Case)

Ubuntu 22.04 LTS

Application

(1)

Require Name Resolve
(To: server.co.jp)

\# The application
(e.g., firefox or wget)
tries to get resolved **both
IPv4 and v6 address** for
server.co.jp

Network Stack

(2)

Request Resolving Name
of server.co.jp(v4/v6)

VARLINK

systemd.resolved

(3)

Resolving Name
via all NICs

eth0

Query for server.co.jp(**A**)

Query for server.co.jp(**AAAA**)

A Record

AAAA Record

DNS
(eth0's)

Query for server.co.jp(**A**)

Query for server.co.jp(**AAAA**)

eth1

A Record

AAAA Record

DNS
(eth1's)

(4)

(5) Application get the
addresses of server.co.jp
(v4 and v6)

Returning the Result of Response **via all NICs**
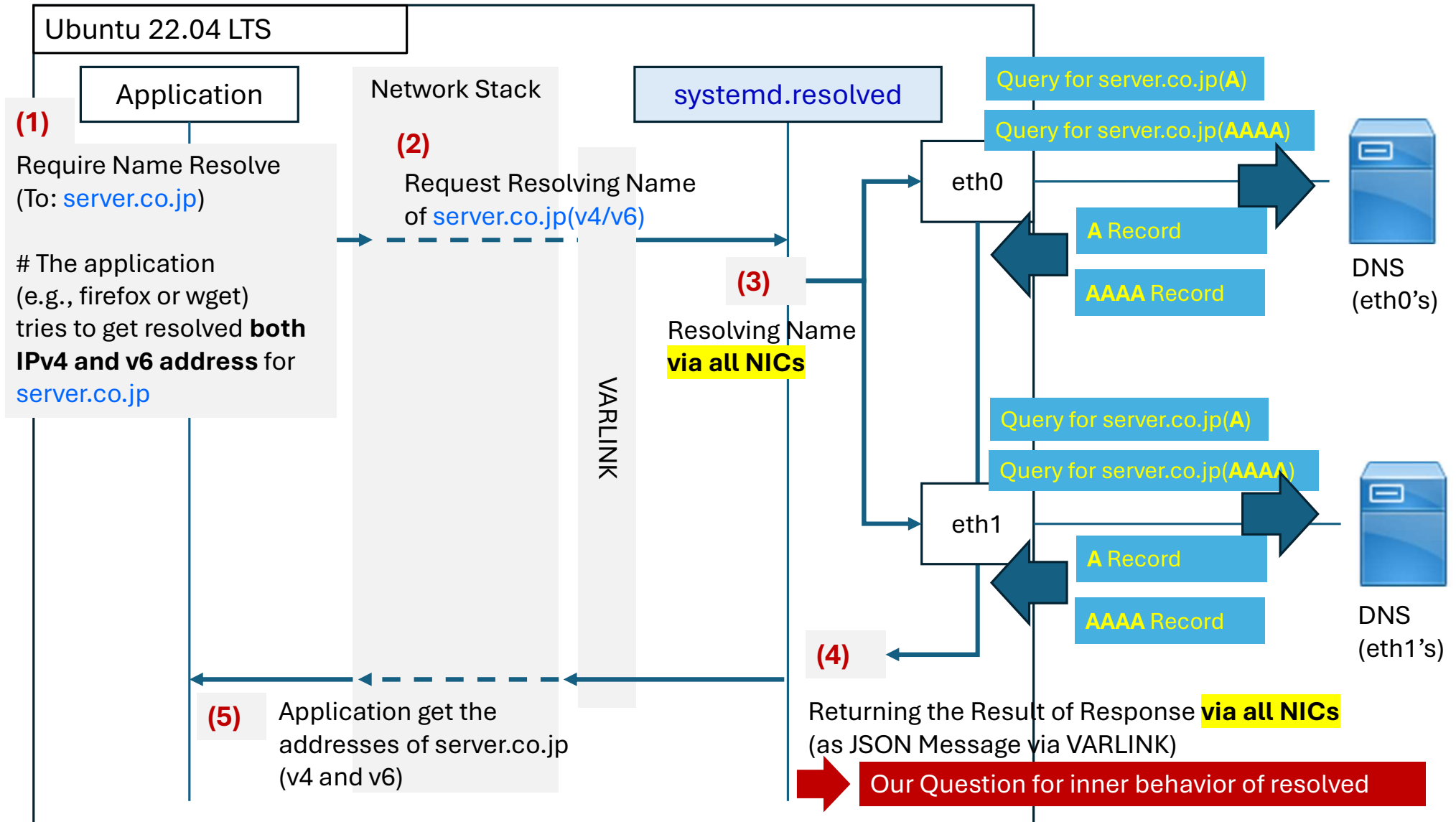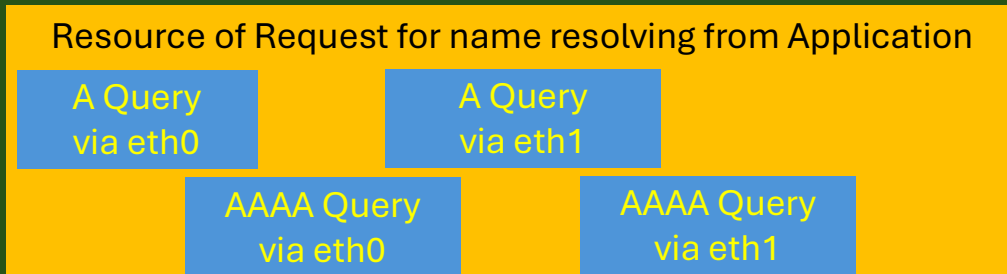(as JSON Message via VARLINK)

Our Question for inner behavior of resolved
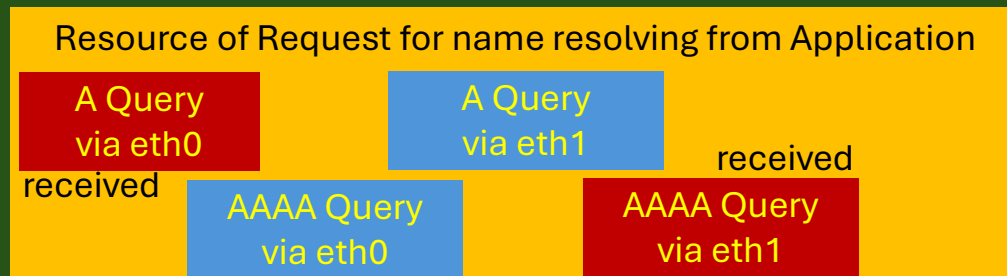
How to manage the resource of query
for resolving name ?

1. When Resolving Name

Allocate the resource for both A and AAAA query

Resource of Request for name resolving from Application

A Query
via eth0

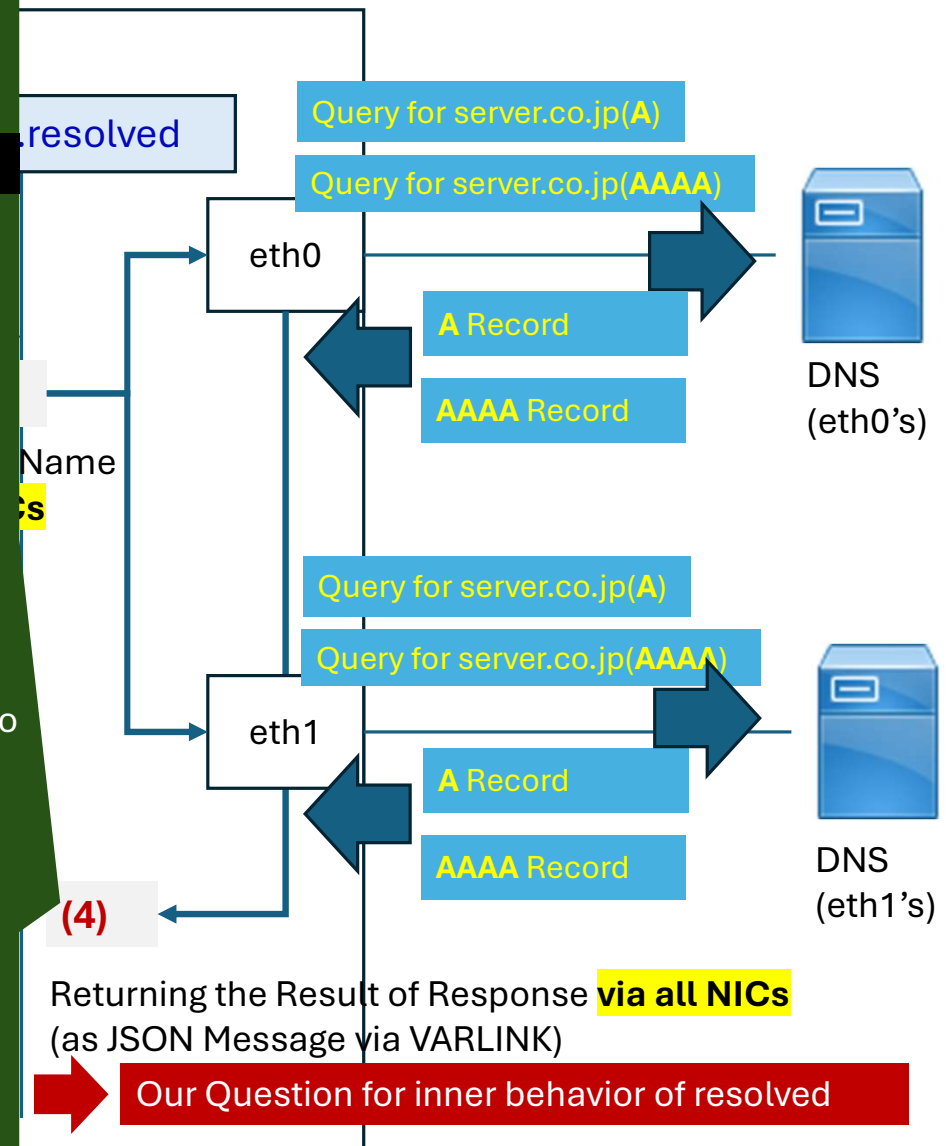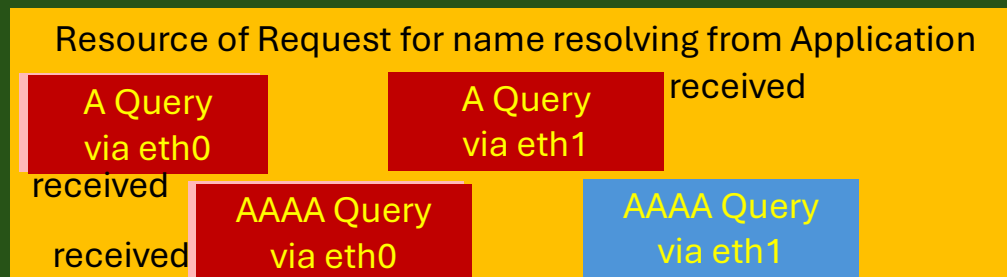A Query
via eth1

AAAA Query
via eth0

AAAA Query
via eth1

2. Resolved A and AAAA record through eth0 and eth1, expectively

A via eth0 and AAAA via eth1 is received.
At this time, resolved got the A and AAAA record, but wait for
completing queries via any interface.

Resource of Request for name resolving from Application

A Query
via eth0

A Query
via eth1

received

received

AAAA Query
via eth0

AAAA Query
via eth1

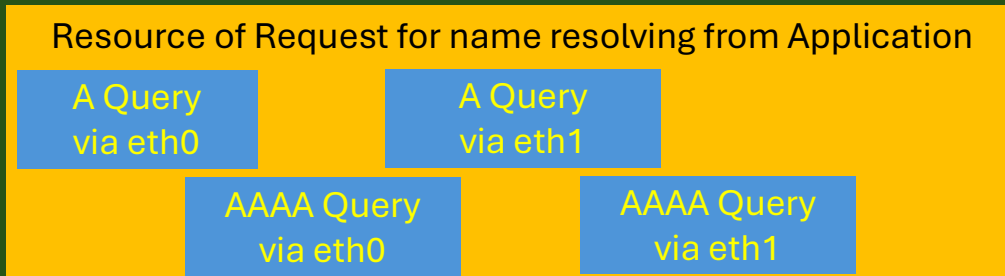**3. Resolved A and AAAA record through one of the interfaces**

one of the interfaces(e.g., eth0) received all queries and get ready to
return the result
(AAAA query for eth1 is not resolved, but not being waited)
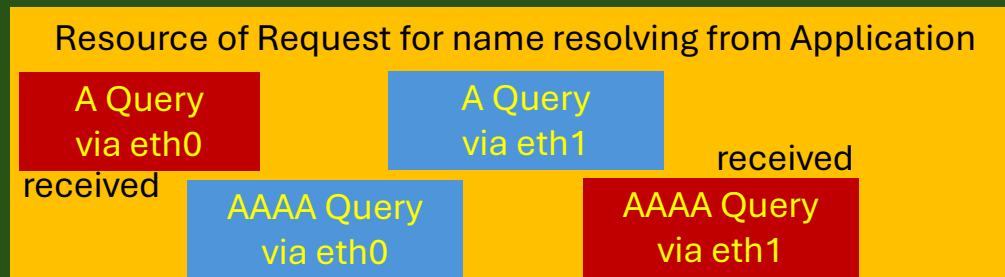
Resource of Request for name resolving from Application

received

A Query
via eth0

A Query
via eth1

received

AAAA Query
via eth0

AAAA Query
via eth1

received

resolved

Query for server.co.jp(**A**)

Query for server.co.jp(**AAAA**)

eth0

DNS
(eth0's)

A Record

AAAA Record

Name

Query for server.co.jp(**A**)

Query for server.co.jp(**AAAA**)

eth1

DNS
(eth1's)

A Record

AAAA Record

(4)

Returning the Result of Response **via all NICs**
(as JSON Message via VARLINK)

Our Question for inner behavior of resolved

How to manage the resource of query for resolving name ?

1. When Resolving Name

Allocate the resource for both A and AAAA query

Resource of Request for name resolving from Application

- A Query via eth0
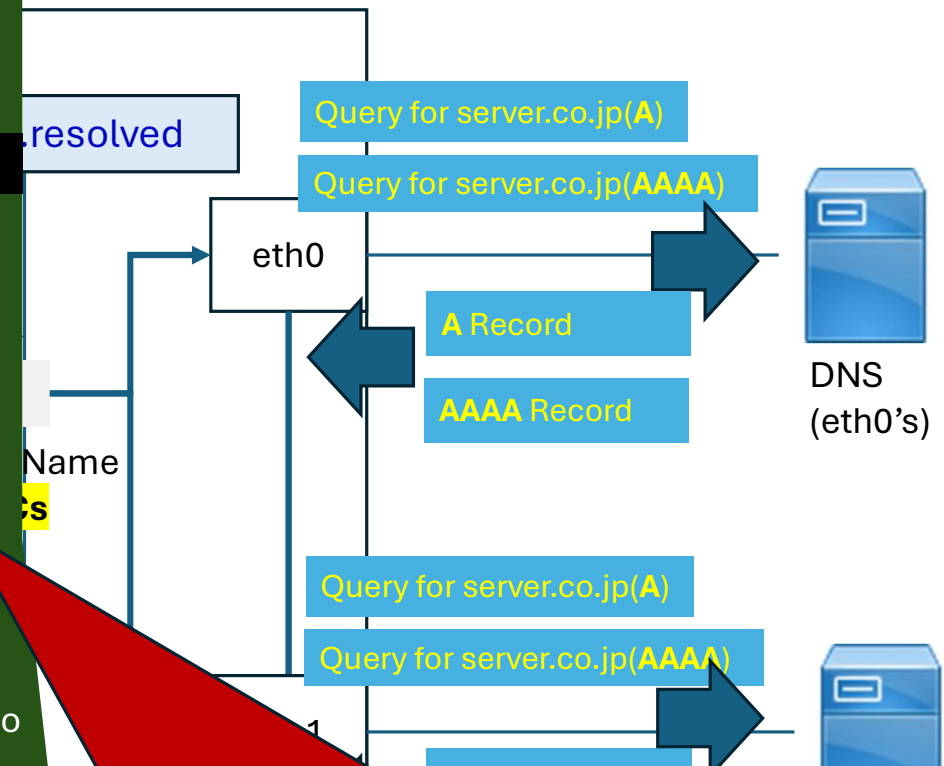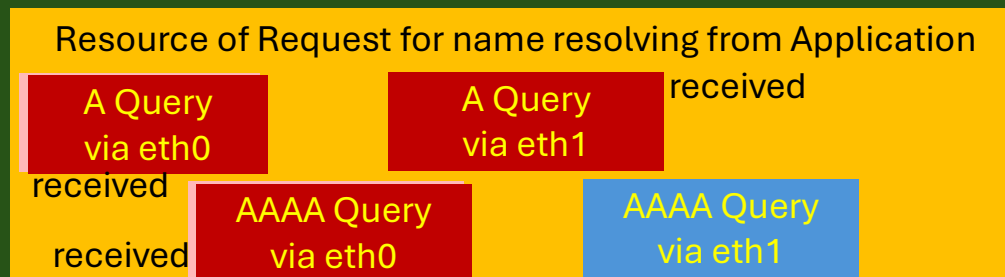- A Query via eth1
- AAAA Query via eth0
- AAAA Query via eth1

2. Resolved A and AAAA record through eth0 and eth1, expectively

A via eth0 and AAAA via eth1 is received.
At this time, resolved got the A and AAAA record, but wait for completing queries via any interface.

Resource of Request for name resolving from Application

- A Query via eth0 — received
- A Query via eth1
- AAAA Query via eth0
- AAAA Query via eth1 — received

3. Resolved A and AAAA record through one of the interfaces

one of the interfaces(e.g., eth0) received all queries and get ready to return the result
(AAAA query for eth1 is not resolved, but not being waited)

Resource of Request for name resolving from Application

- A Query via eth0 — received
- A Query via eth1 — received
- AAAA Query via eth0 — received
- AAAA Query via eth1

resolved

Query for server.co.jp(**A**)
Query for server.co.jp(**AAAA**)

eth0

A Record
AAAA Record

DNS (eth0's)

Name

Query for server.co.jp(**A**)
Query for server.co.jp(**AAAA**)

Question)
At this point, each interfaces unfinished completing to resolve, but A and AAAA record have been received, namely, could resolved return the result to application at this point?

Is there any reason or restriction that systemd should wait for the rest of queries?