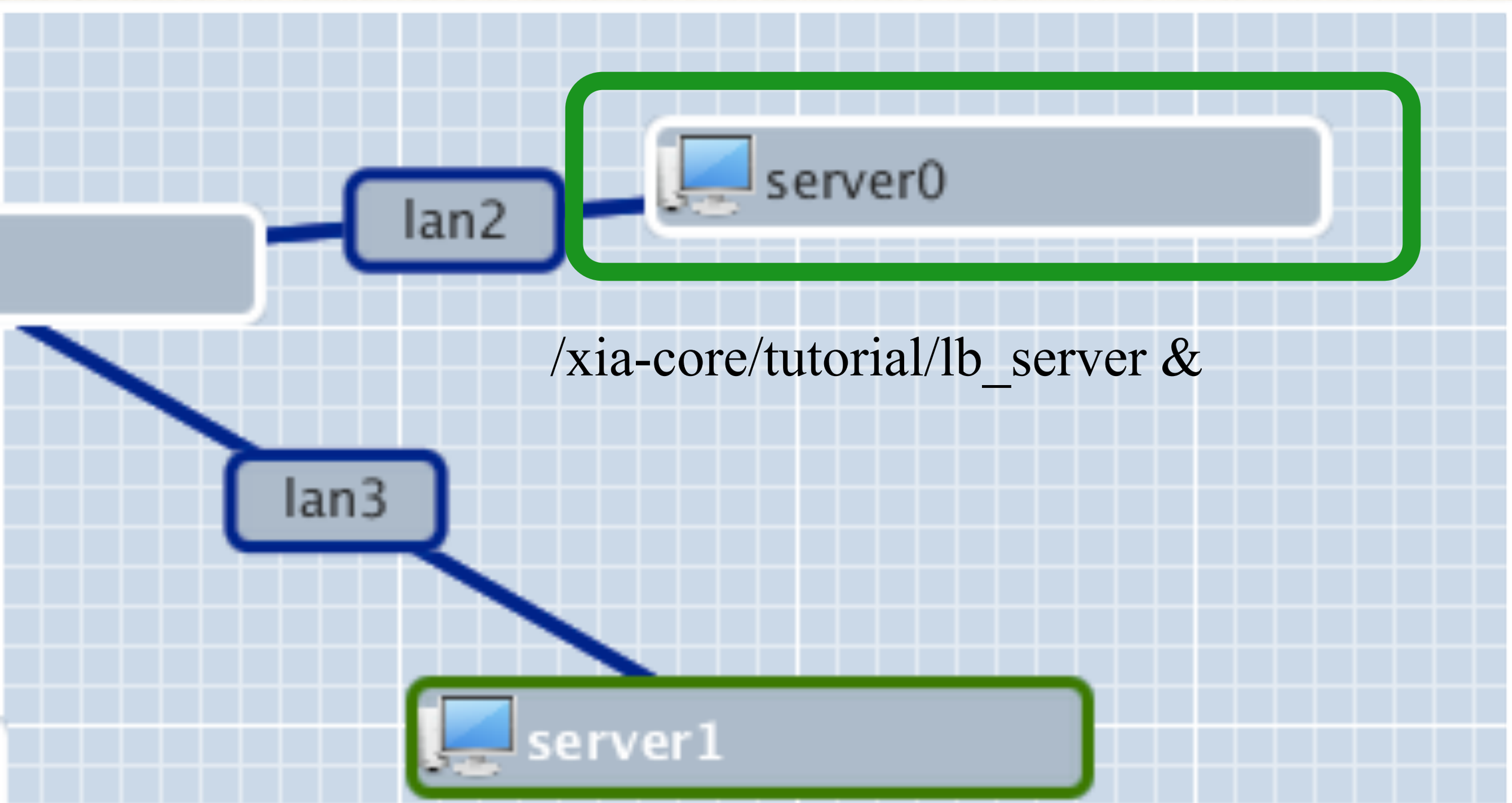
1. Bring up two service replicas 2. Add new principal type: LID

3. Add a new forwarding engine 4. Compile and reboot network 5. add routes



6. Verify the behavior of LID

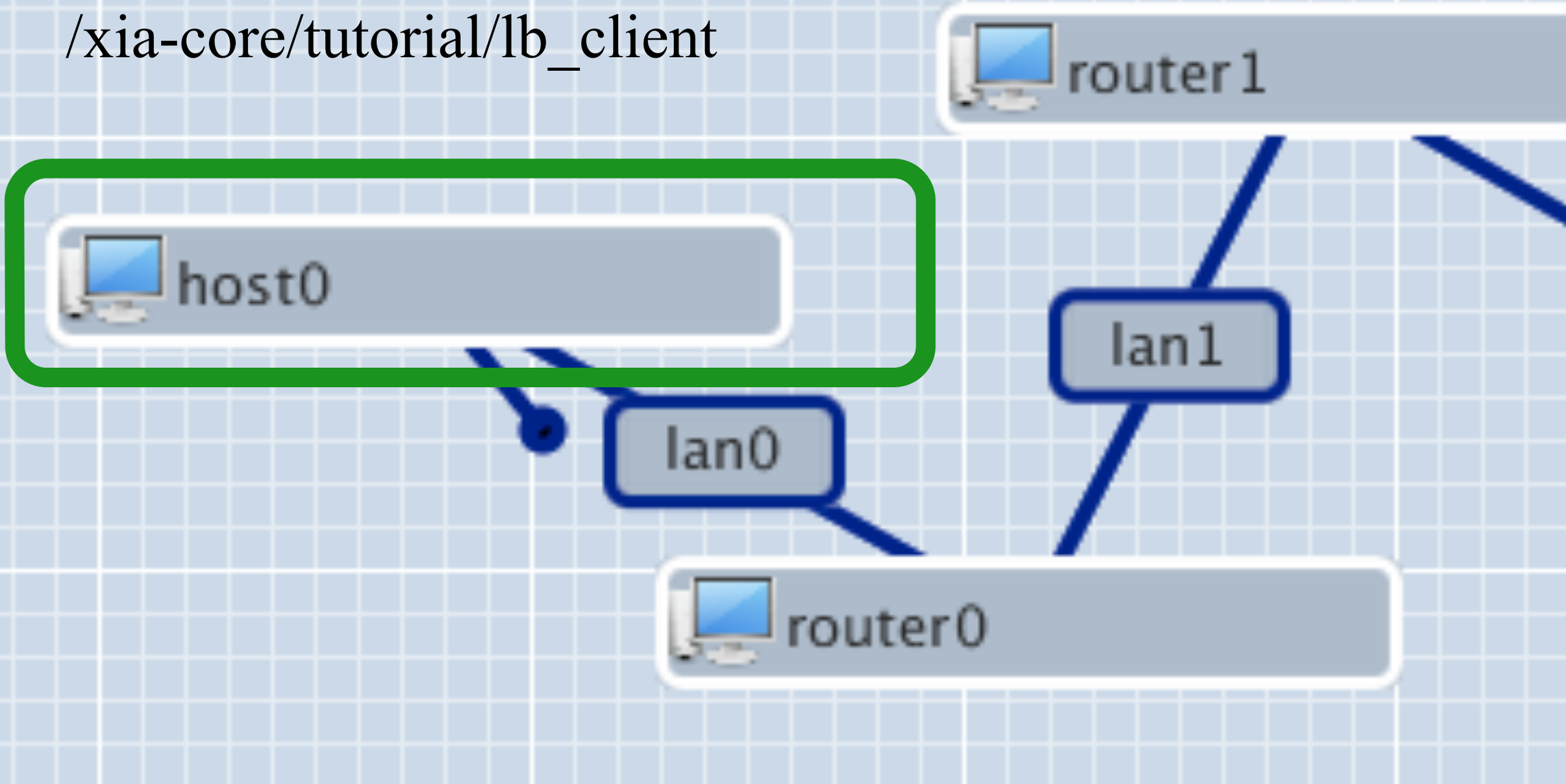
1.Start the services



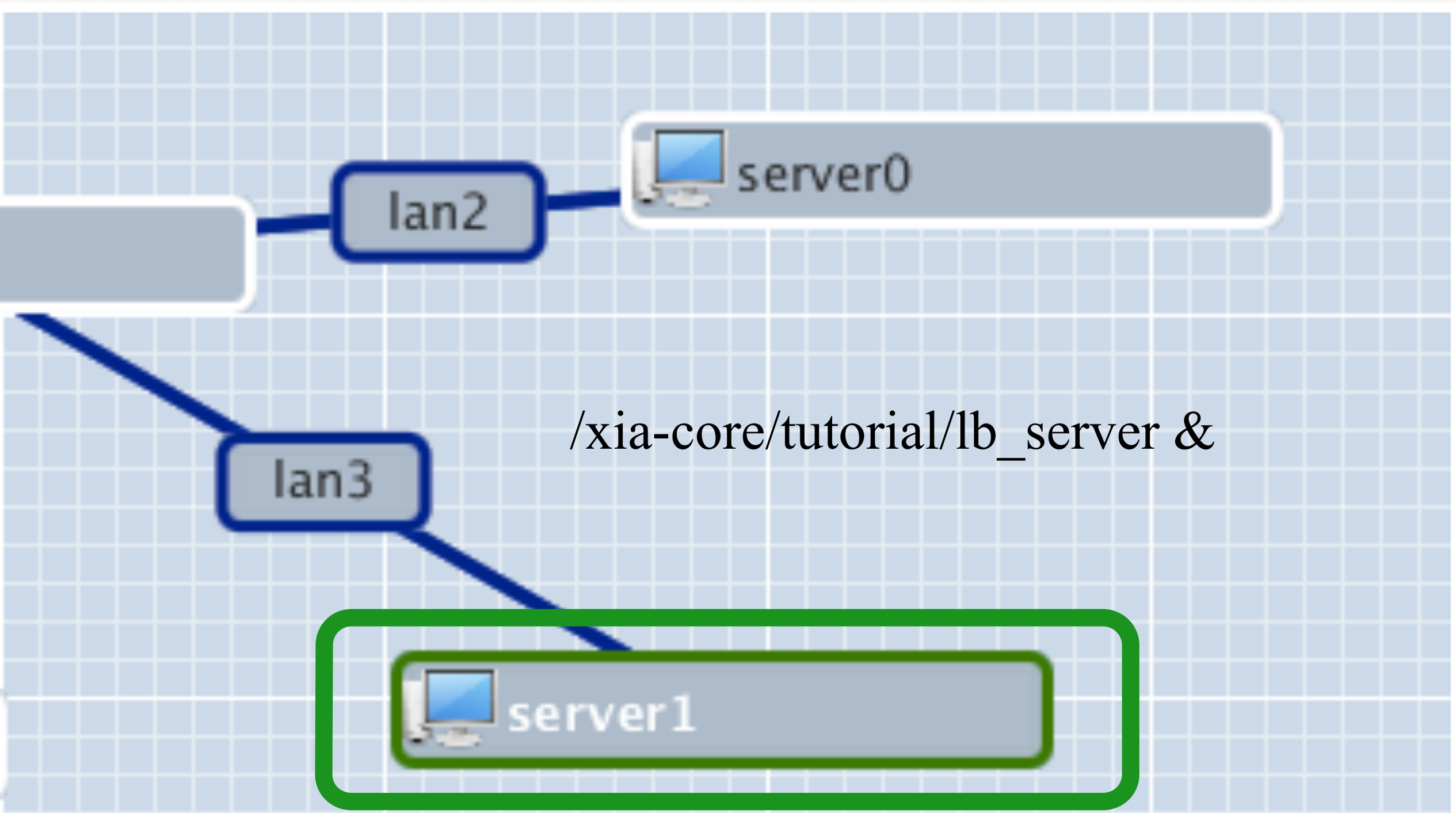
/xia-core/tutorial/lb_server &

1.Start the services

/xia-core/tutorial/lb_client



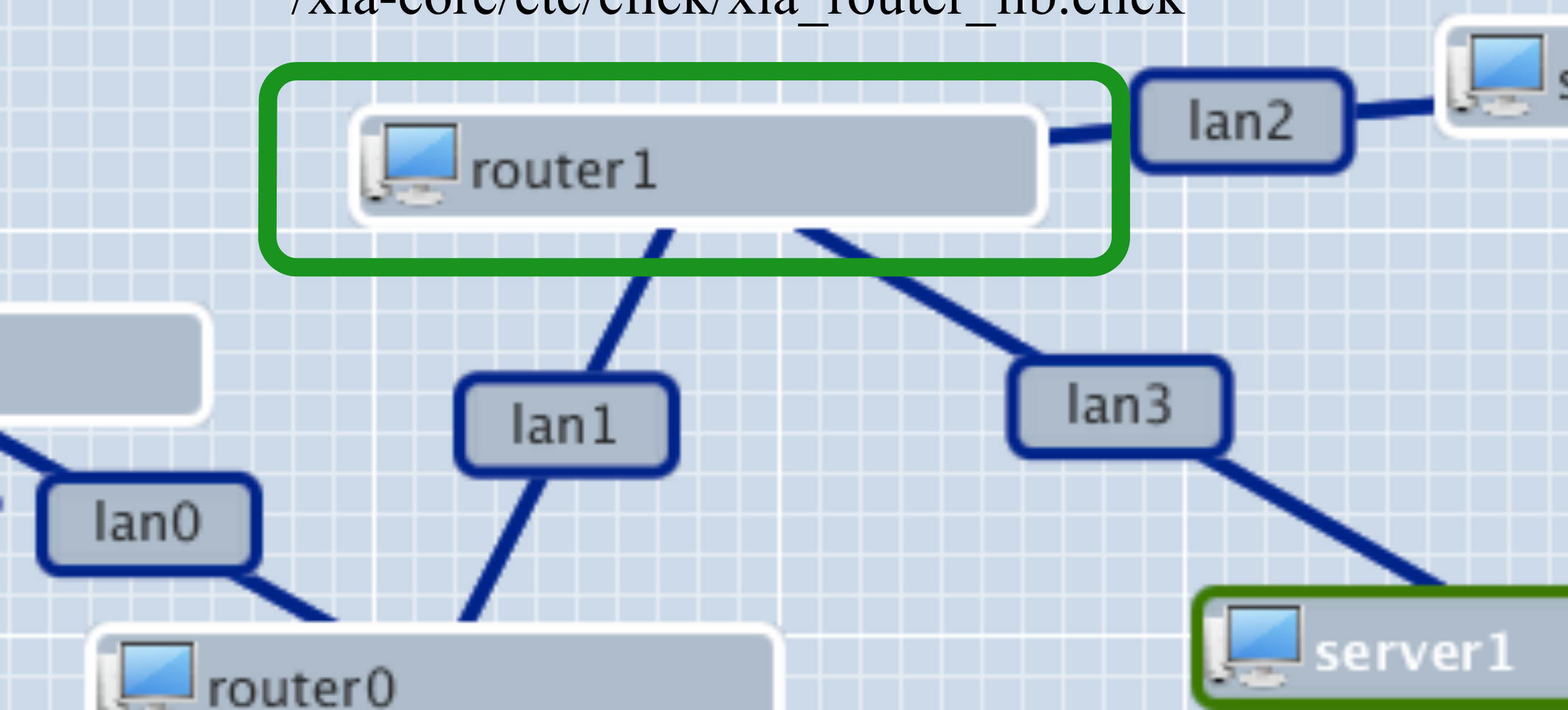
1.Start the services



/xia-core/tutorial/lb_server &

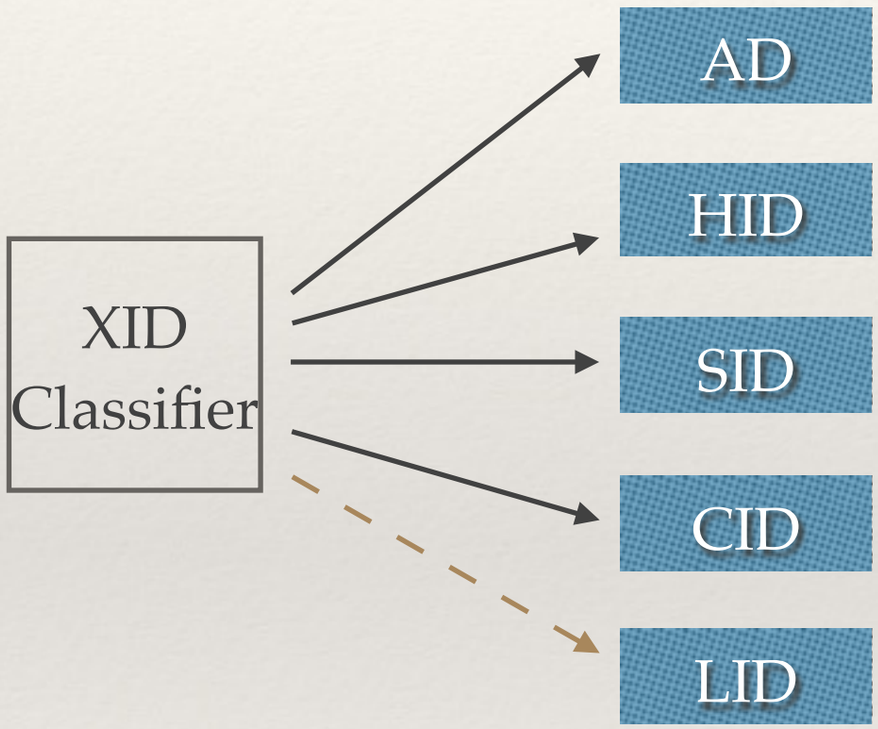
2.Add the new principal type: LID

/xia-core/etc/click/xia_router_lib.click



2.Add the new principal type: LID

/xia-core/etc/click/xia_router_lib.click

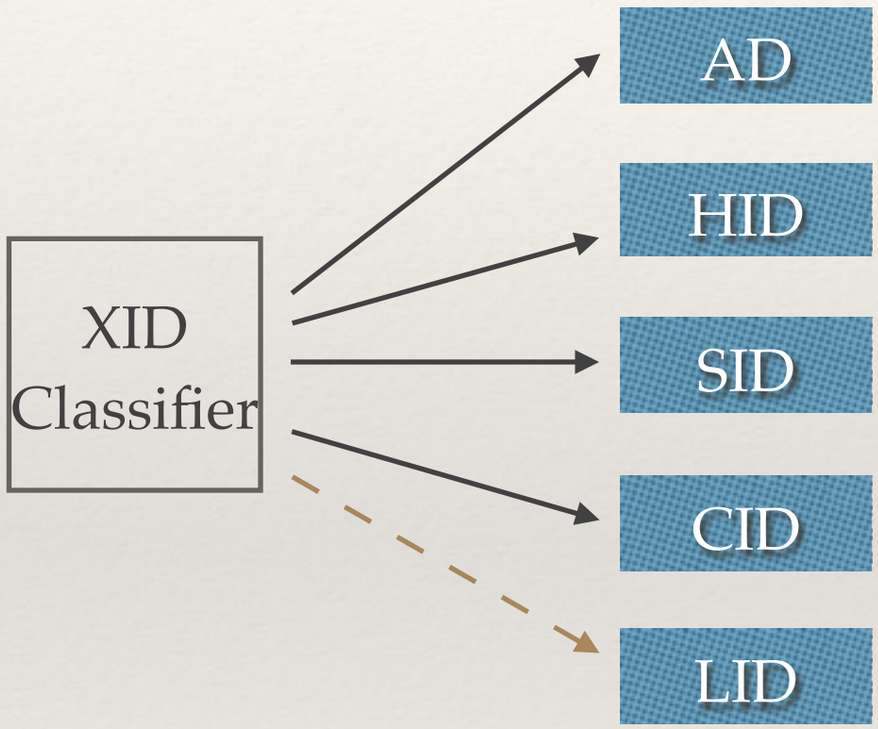


Line 44

c :: XIAXIDTypeClassifier(next AD, next HID, next SID, next CID, next IP, next LID, -);

2.Add the new principal type: LID

/xia-core/etc/click/xia_router_lib.click

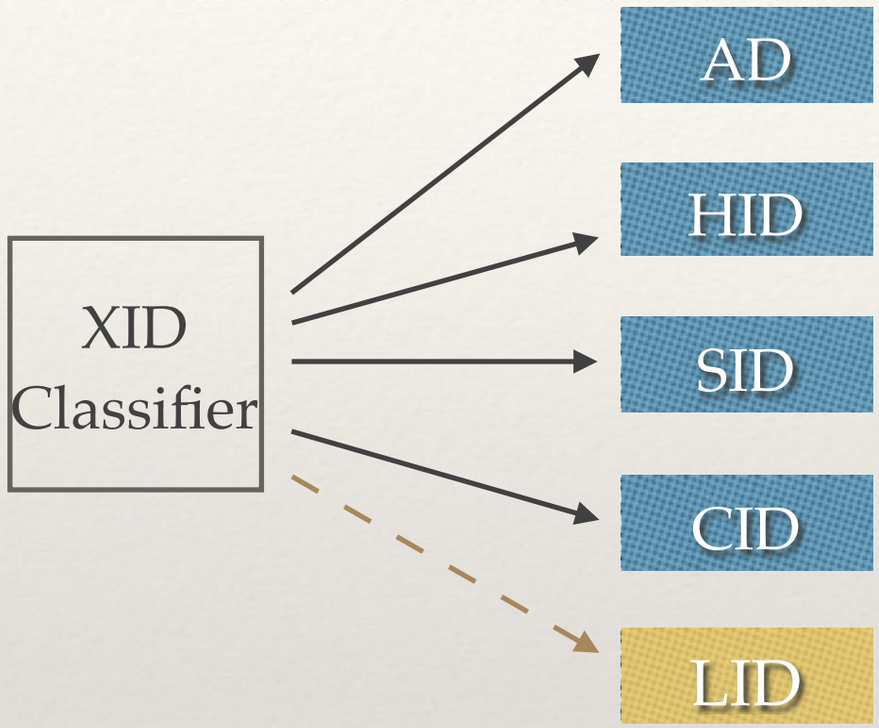


Line 44

c :: XIAXIDTypeClassifier(next AD, next HID, next SID, next CID, next IP, next LID, -);

2.Add the new principal type: LID

Line 70 /xia-core/etc/click/xia_router_lib.click



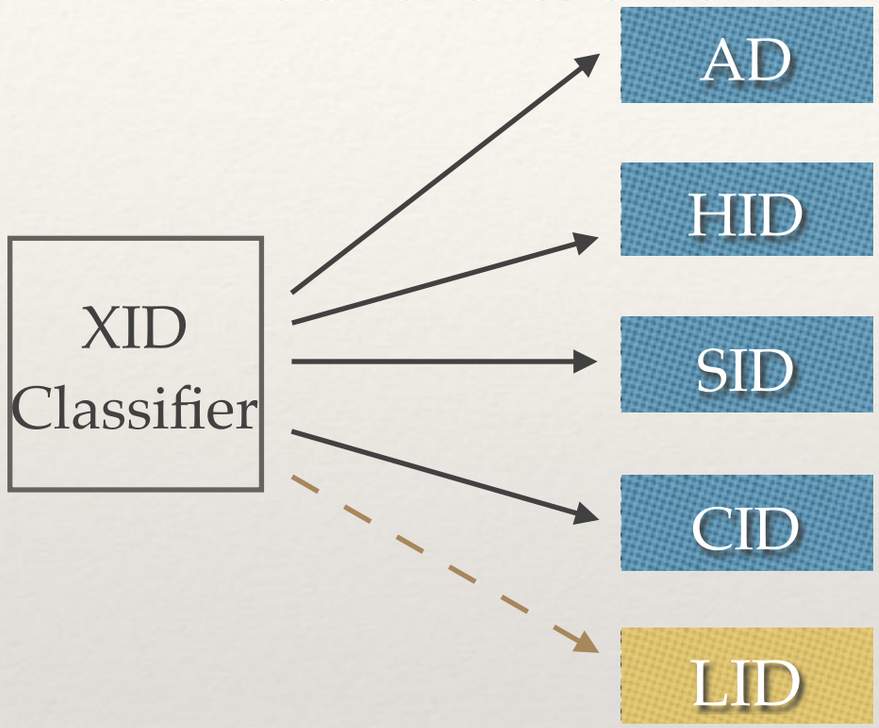
rt_AD, rt_HID, rt_SID, rt_CID, rt_IP ::
XIAXIDRouteTable($local_addr, $num_ports);

rt_LID :: XIANEWXIDRouteTable($local_addr, $num_ports);

c => rt_AD, rt_HID, rt_SID, rt_CID, rt_IP, rt_LID, [2]output;

2.Add the new principal type: LID

Line 70 /xia-core/etc/click/xia_router_lib.click



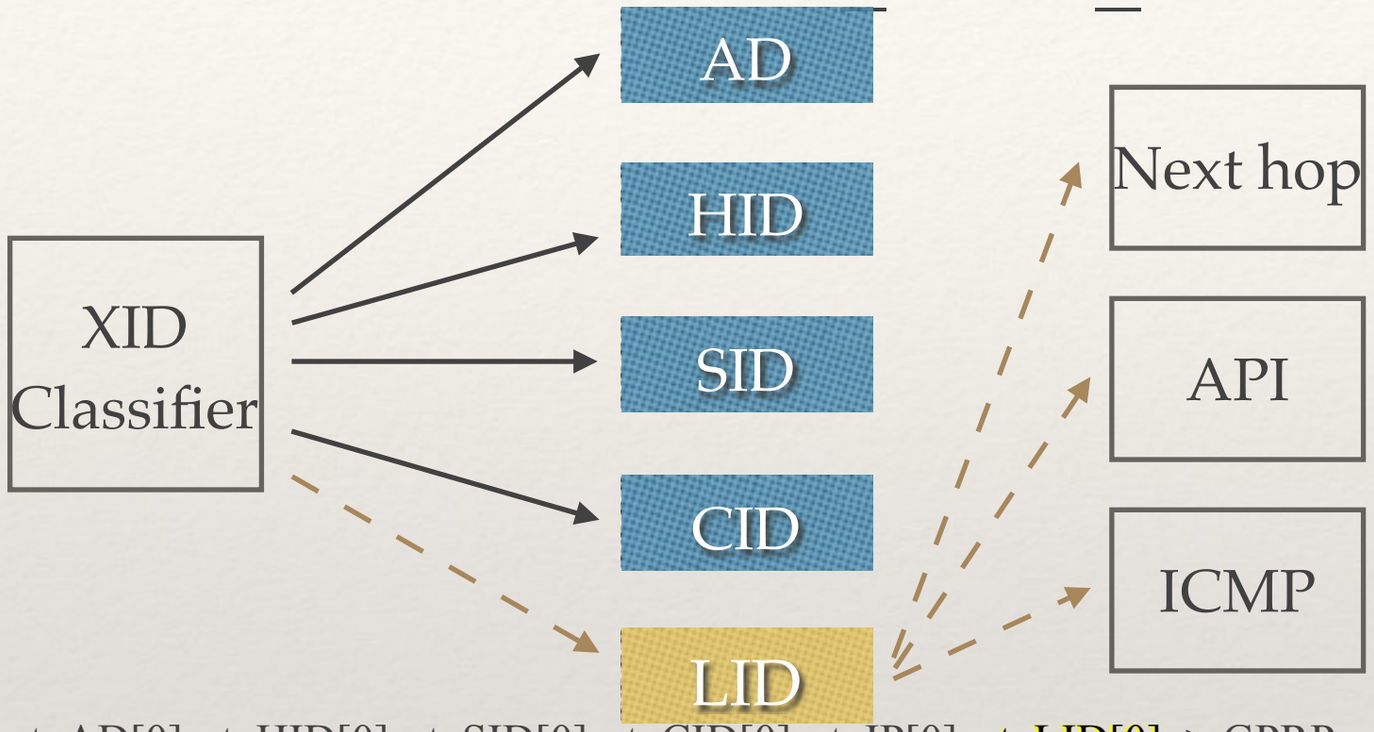
rt_AD, rt_HID, rt_SID, rt_CID, rt_IP ::
XIAXIDRouteTable($local_addr, $num_ports);

rt_LID :: XIANEWXIDRouteTable($local_addr, $num_ports);

c => rt_AD, rt_HID, rt_SID, rt_CID, rt_IP, rt_LID, [2]output;

2.Add the new principal type: LID

Line 80 /xia-core/etc/click/xia_router_lib.click



rt_AD[0], rt_HID[0], rt_SID[0], rt_CID[0], rt_IP[0], rt_LID[0] -> GPRP;

rt_AD[1], rt_HID[1], rt_LID[1], rt_CID[1], rt_IP[1] -> XIANextHop -> check_dest;

rt_SID[1] -> XIANextHop -> XIAPaint($DESTINED_FOR_LOCALHOST) -> [1]output;

rt_AD[2], rt_HID[2], rt_SID[2], rt_CID[2], rt_IP[2], rt_LID[2] -> consider_next_path;

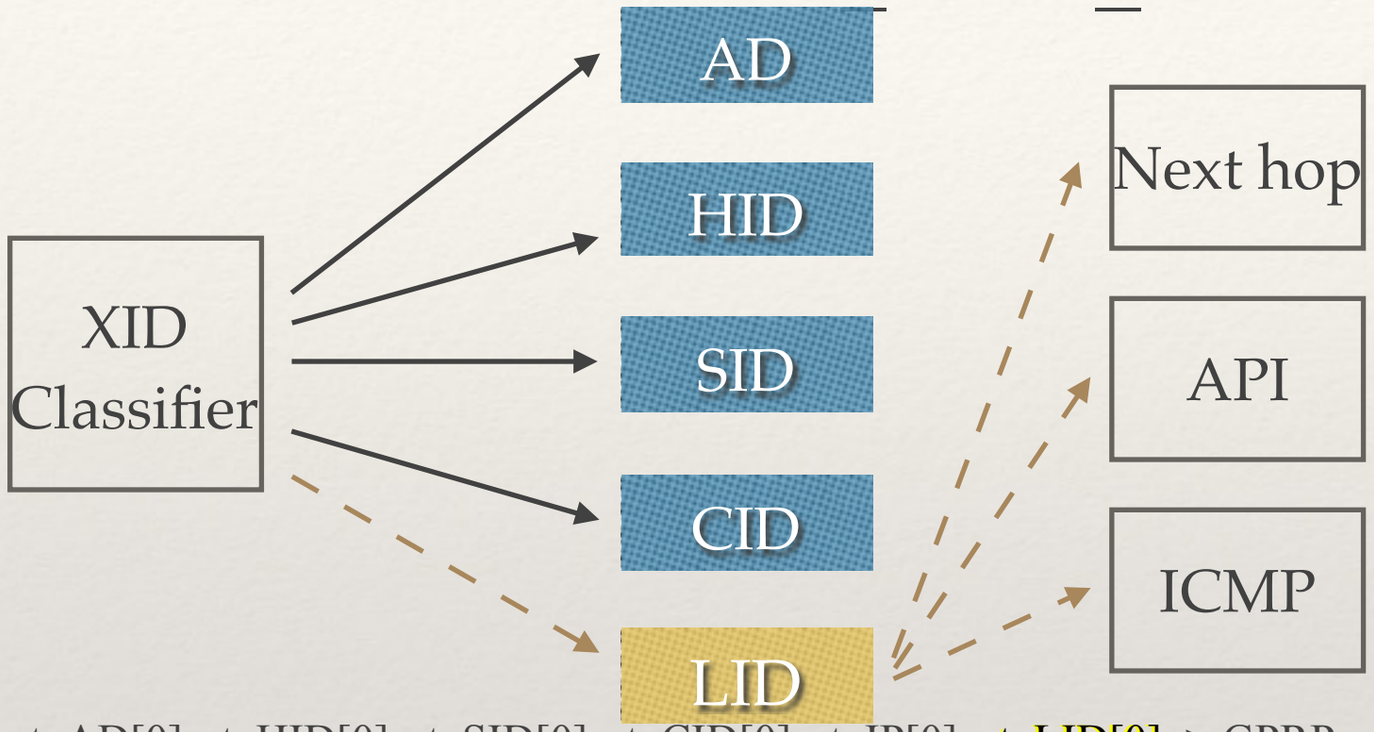
rt_AD[3], rt_HID[3], rt_LID[3], rt_CID[3], rt_IP[3] -> Discard;

rt_SID[3] -> [3]output;

rt_AD[4], rt_HID[4], rt_SID[4], rt_CID[4], rt_IP[4], rt_LID[4] -> x; // xcmp redirect message

2.Add the new principal type: LID

Line 80 /xia-core/etc/click/xia_router_lib.click



rt_AD[0], rt_HID[0], rt_SID[0], rt_CID[0], rt_IP[0], rt_LID[0] -> GPRP;

rt_AD[1], rt_HID[1], rt_LID[1], rt_CID[1], rt_IP[1] -> XIANextHop -> check_dest;

rt_SID[1] -> XIANextHop -> XIAPaint($DESTINED_FOR_LOCALHOST) -> [1]output;

rt_AD[2], rt_HID[2], rt_SID[2], rt_CID[2], rt_IP[2], rt_LID[2] -> consider_next_path;

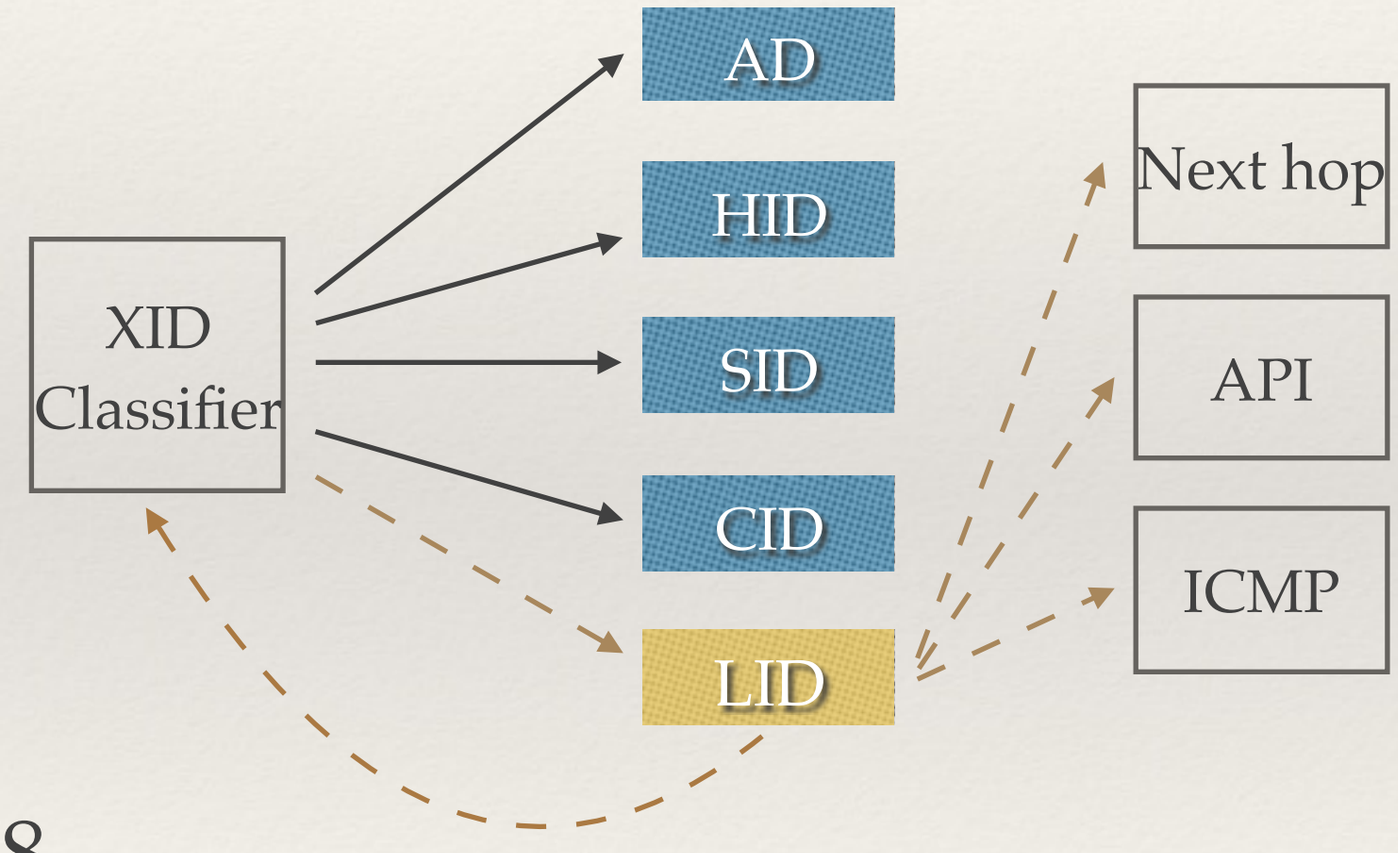
rt_AD[3], rt_HID[3], rt_LID[3], rt_CID[3], rt_IP[3] -> Discard;

rt_SID[3] -> [3]output;

rt_AD[4], rt_HID[4], rt_SID[4], rt_CID[4], rt_IP[4], rt_LID[4] -> x; // xcmp redirect message

2.Add the new principal type: LID

/xia-core/etc/click/xia_router_lib.click

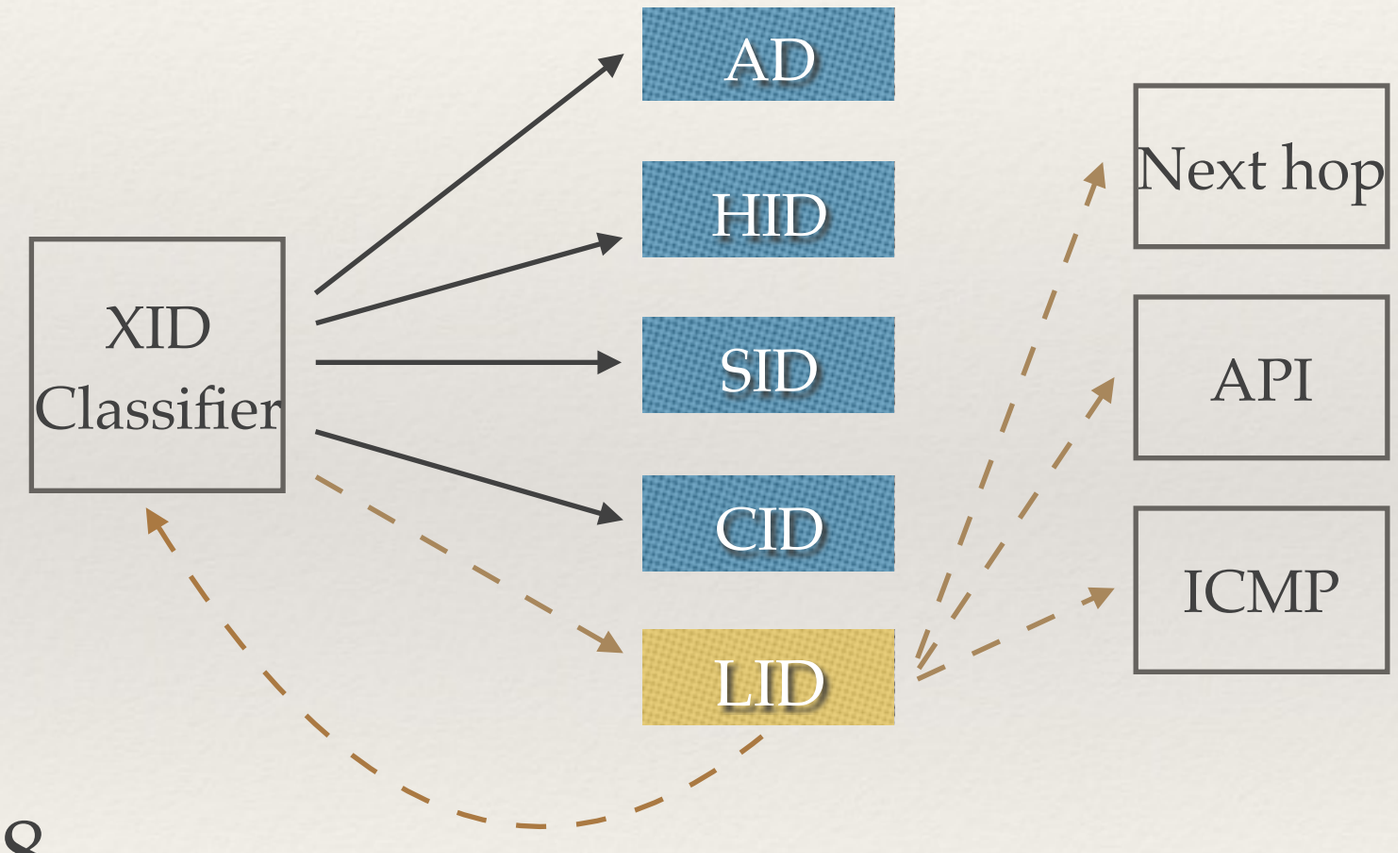


Line 278

Script(write n/proc/rt_LID.add - $FALLBACK);

2.Add the new principal type: LID

/xia-core/etc/click/xia_router_lib.click



Line 278

Script(write n/proc/rt_LID.add - $FALLBACK);

2.Add the new principal type: LID

Save the file, come back to shell

```
echo "0x50 LID" > /xia-core/etc/xids
```

2.Add the new principal type: LID

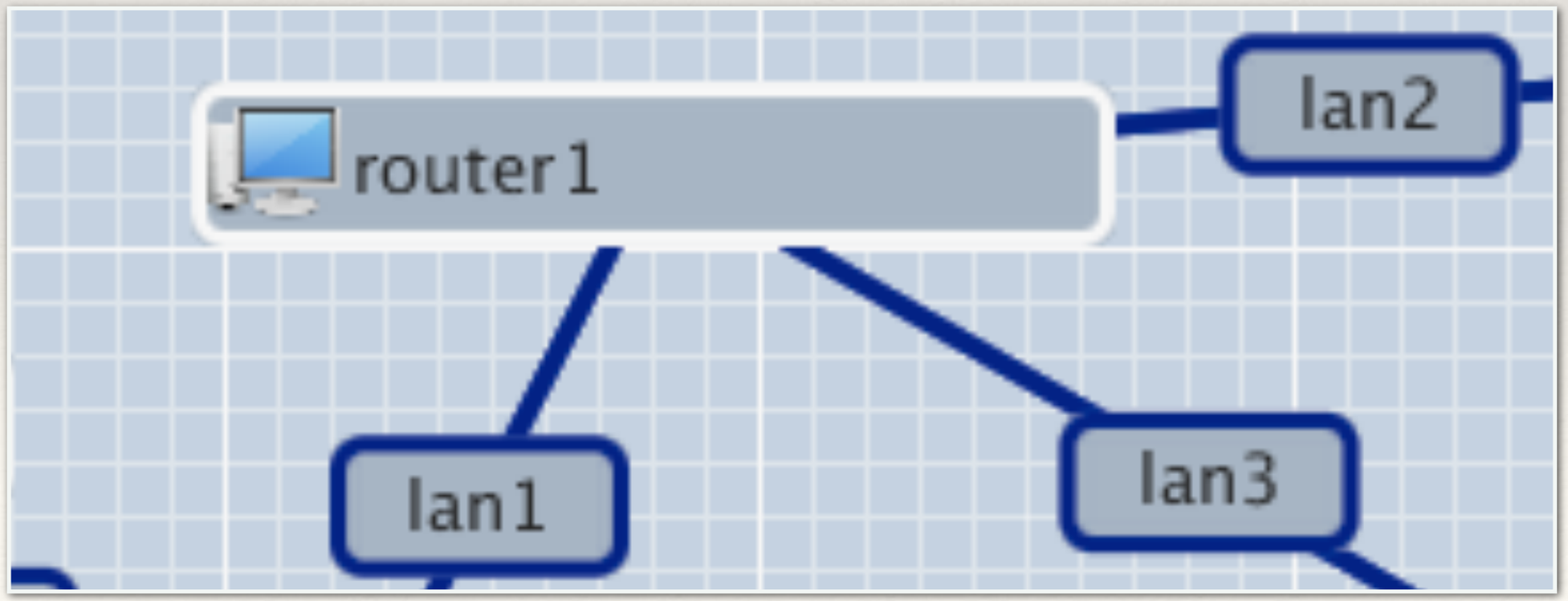
Save the file, come back to shell

```
echo "0x50 LID" > /xia-core/etc/xids
```

And a new principal type is born.

3. Add a new forwarding engine

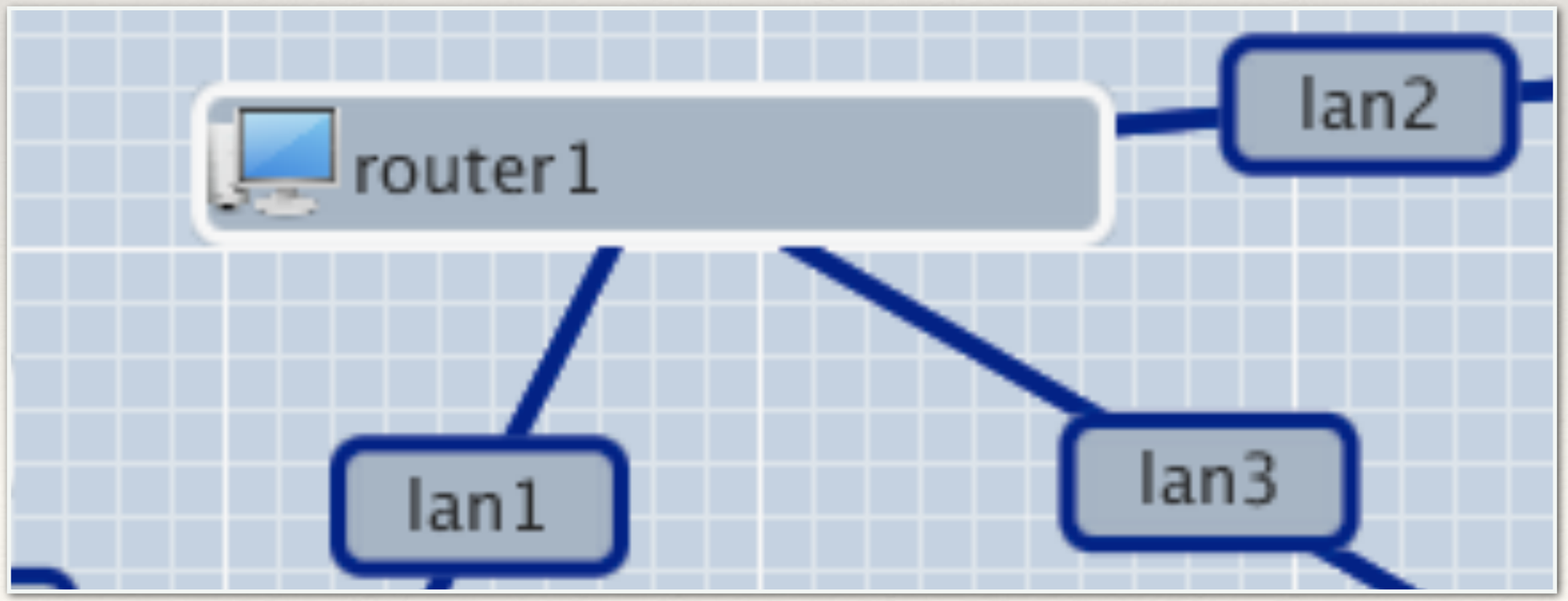
/xia-core/click/elements/xia/xianewxidroutetable.cc



3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

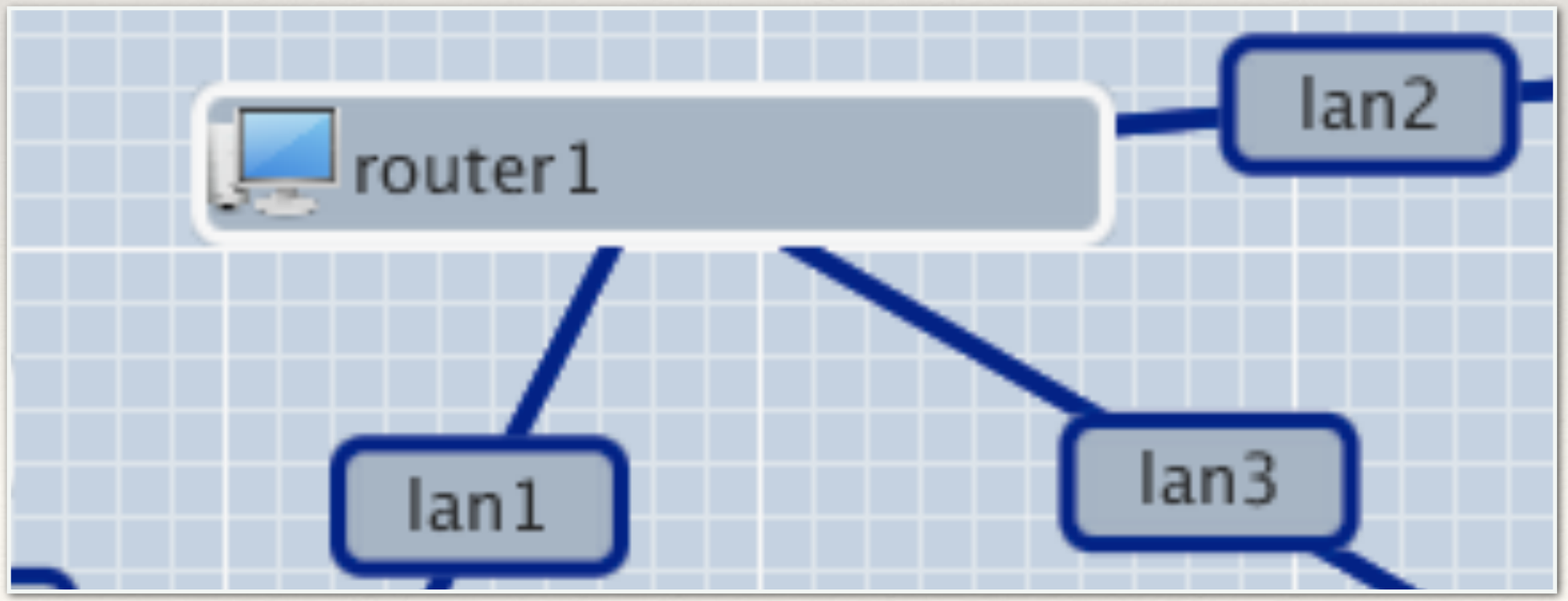
LID →



3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

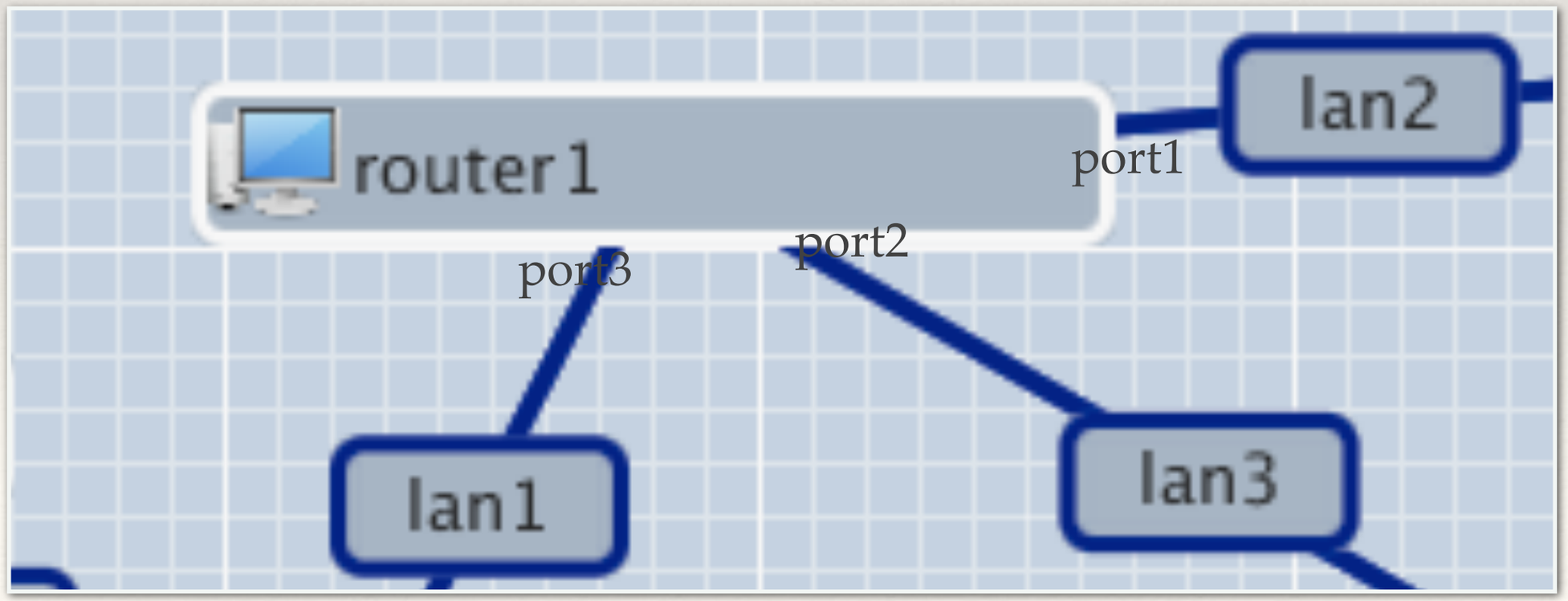
LID → Next hop(port)



3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

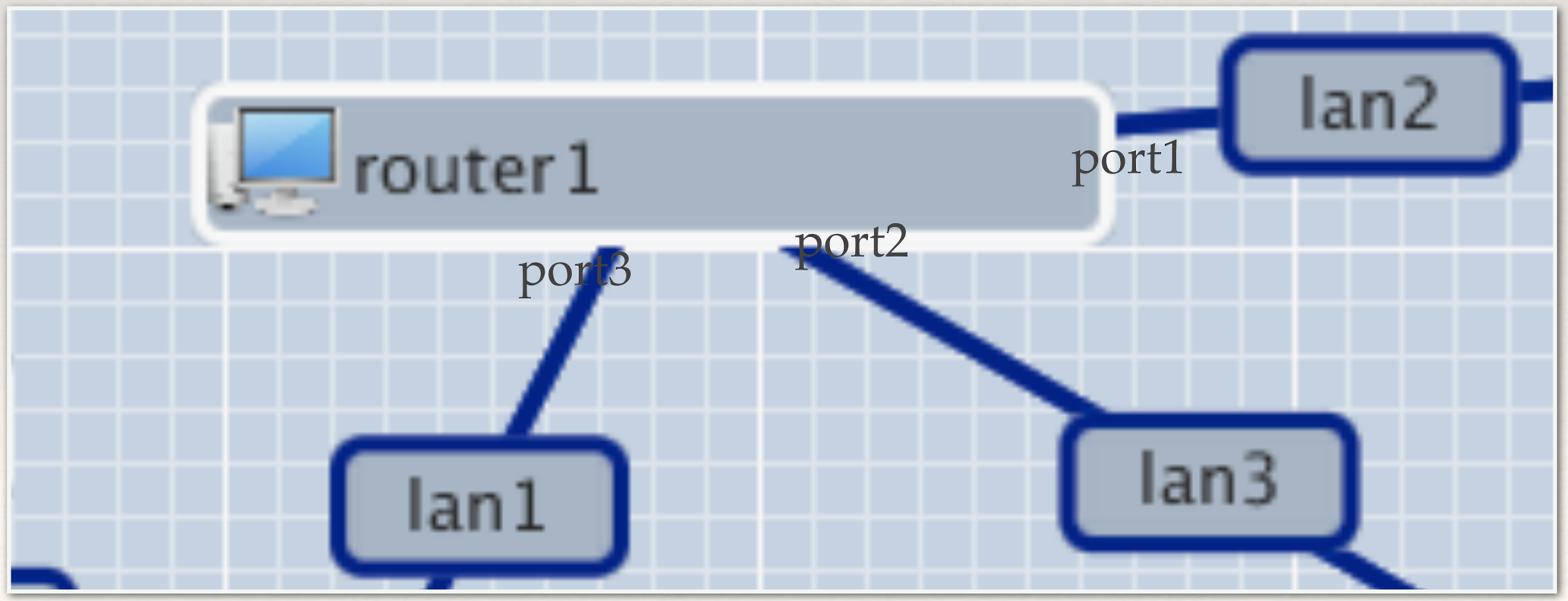
LID → Next hop(port)



3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

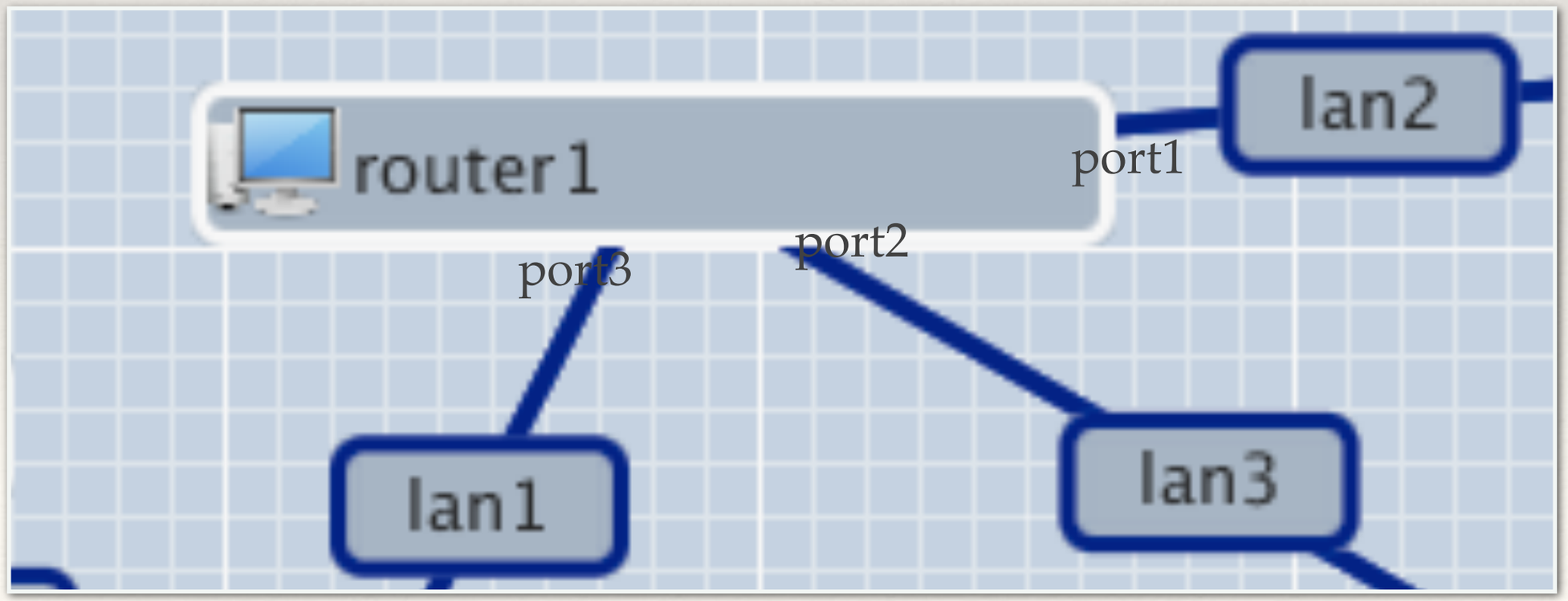
LID →



3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID → NH(port1) | NH(port2)

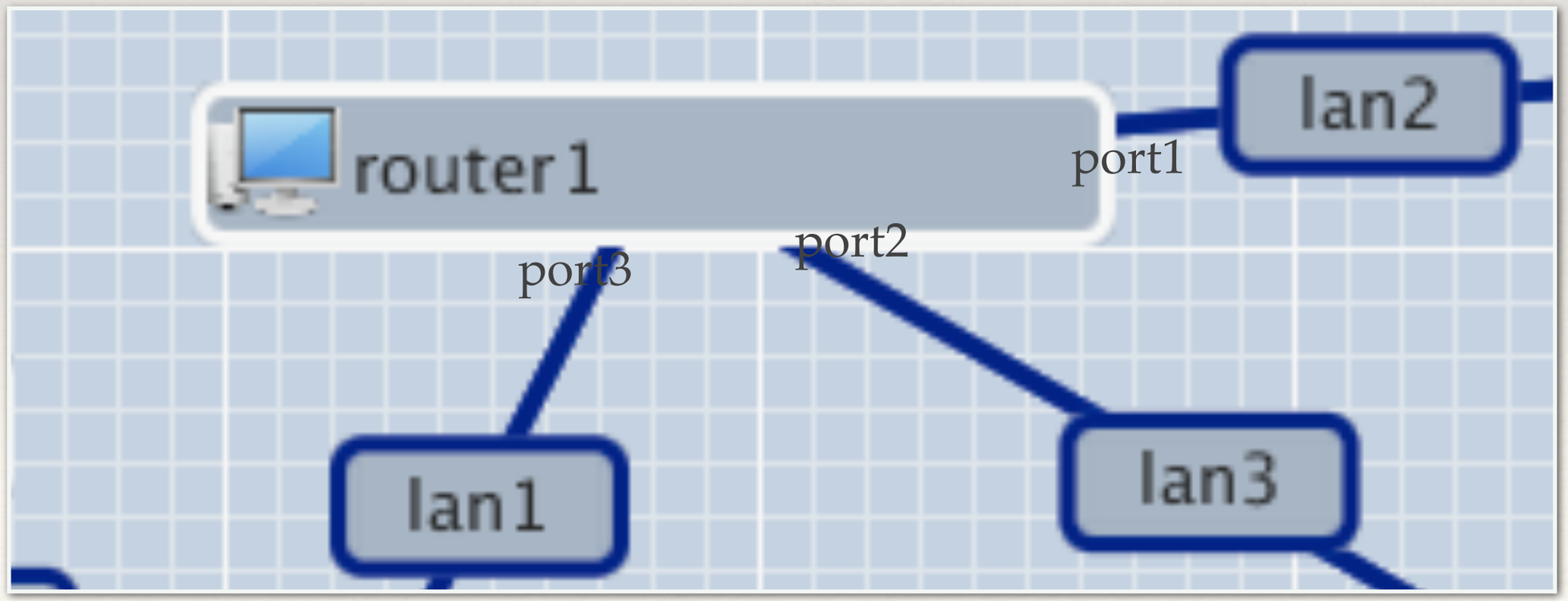


3.Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID → NH(port1) | NH(port2)

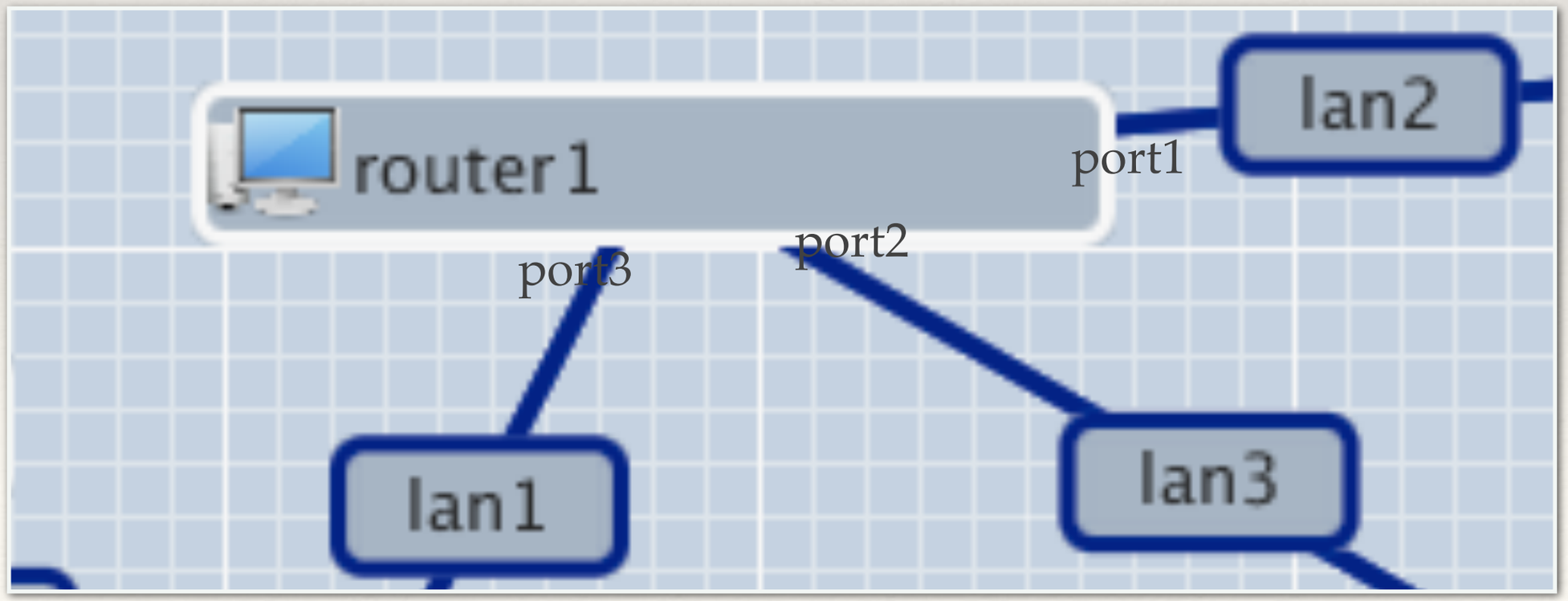
int port



3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID → NH(port1) | NH(port2)



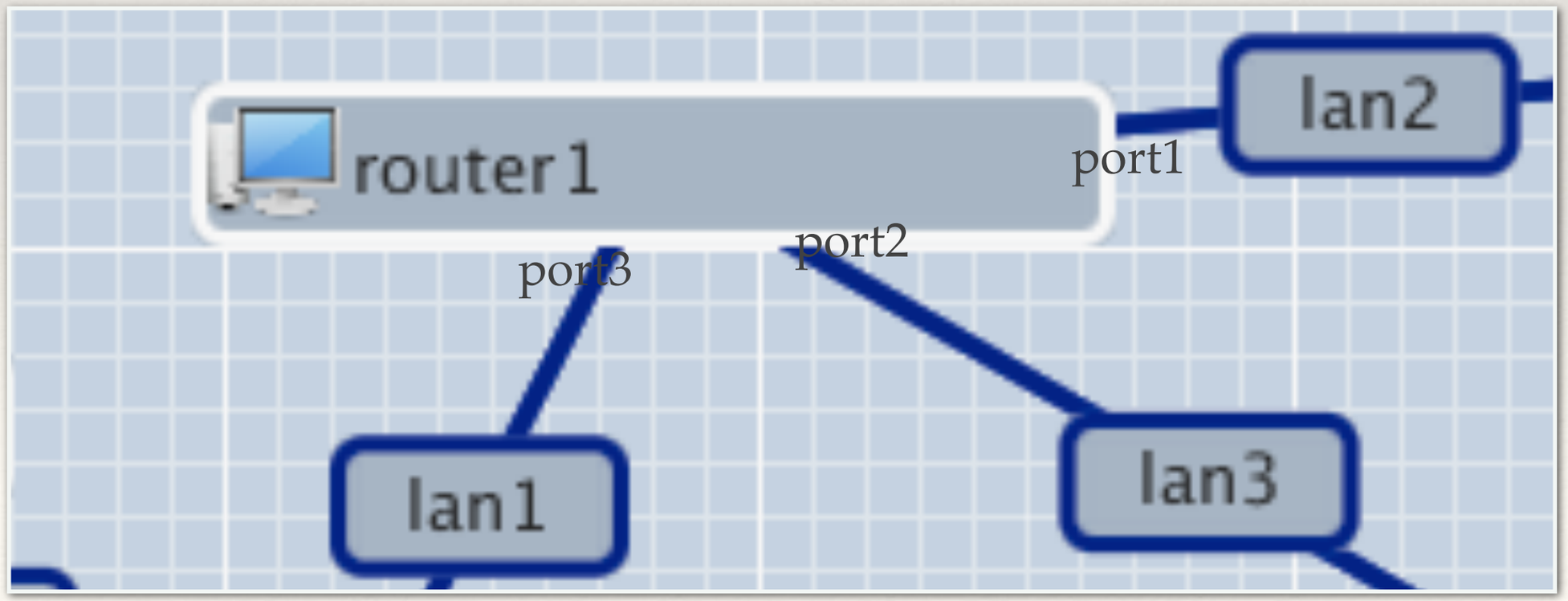
3.Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID → NH(port1) | NH(port2)

int port

Control bit



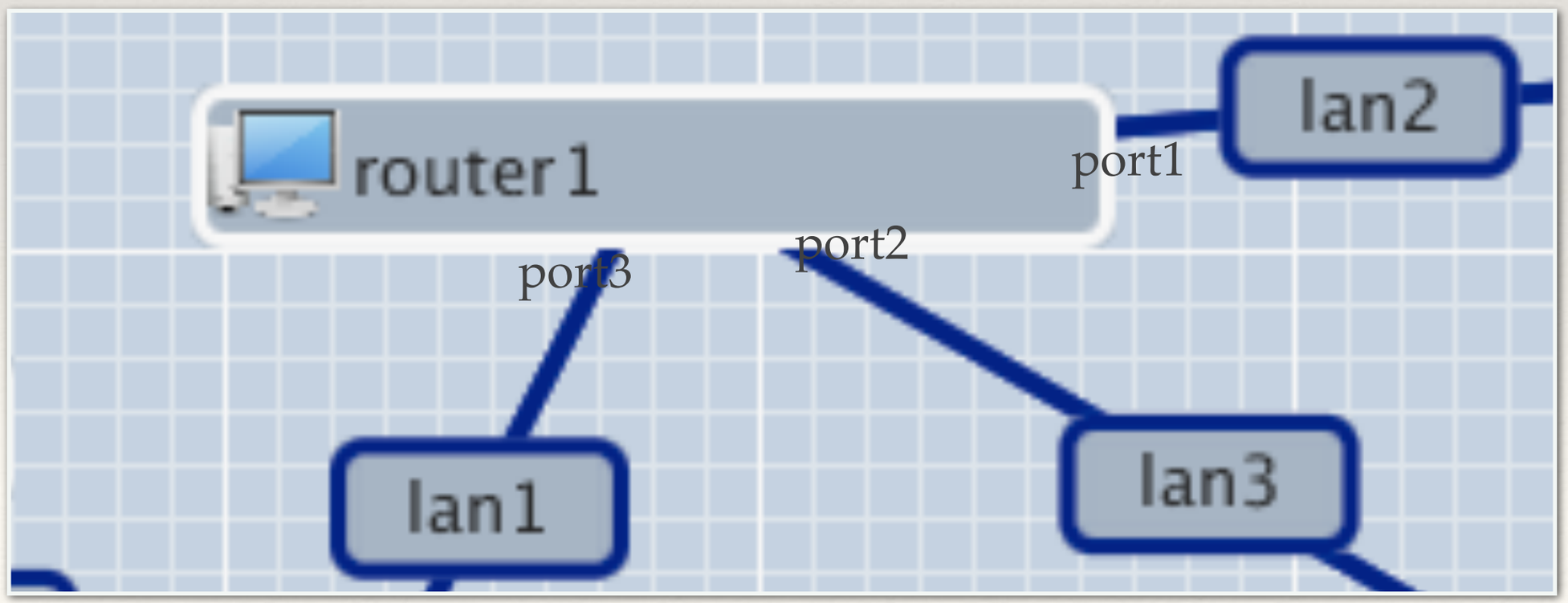
3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID →

int port

Control bit



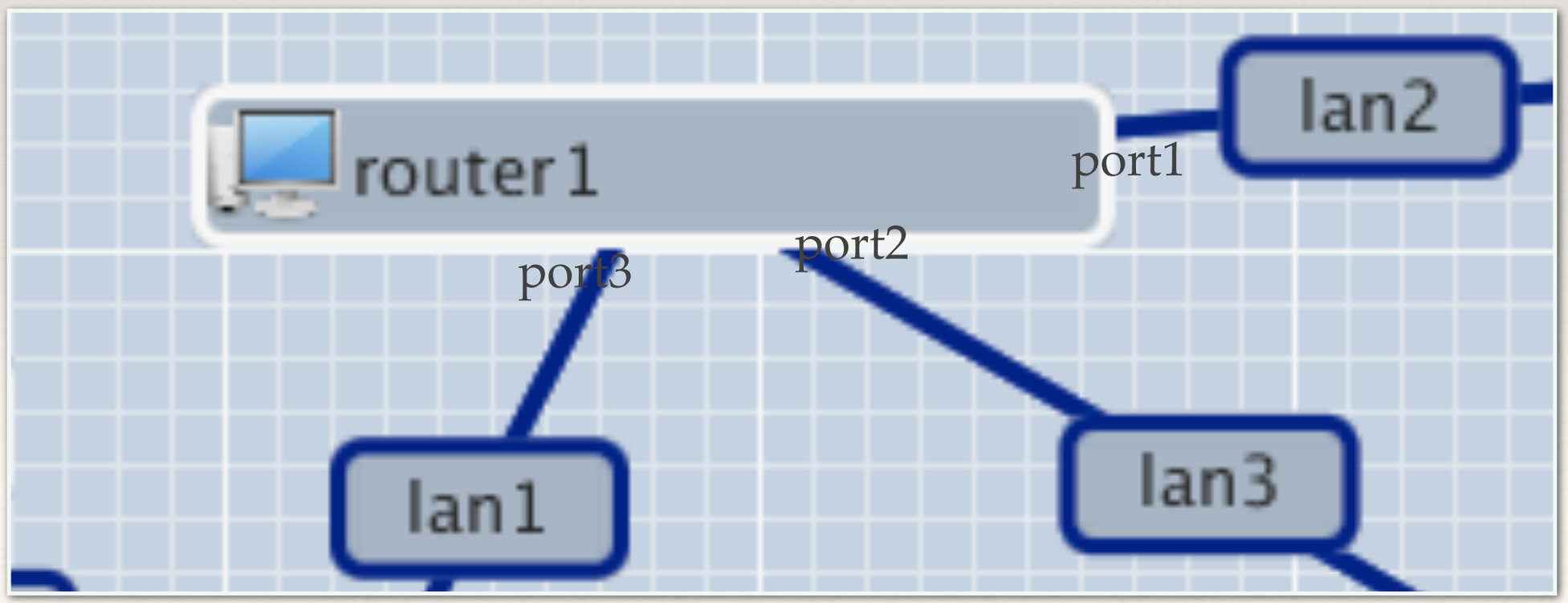
3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID →

int port

Control bit



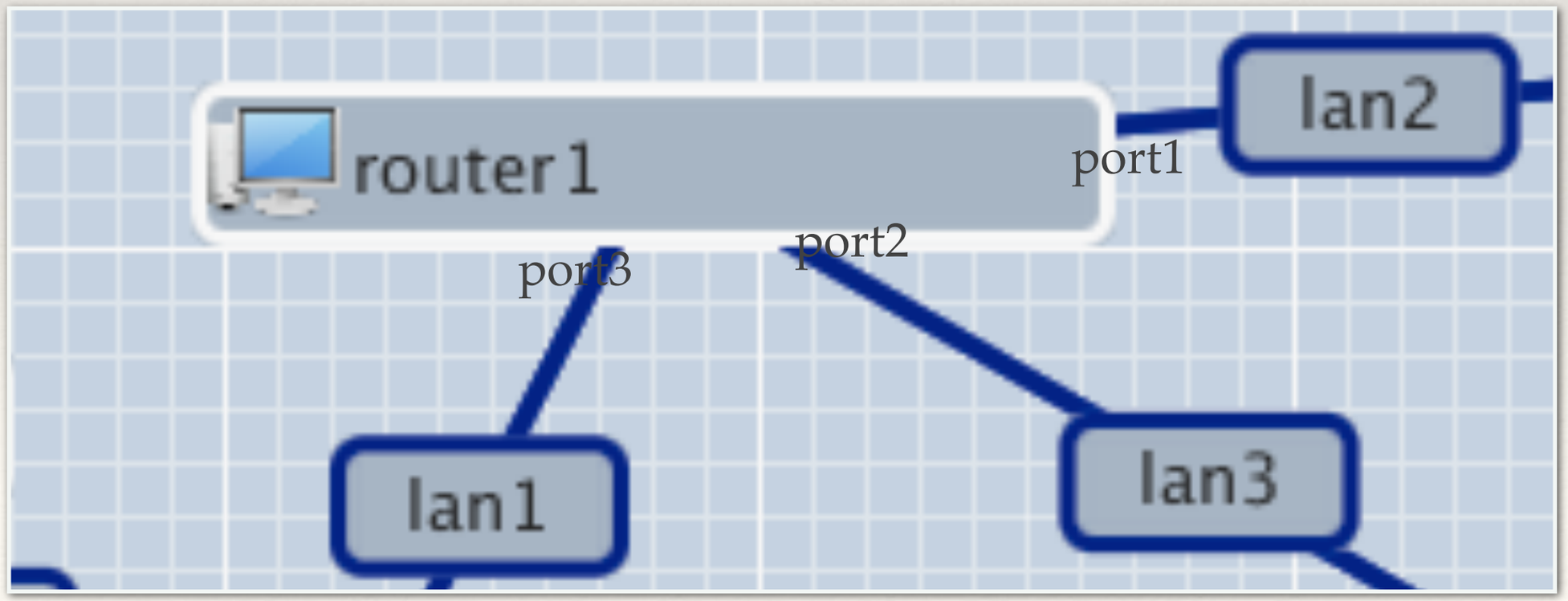
3. Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID →

Control bit

Indicator bit



3.Add a new forwarding engine

/xia-core/click/elements/xia/xianewxidroutetable.cc

LID ->

int port:	00000000	00000001	00000010	00000001
	Control=0	Indicator=1	port2=2	port1=1
	physical ports	use both ports		

line 582

IF port < (1<<16) THEN only port1 is used
ELSE both port1 and port2 are used, randomly choose one from them

4. Compile and Restart XIA

We are still on router1

```
cd /xia-core
make
tutorial/xnode restart
```

5. Add the New Route

We are still on router1

As we have no time to implement a routing protocol, we just manually add a route of LID

int port= 00000000 00000001 00000001 00000000

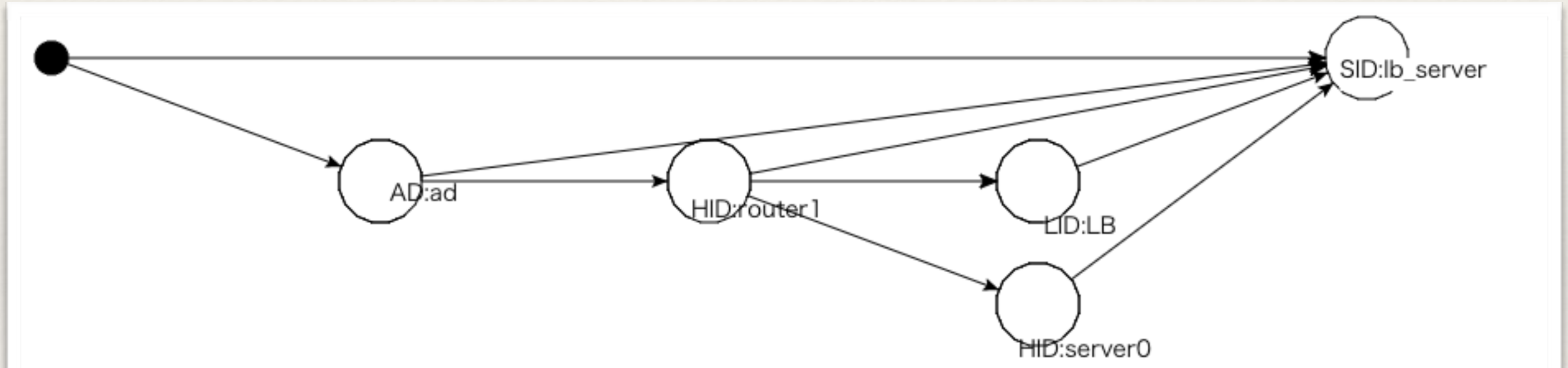
= 65792 Indicator port2=1 port1=0

/xia-core/bin/xroute -a router1,LID,LID:
0e0000000000000000000000000000009876543210,65792

5. Using the New Route

We are still on router1

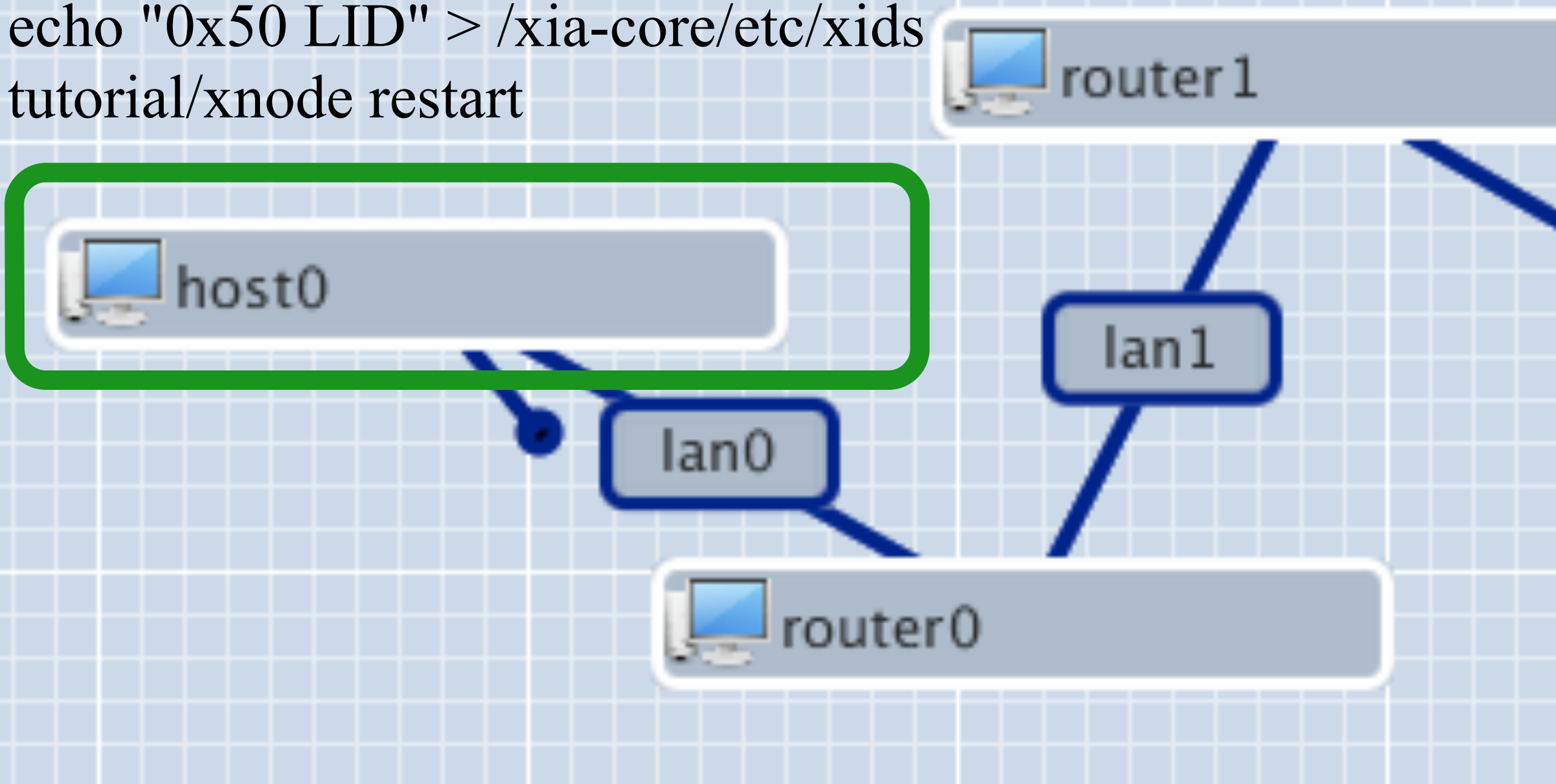
Next, draw a DAG to tell network how to use LID.
That DAG need to attach to the service name lb.tutorial.xia on the name server



/xia-core/tutorial/addname lb.tutorial.xia "DAG 4 0 - AD:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb 4 2 - LID: 0e000000000000000000000009876543210 4 - HID: 6666666666666666666666666666666666666666 4 1 3 - HID: 333333333333333333333333333333333333 4 - SID: 0f0000000000000000000000000123456789"

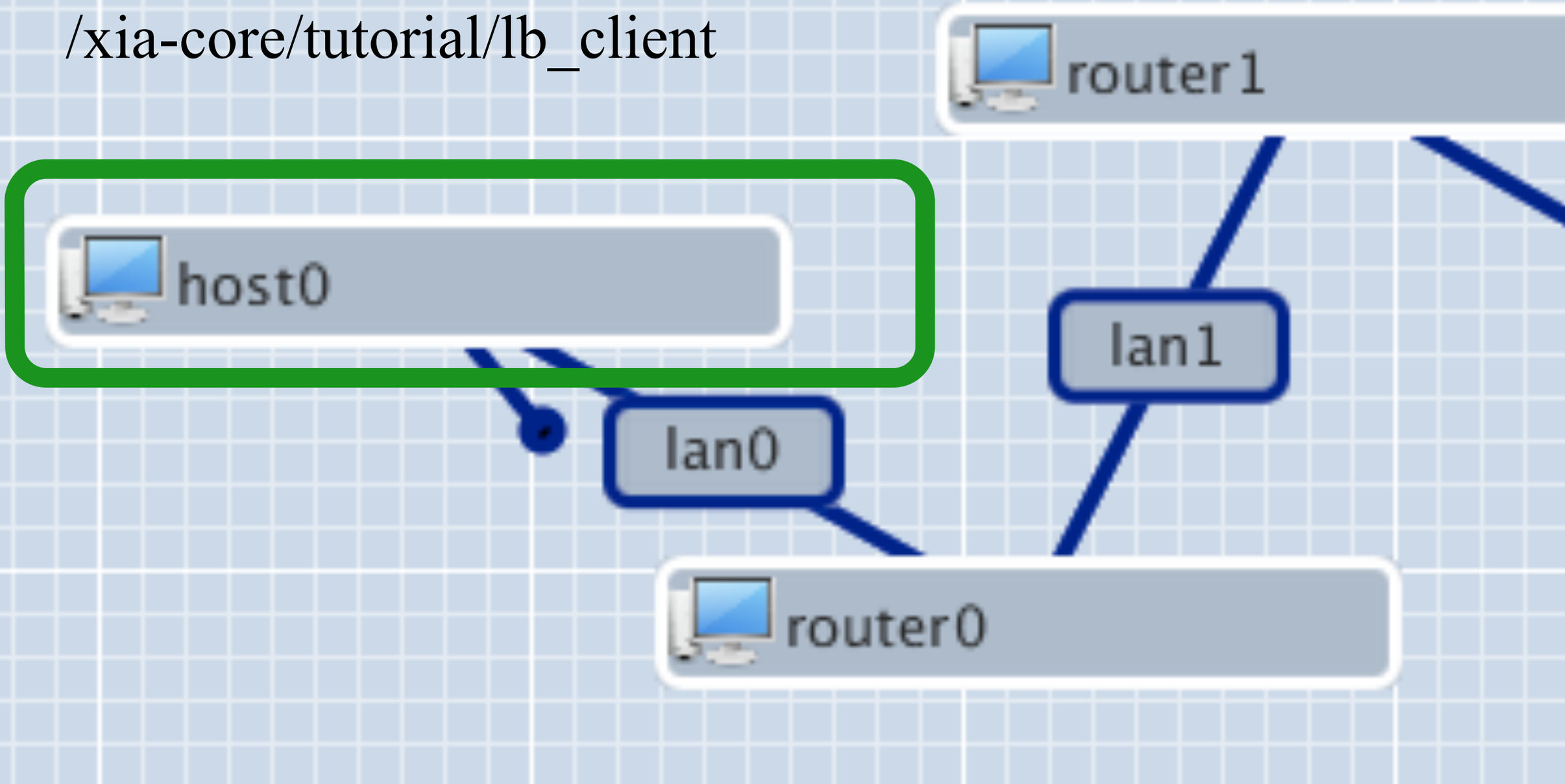
6. Verify the behavior of LID

cd /xia-core
echo "0x50 LID" > /xia-core/etc/xids
tutorial/xnode restart

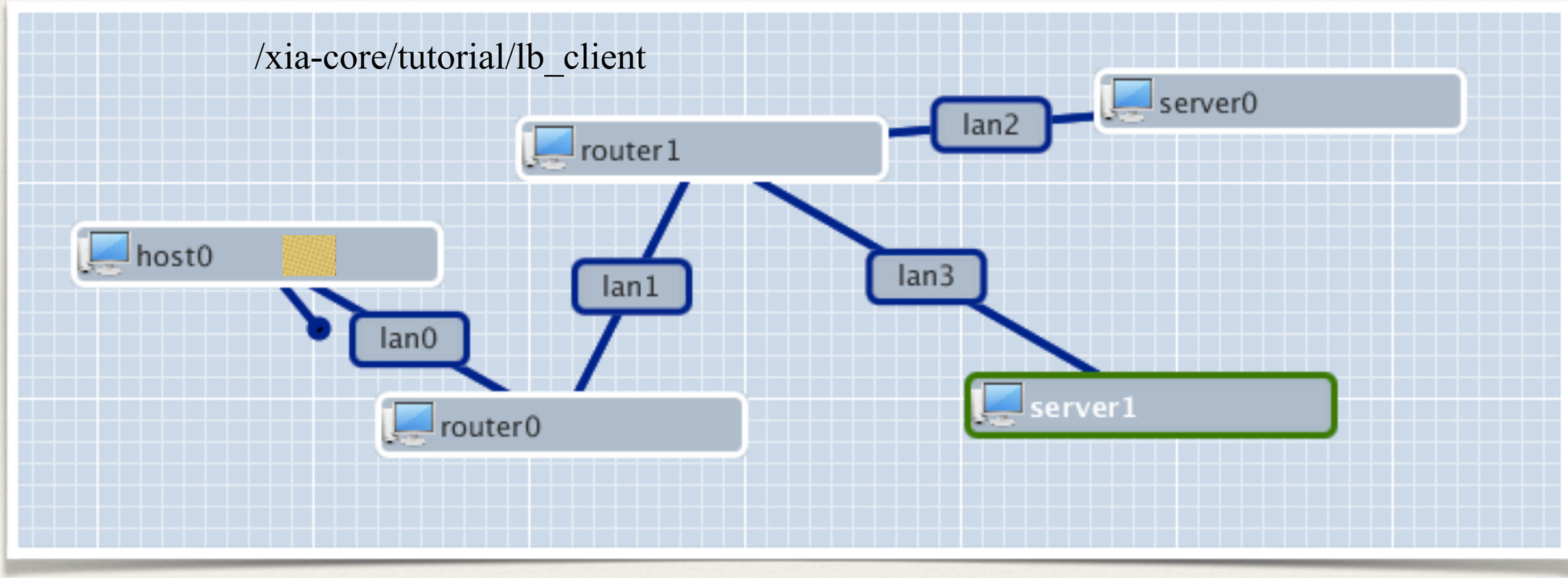


6. Verify the behavior of LID

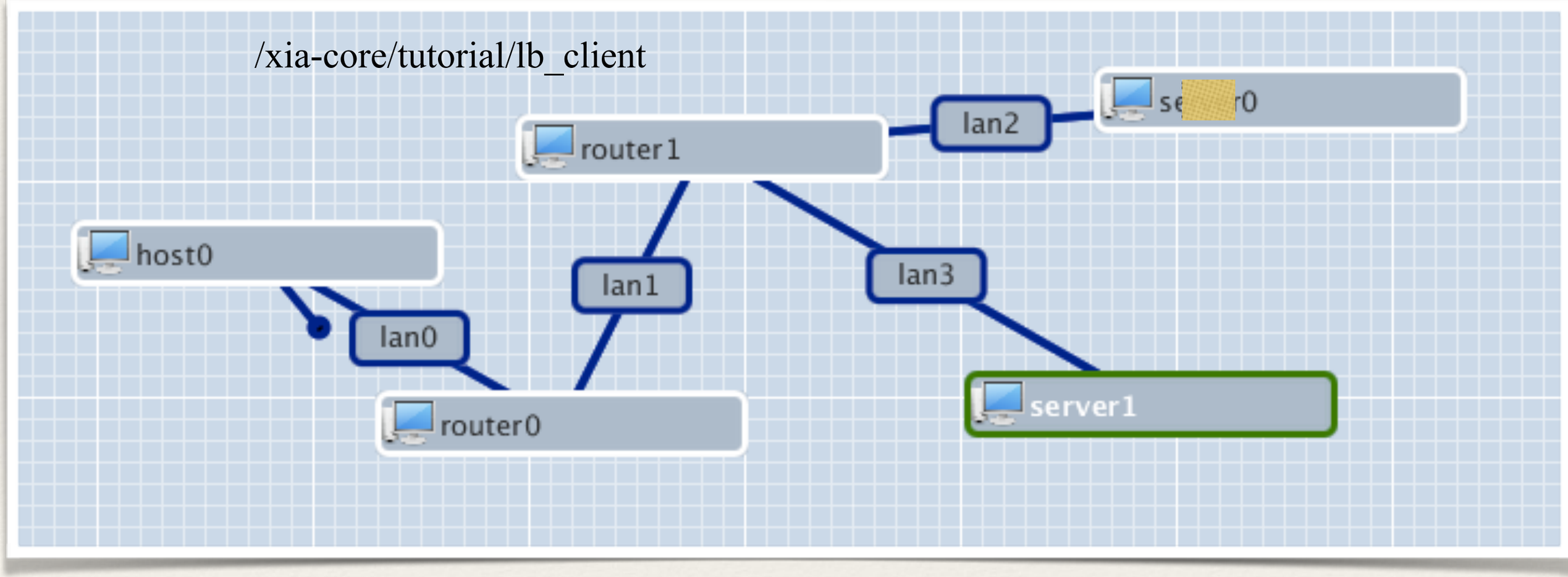
/xia-core/tutorial/lb_client



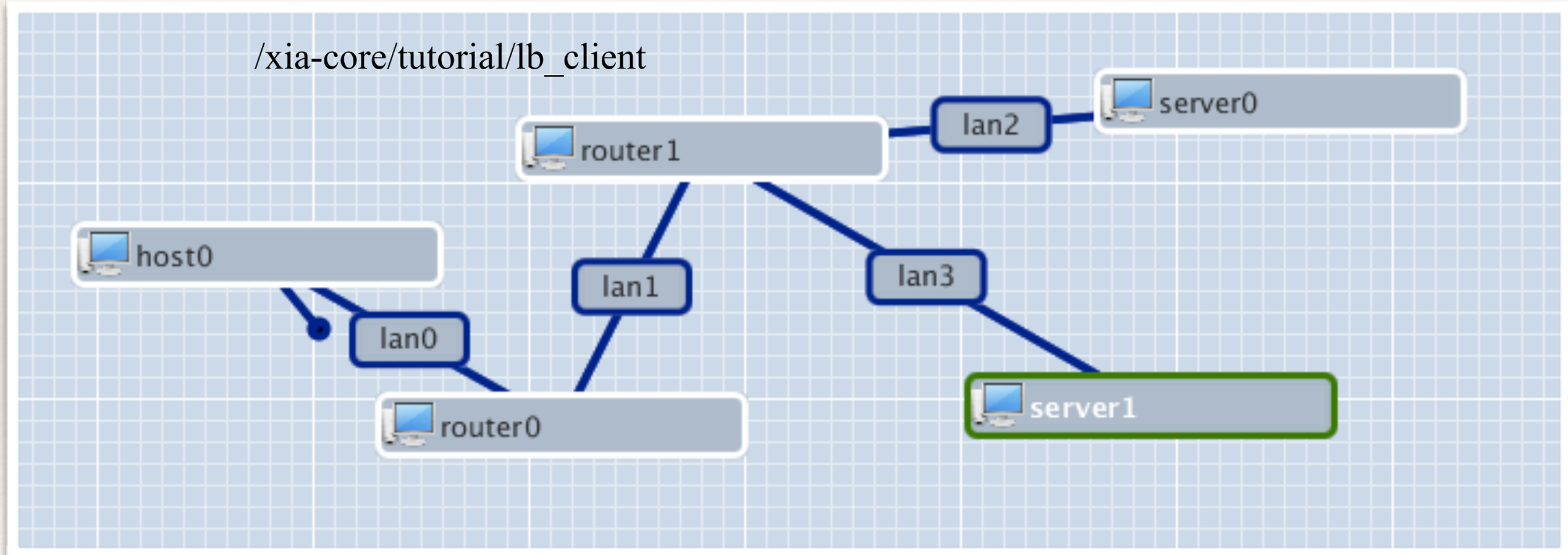
6. Verify the behavior of LID



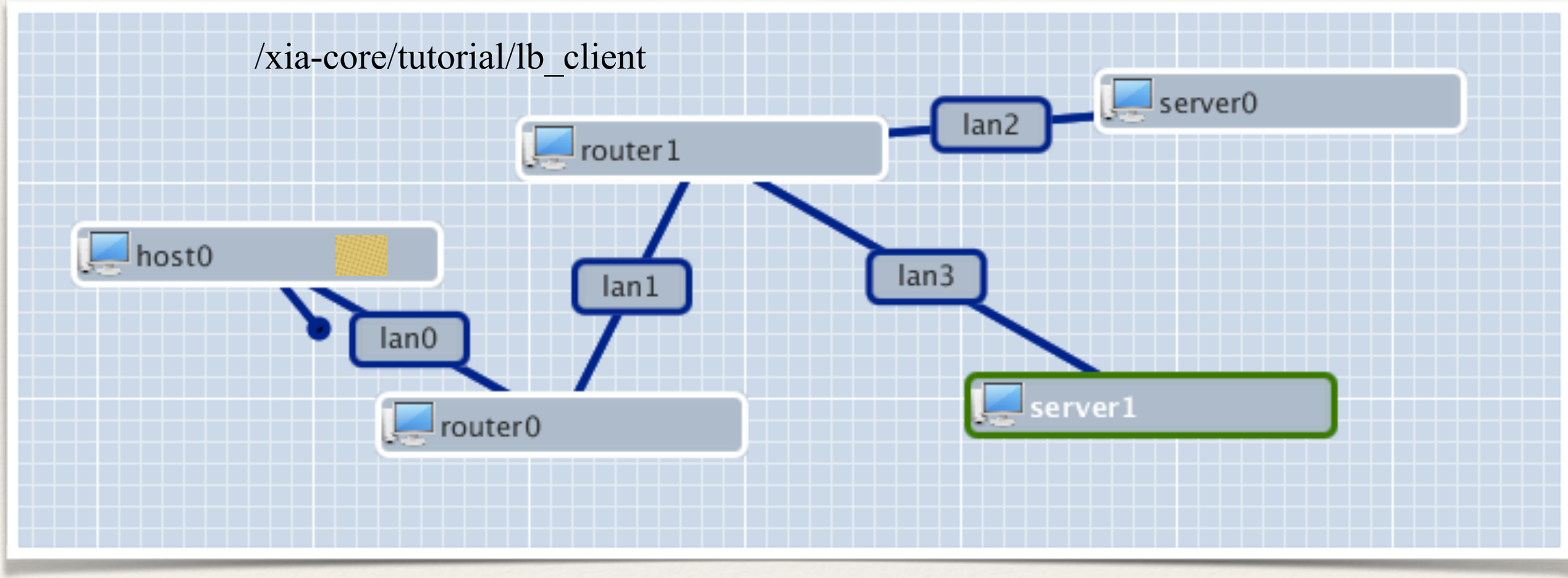
6. Verify the behavior of LID



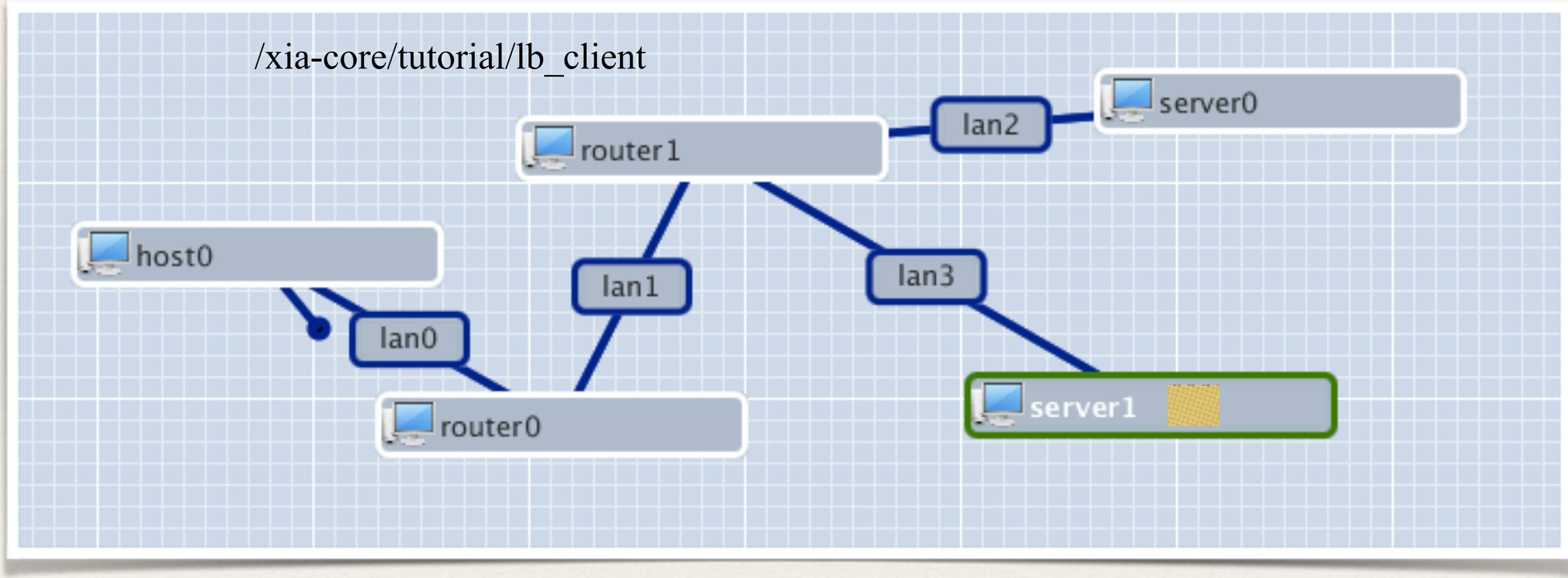
6. Verify the behavior of LID



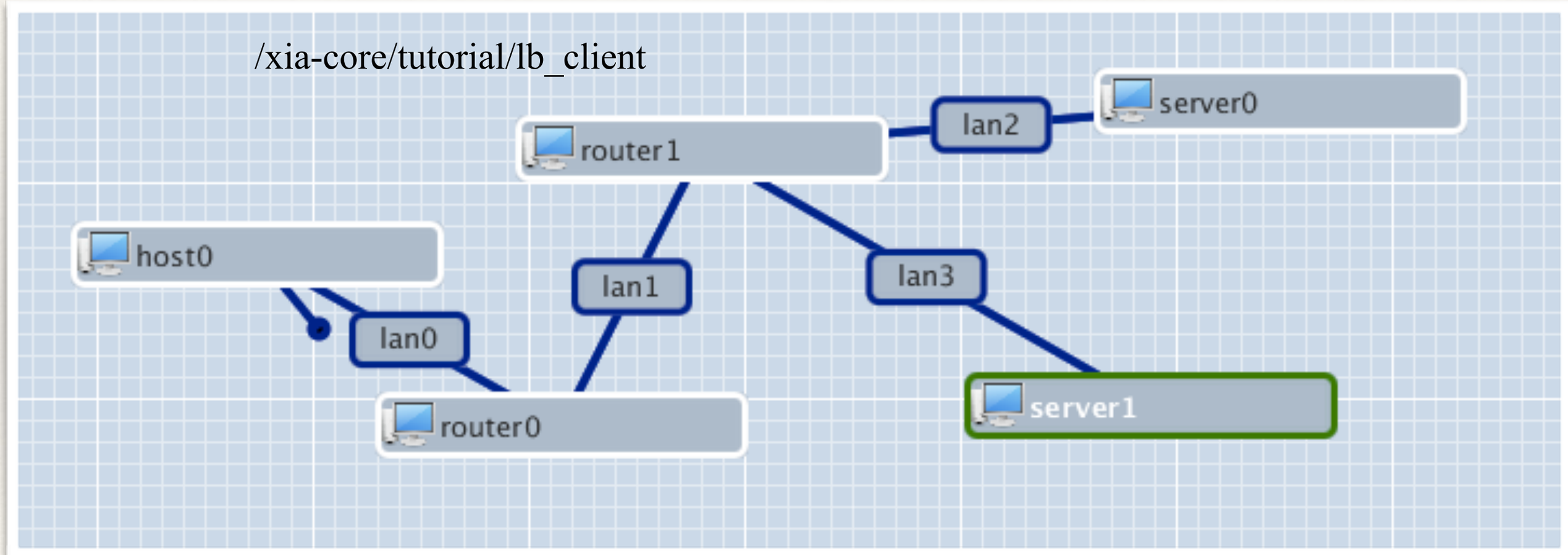
6. Verify the behavior of LID



6. Verify the behavior of LID



6. Verify the behavior of LID



You created a novel network principal, LID,
within 10 lines of codes

You indued LID with load balancing primitive
within another 15 lines of codes

You enabled LID in the network
without upgrading other nodes

Evolvability and Extensibility

Possible Research Directions

Principal\Network Component	Per-hop	Ctl Plane	Addressing/ Intrinsic Security	APIs
CID	Yes	Yes	Yes	Yes
Multicast	Yes	Yes	Yes	-
Mobile	?	Yes	-	?
QoS	Yes	Yes	?	?
?	?	?	?	?
Load balancing	**Yes**	Yes	?	**-**

Possible Research Directions

Principal\Network Component	Per-hop	Ctl Plane	Addressing/ Intrinsic Security	APIs
CID	Yes	Yes	Yes	Yes
Multicast	Yes	Yes	Yes	-
Mobile	?	Yes	-	?
QoS	Yes	Yes	?	?
?	?	?	?	?
Load balancing	**Yes** ✔	Yes	?	-

Questions

Thanks