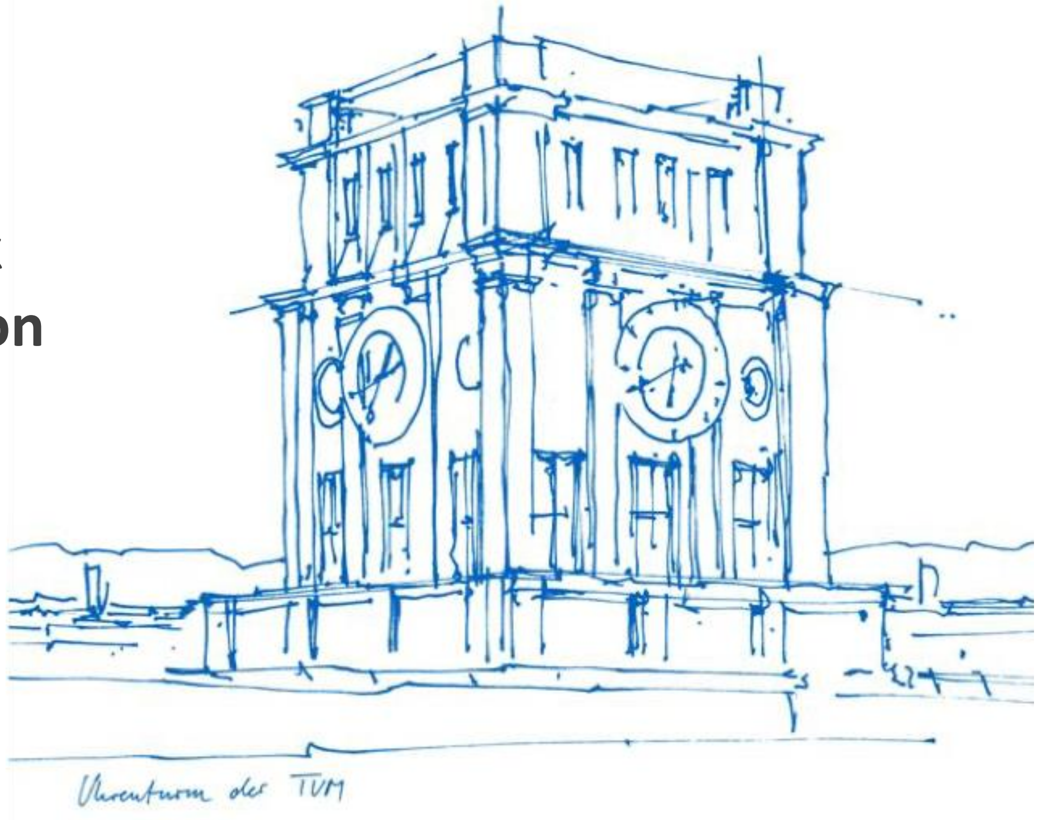
Academic Salon on Low-Latency Communication, Programmable Network Components and In-Network Computation

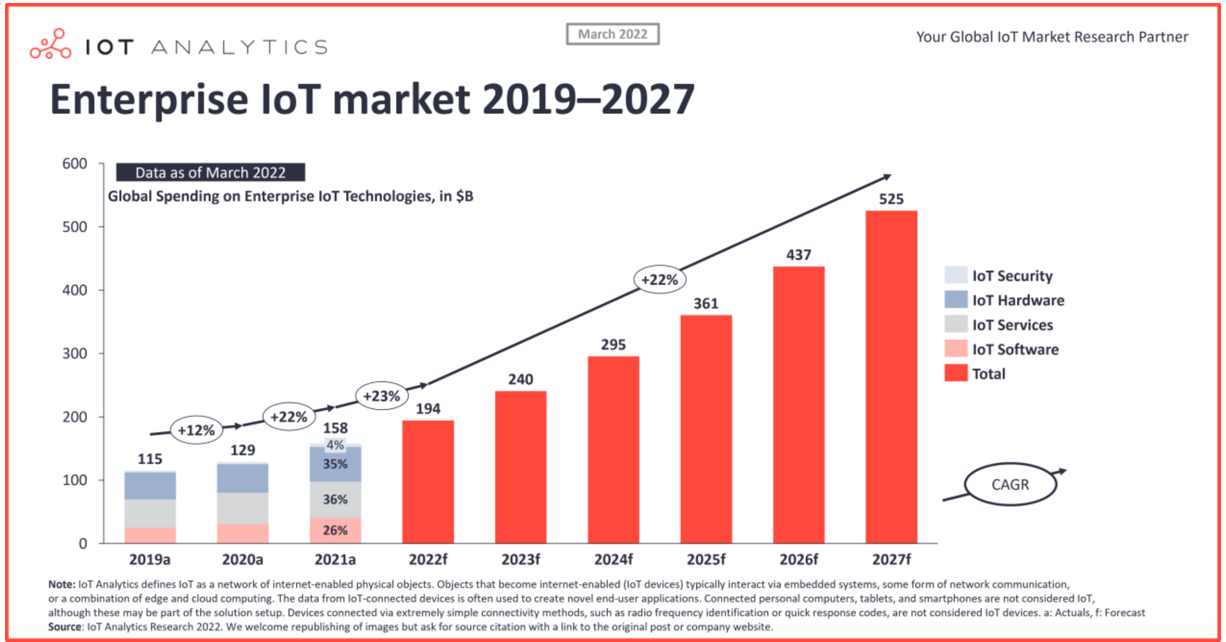


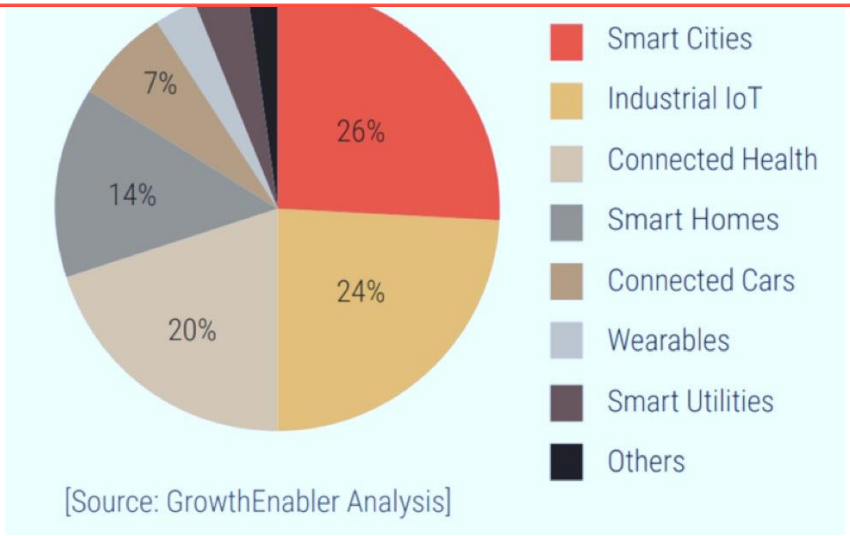
Topological Addressing Enabling Efficient IoT Communication

Luigi Iannone, Zhe Lou, Li Guangpeng

Motivation

- **Massive expansion of the network edge that is driven by the "Internet of Things" (IoT) technology**

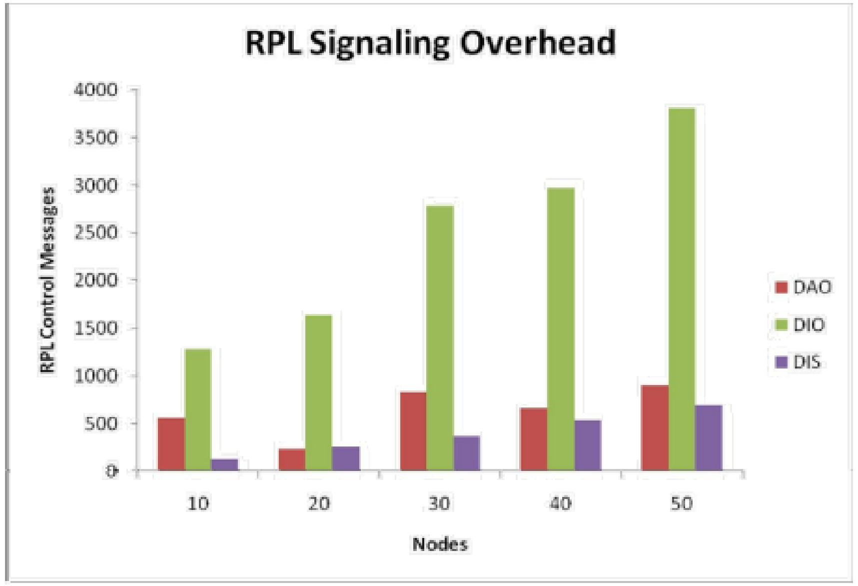




Usual Routing approach

- 6LowPAN/6Lo/LPWAN WGs, in the IETF, address many foundational issues for those type of deployments
- Existing solutions, however, may have some shortcomings:

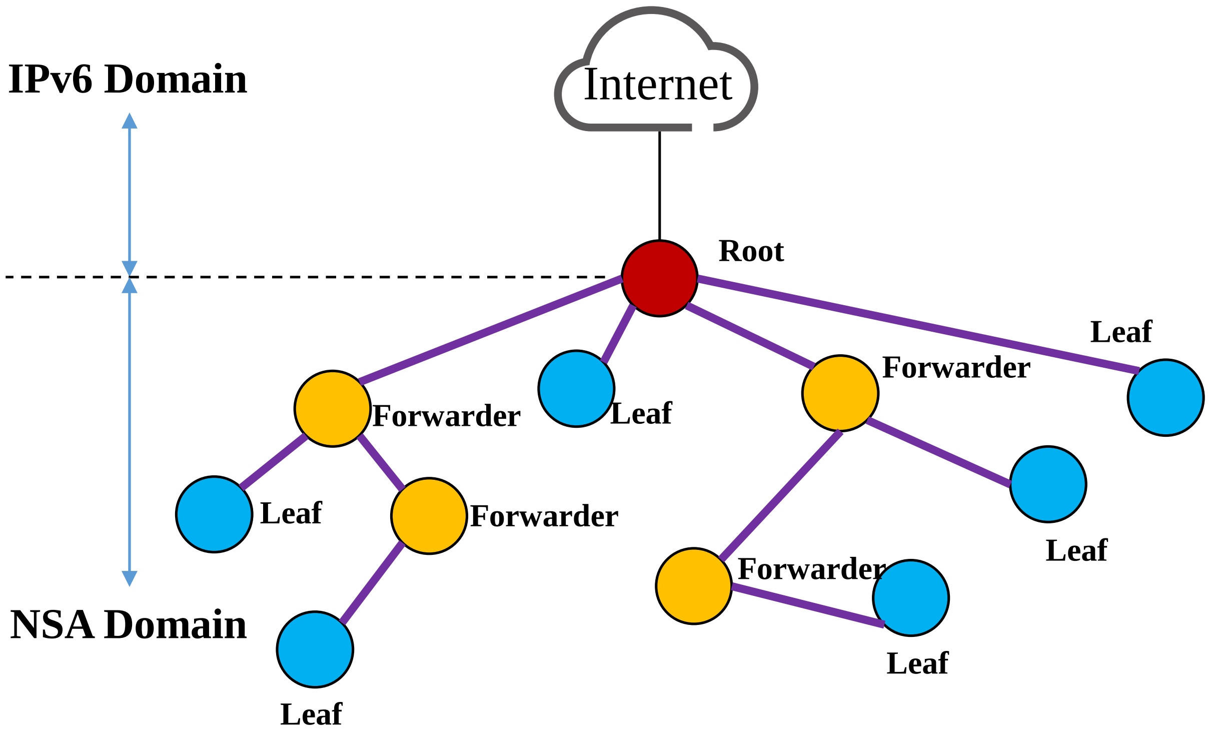
Technology	Problem(especially in large scale LLN)
6lowpan-DHCPv6	Consume bandwidth and time before node working, by applying for address from centralized server through multiple hops
6lowpan-AutoConfig	Using large address space to reduce confliction, implicating longer address and larger routing table, thus limit scale of network
6lowpan-RH	RPL information causes extra overhead of packet. Routers consumes resource to advertise、store、manage routing table
6lowpan-IPHC	Context based address uncompressing consumes extra computing resource. 6lo-RPL(RFC 8138) avoids uncompressing hop-by-hop, however bring much more complexity in routing.



Source: Umamaheswari S, Negi A. Internet of Things and RPL routing protocol: A study and evaluation. In2017 International Conference on Computer Communication and Informatics (ICCCI) 2017 Jan 5 (pp. 1-7). IEEE.

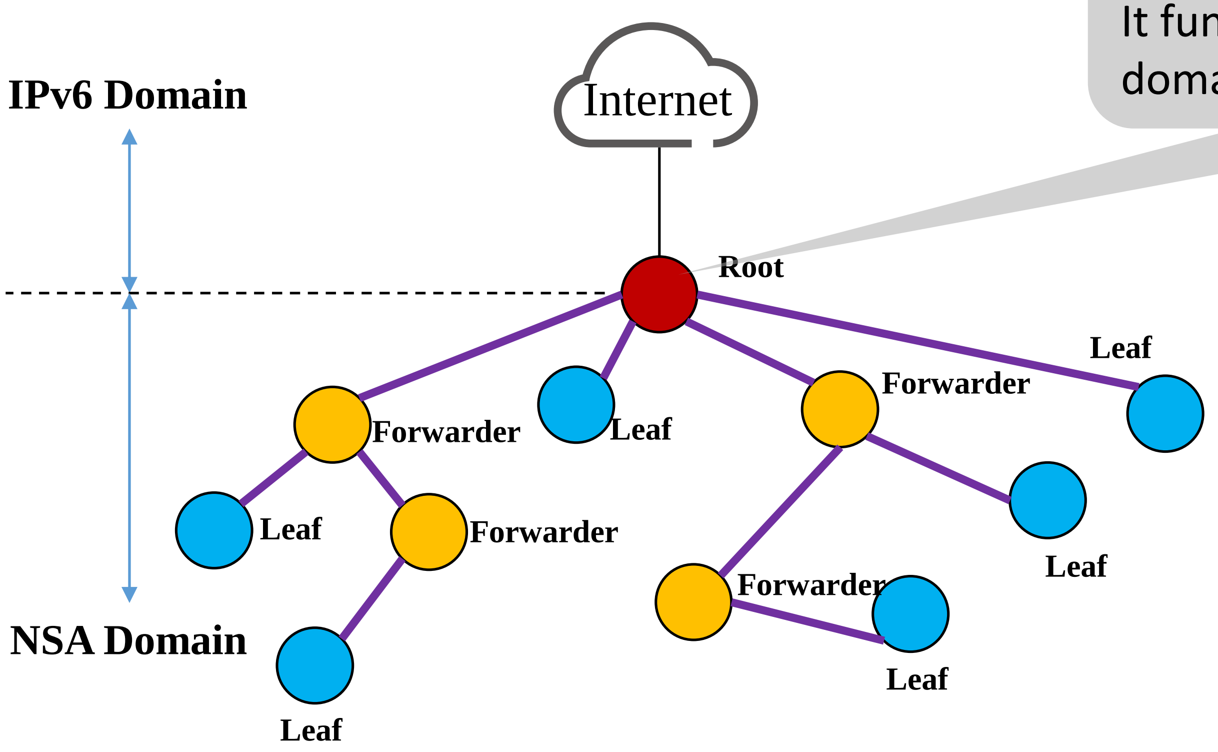
Does simpler and more efficient addressing/routing exist based on (but beyond) previous work?

Native Short Addresses (NSA) Architecture

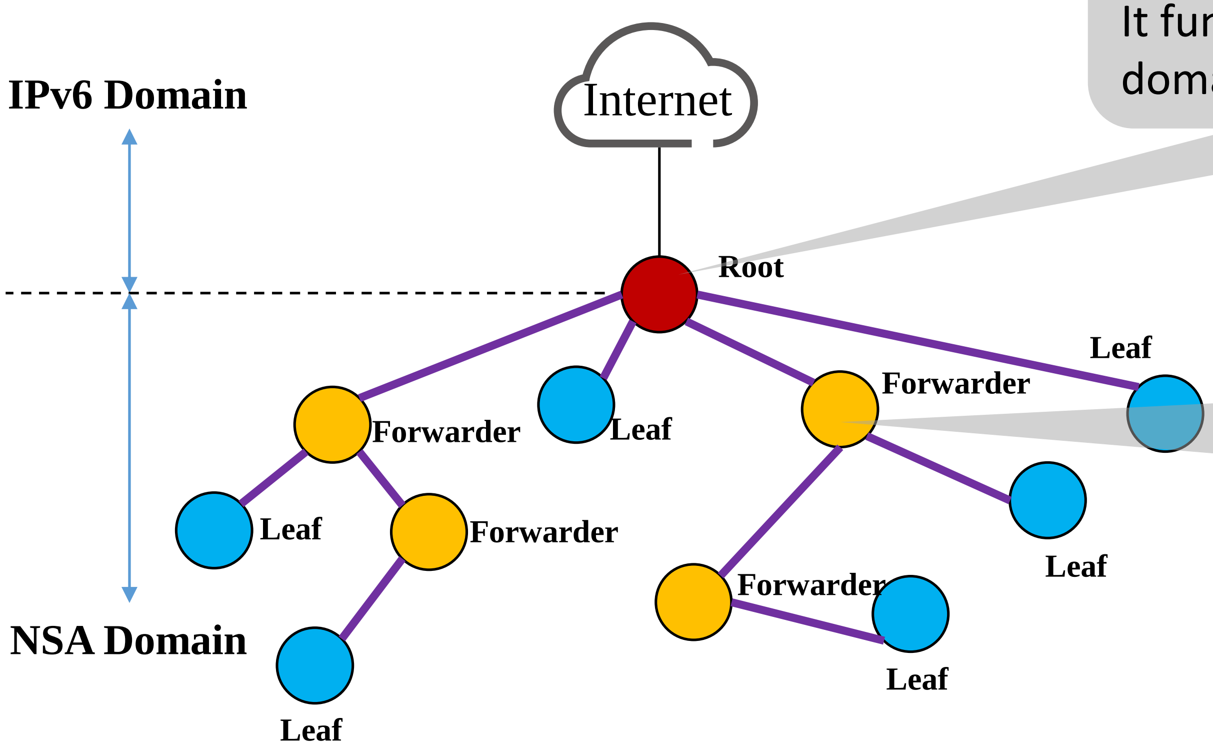


Native Short Addresses (NSA) Architecture

The root node is responsible for the management of the overall NSA network. It functions as a gateway between the NSA domain and the Internet.



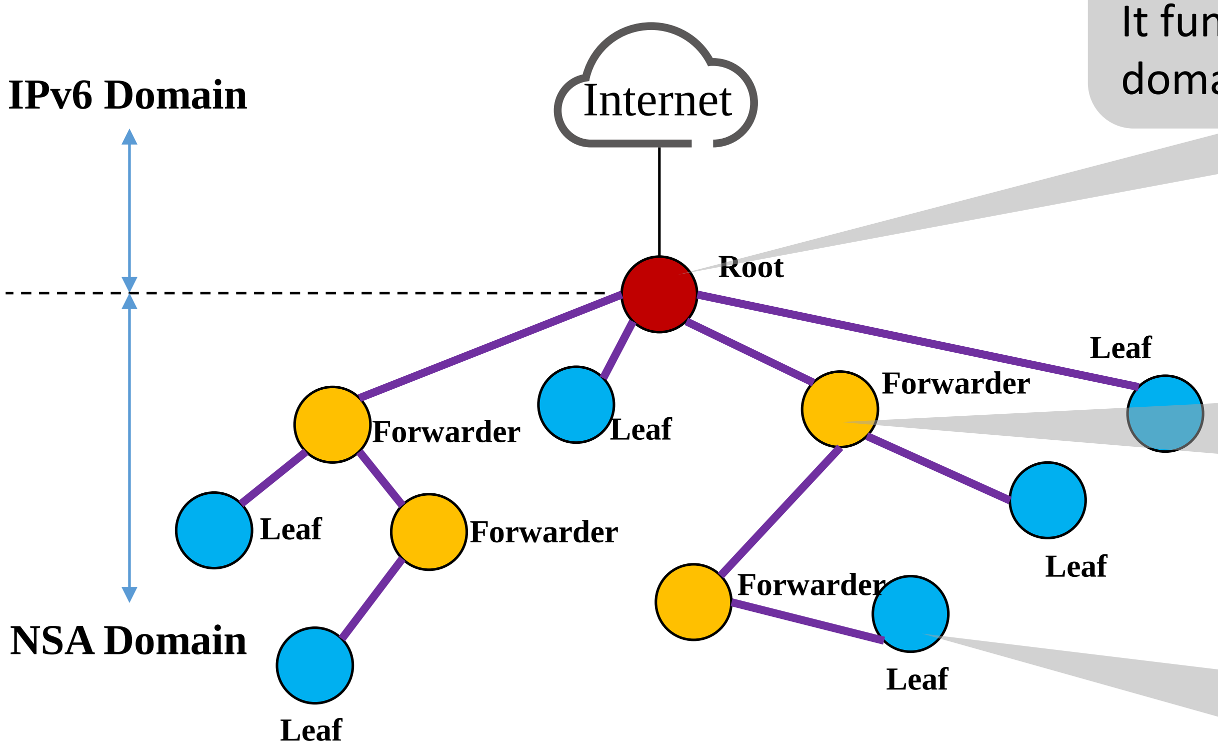
Native Short Addresses (NSA) Architecture



The root node is responsible for the management of the overall NSA network. It functions as a gateway between the NSA domain and the Internet.

A forwarder node is responsible for, as the name suggests, forwarding traffic between its parent and the children, according to NSA addressing.

Native Short Addresses (NSA) Architecture



The root node is responsible for the management of the overall NSA network. It functions as a gateway between the NSA domain and the Internet.

A forwarder node is responsible for, as the name suggests, forwarding traffic between its parent and the children, according to NSA addressing.

A leaf node has relatively simpler operation, as it is either a source or destination of traffic.

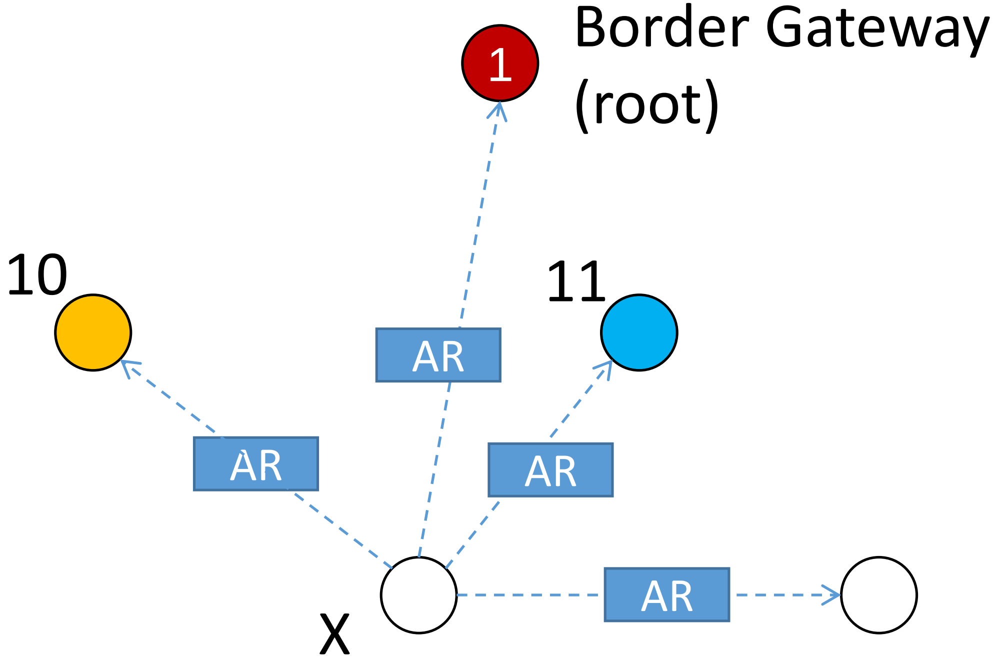
Address Allocation Function

Algorithm 1 NSA Allocation Function pseudo code

function AF($role, f, l$)
    $pfx \leftarrow Current\ Node\ NSA\ Address$
    if ($role = forwarder$) **then**
        $sfx \leftarrow Concatenate(b(f), 0)$
        $f \leftarrow f + 1$
    else
        $sfx \leftarrow Concatenate(b(l), 1)$
        $l \leftarrow l + 1$
    end if
    return Concatenate(pfx, sfx)
end function

New Address = Parent Address + string of ‘1’ + (0 if forwarder; 1 if Leaf)

Address Allocation (1/5)



Algorithm 1 NSA Allocation Function pseudo code

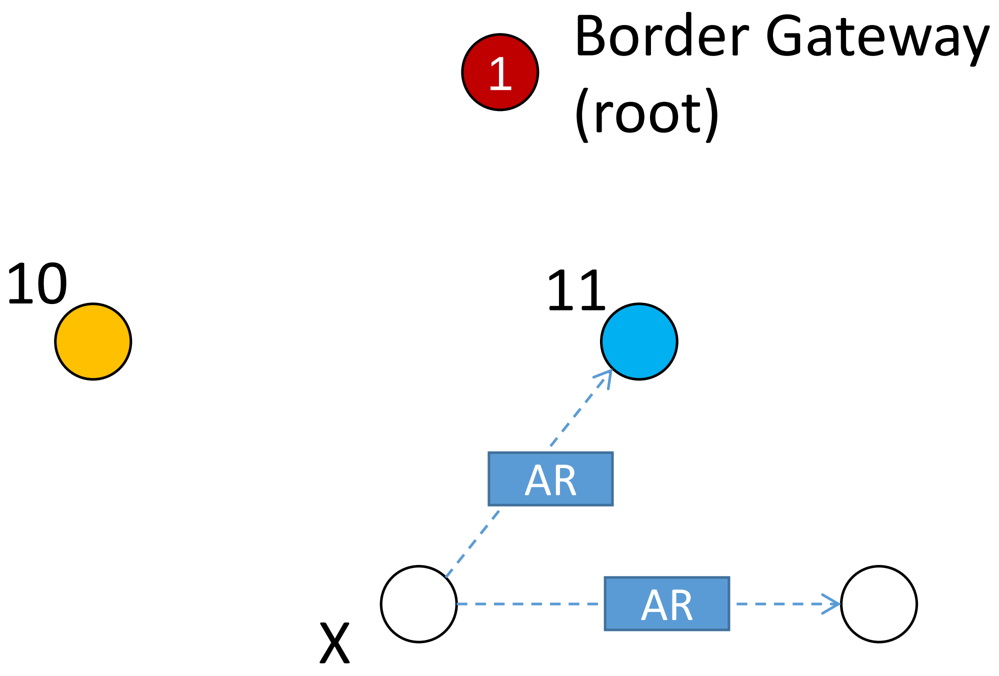
```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```

AR ({**forwarder** | leaf}, Nodeid=X)

AR – Address Request

Node with no address | Root | Forwarder | Leaf

Address Allocation (2/5)



AR ({**forwarder** | leaf}, Nodeid=X)

AR – Address Request

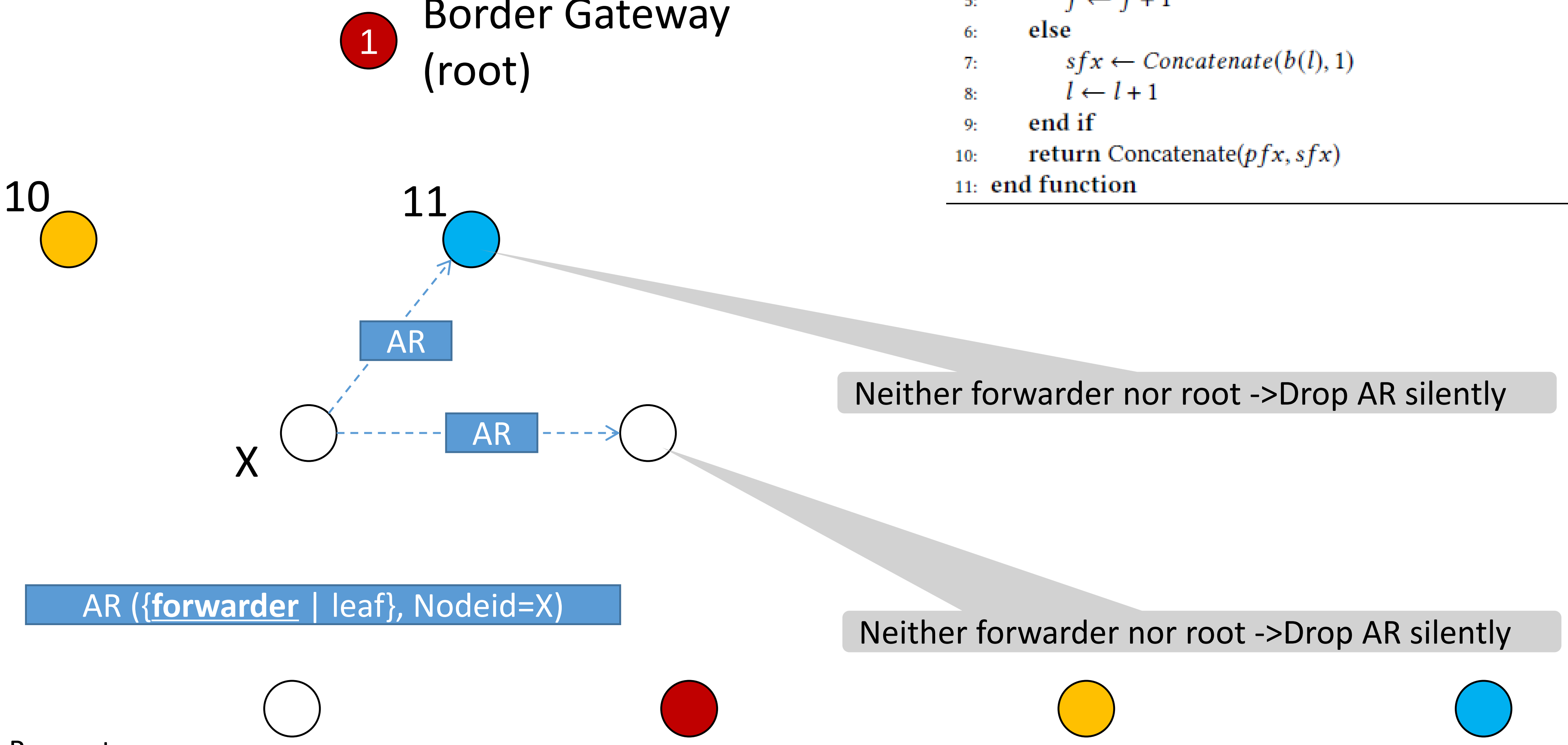
Algorithm 1 NSA Allocation Function pseudo code

```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```

Address Allocation (2/5)

Algorithm 1 NSA Allocation Function pseudo code

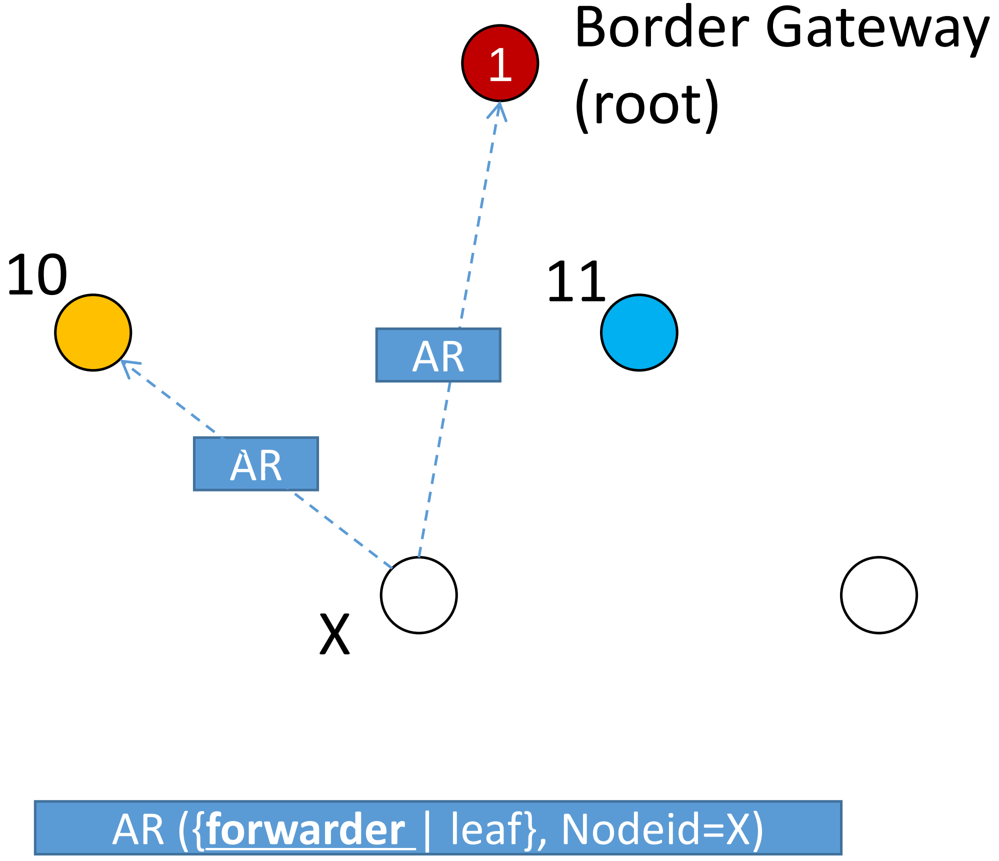
```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```



AR – Address Request

Node with no address | Root | Forwarder | Leaf

Address Allocation (3/5)

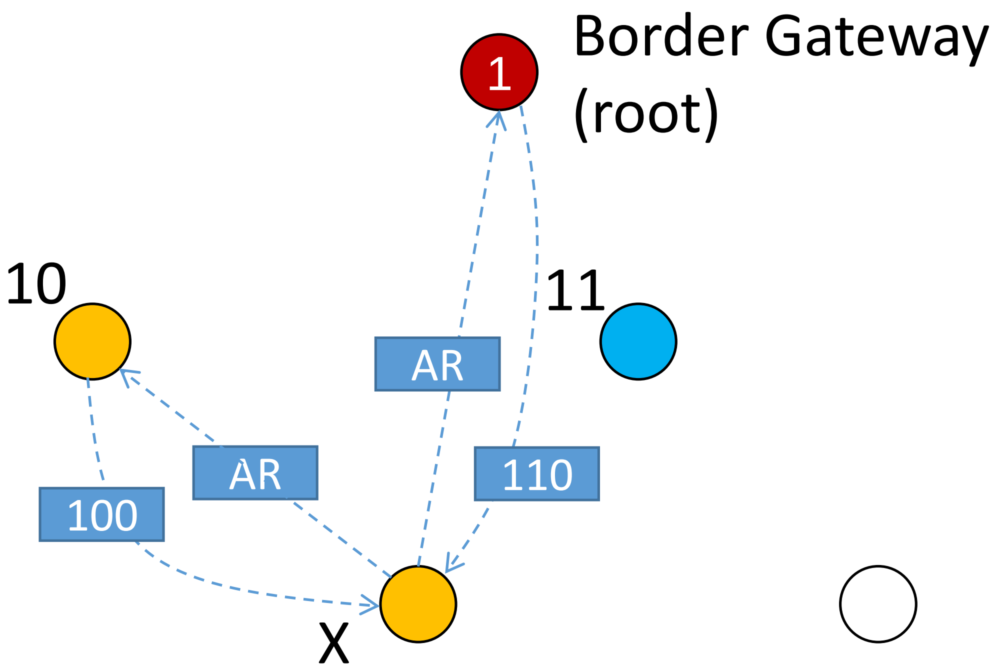


Algorithm 1 NSA Allocation Function pseudo code

```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```

AR – Address Request

Address Allocation (4/5)



AR (**forwarder**, Nodeid=X)

Algorithm 1 NSA Allocation Function pseudo code

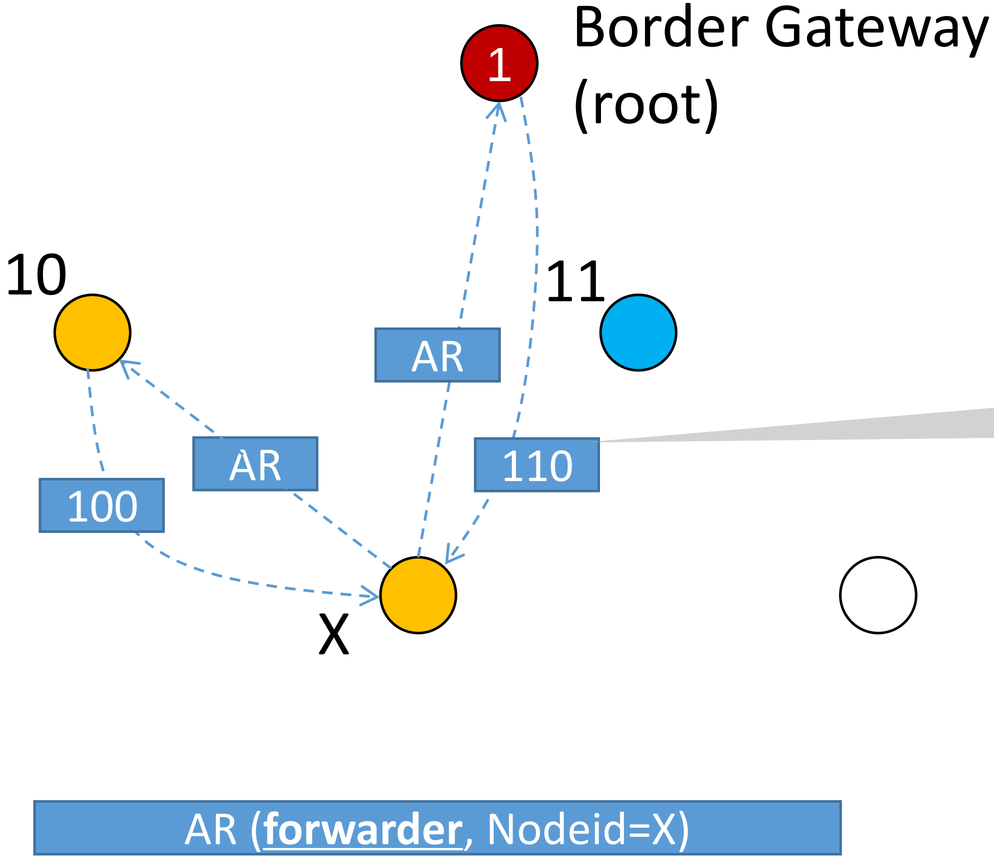
```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```

AR – Address Request

Address Allocation (4/5)

Algorithm 1 NSA Allocation Function pseudo code

```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```



1. Pfx = 1 (Root address)
2. B(f) = 1 (Second forwarder child)
3. Concatenate Pfx+B(f)+0 = 1+1+0
4. Offered Address: 110

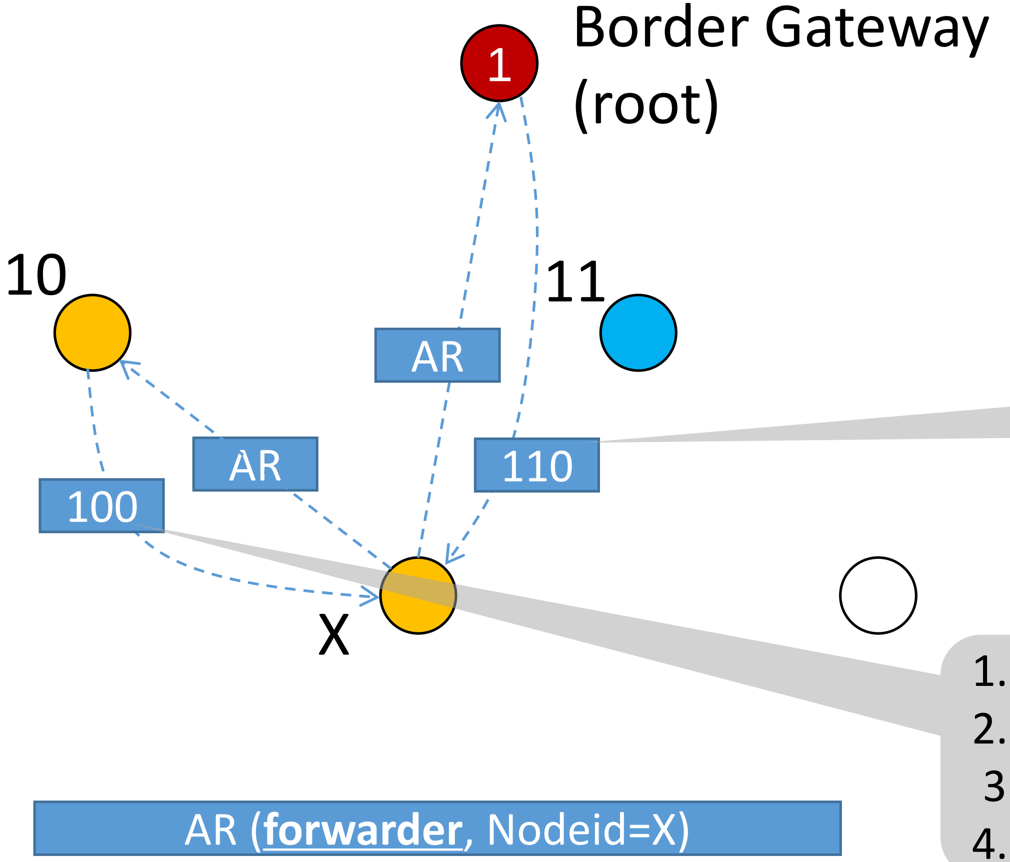
AR – Address Request

Node with no address | Root | Forwarder | Leaf

Address Allocation (4/5)

Algorithm 1 NSA Allocation Function pseudo code

```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```



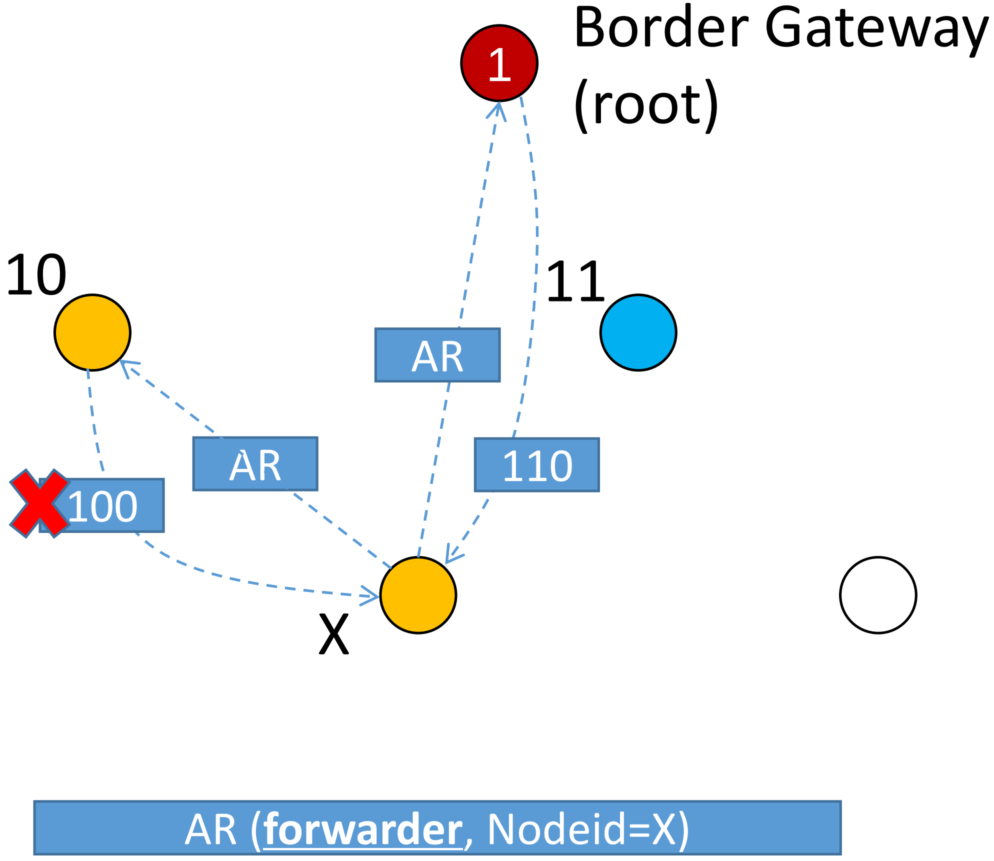
1. Pfx = 1 (Root address)
2. B(f) = 1 (Second forwarder child)
3. Concatenate Pfx+B(f)+0 = 1+1+0
4. Offered Address: 110

1. Pfx = 10 (Forwarder address)
2. B(f) = ‘’ (First forwarder child)
3. Concatenate Pfx+B(f)+0 = 10+‘’+0
4. Offered Address: 100

AR – Address Request

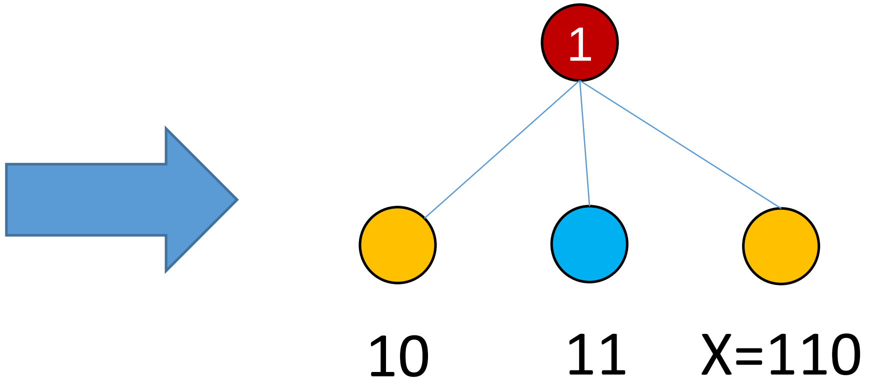
Node with no address | Root | Forwarder | Leaf

Address Allocation (5/5)



Algorithm 1 NSA Allocation Function pseudo code

```
function AF(role, f, l)
    pfx ← Current Node NSA Address
    if (role = forwarder) then
        sfx ← Concatenate(b(f), 0)
        f ← f + 1
    else
        sfx ← Concatenate(b(l), 1)
        l ← l + 1
    end if
    return Concatenate(pfx, sfx)
end function
```



AR – Address Request

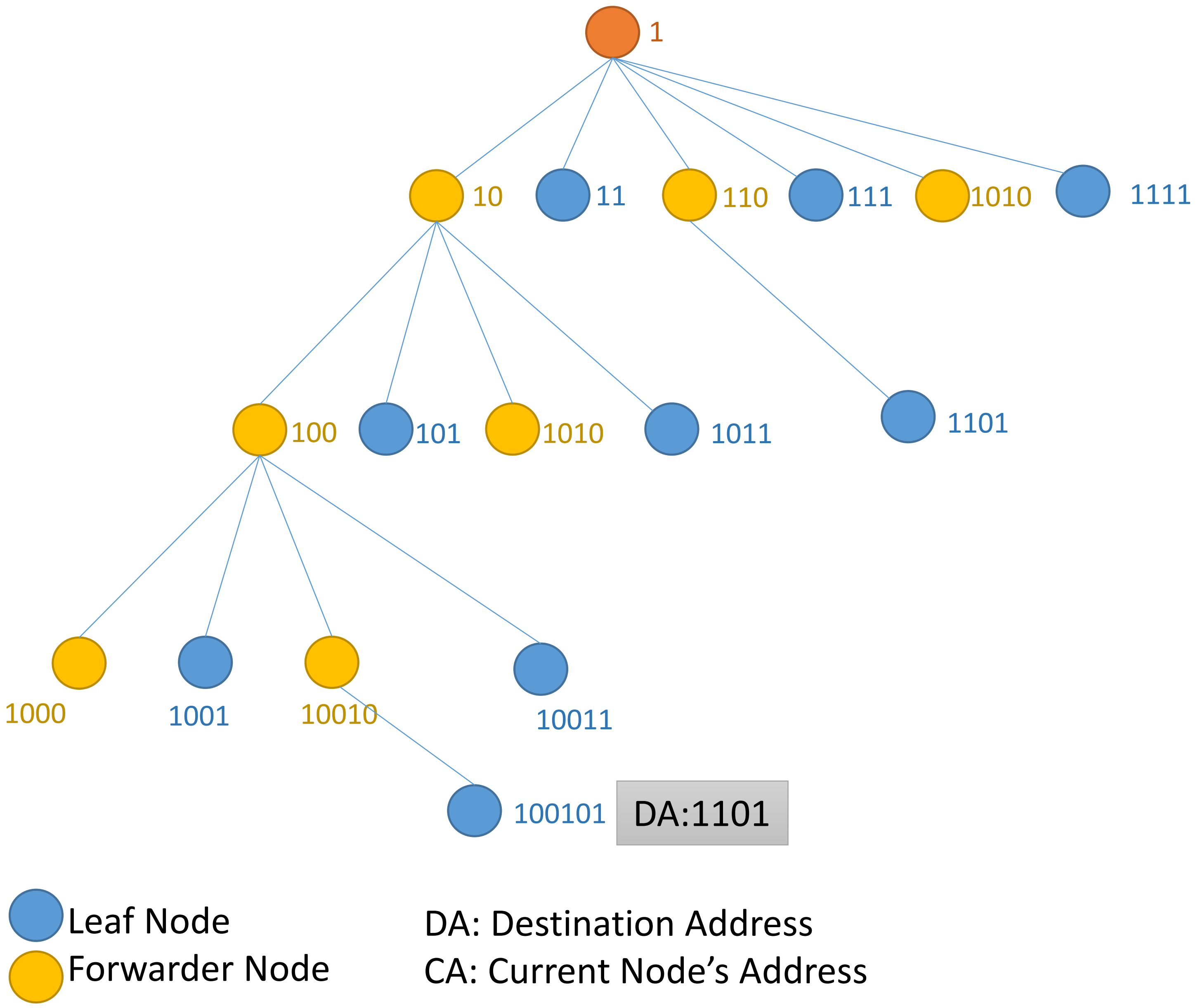
Node with no address

Root

Forwarder

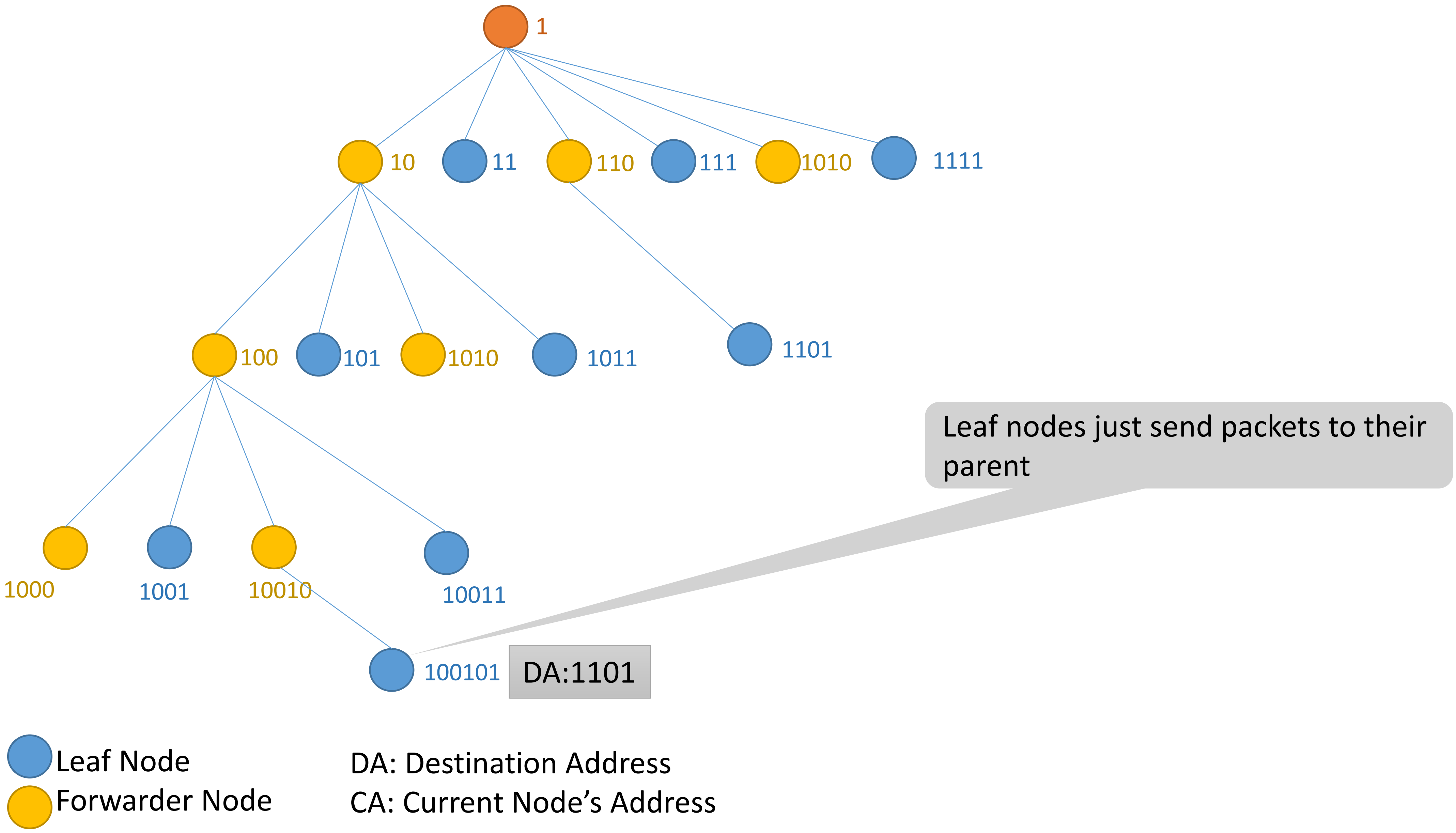
Leaf

NSA Stateless Forwarding



DA: Destination Address
CA: Current Node’s Address

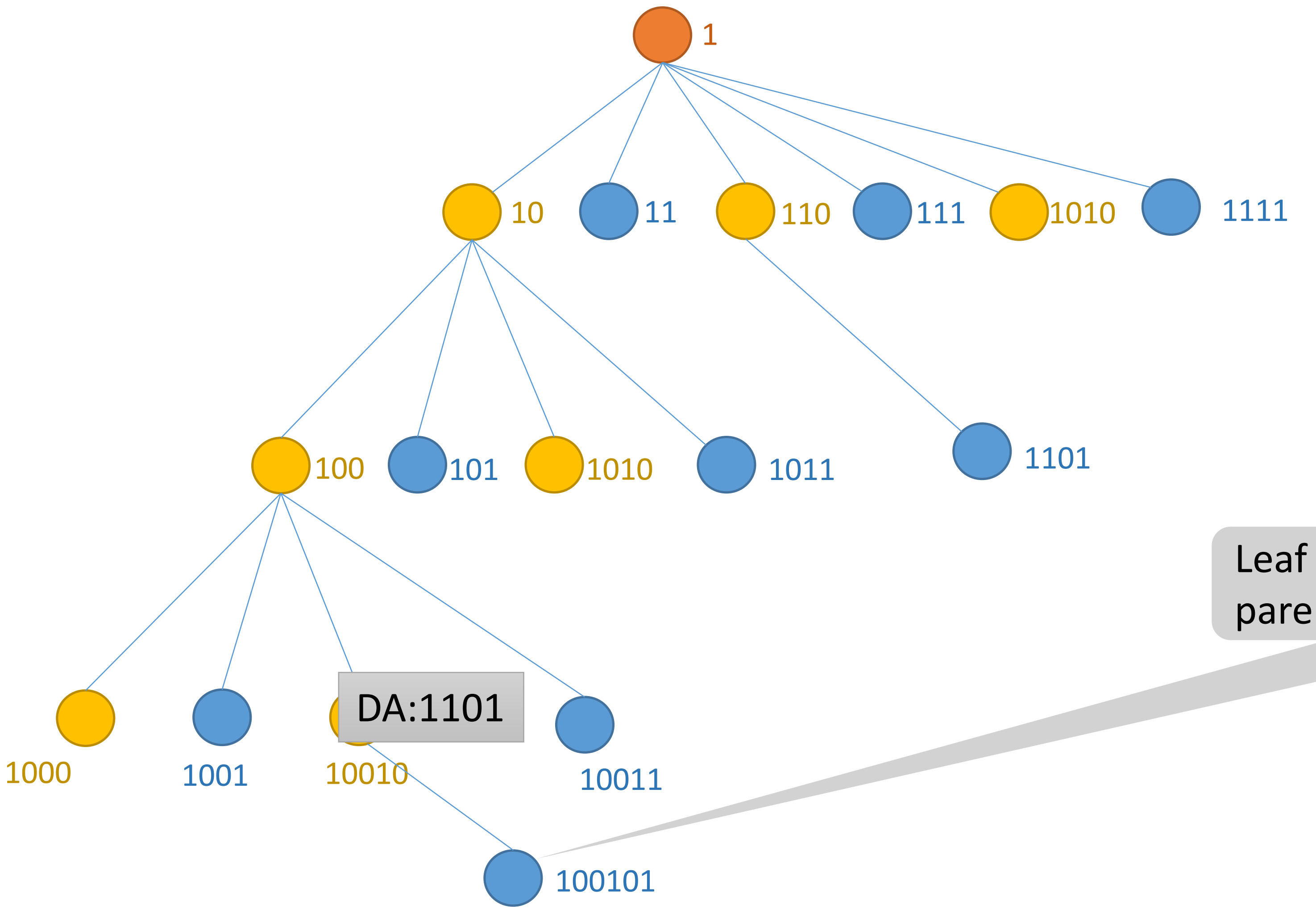
NSA Stateless Forwarding



DA: Destination Address

CA: Current Node's Address

NSA Stateless Forwarding

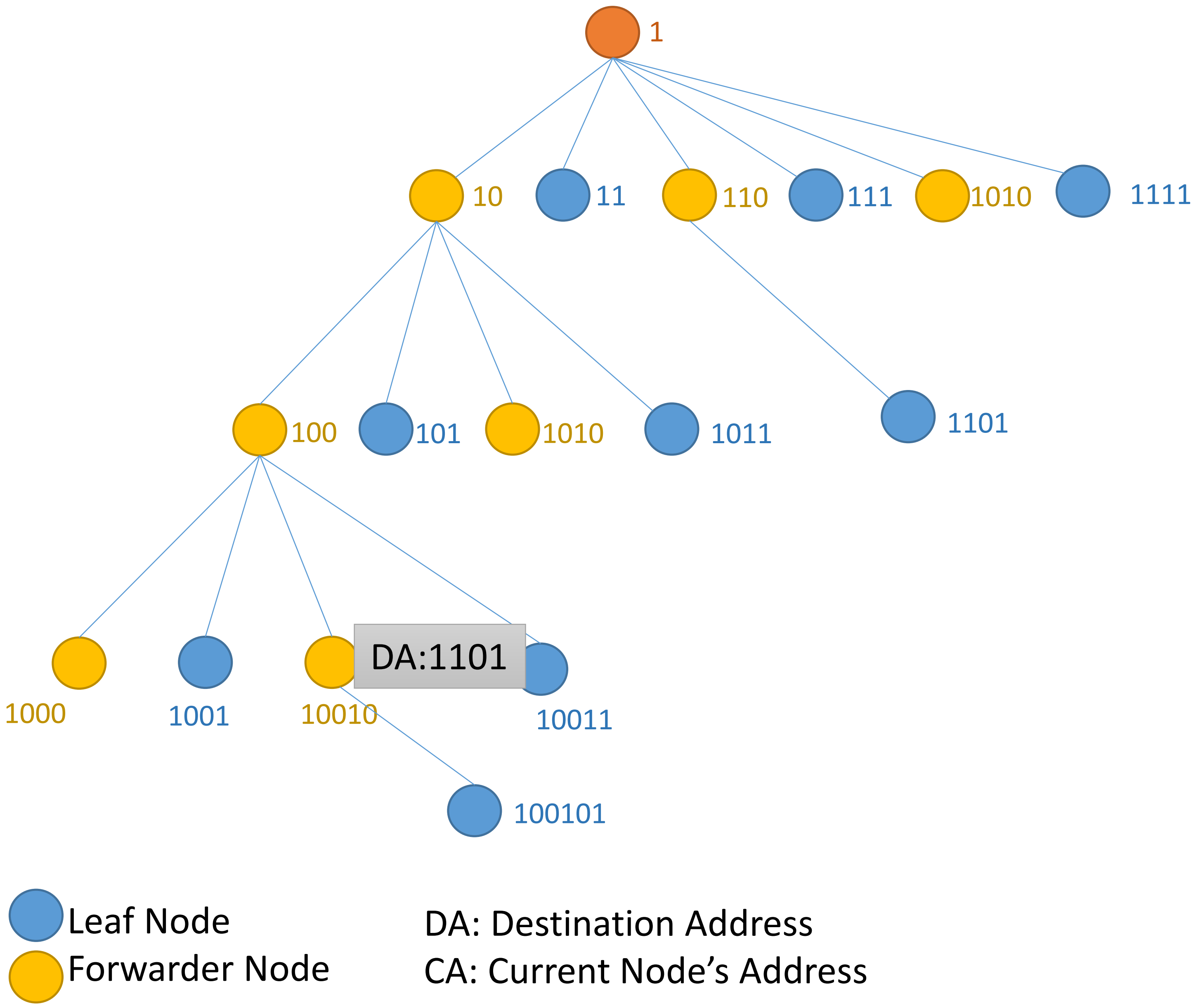


Leaf nodes just send packets to their parent

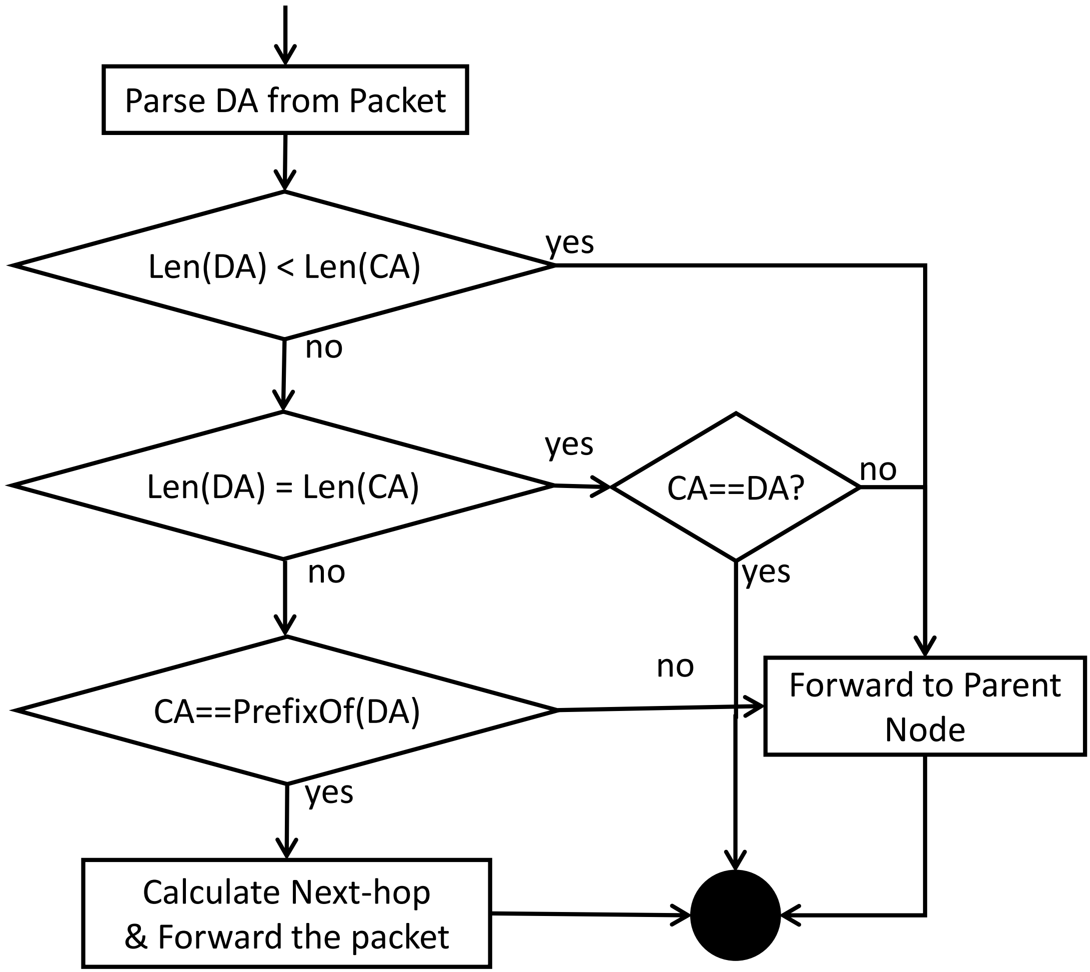
Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node’s Address

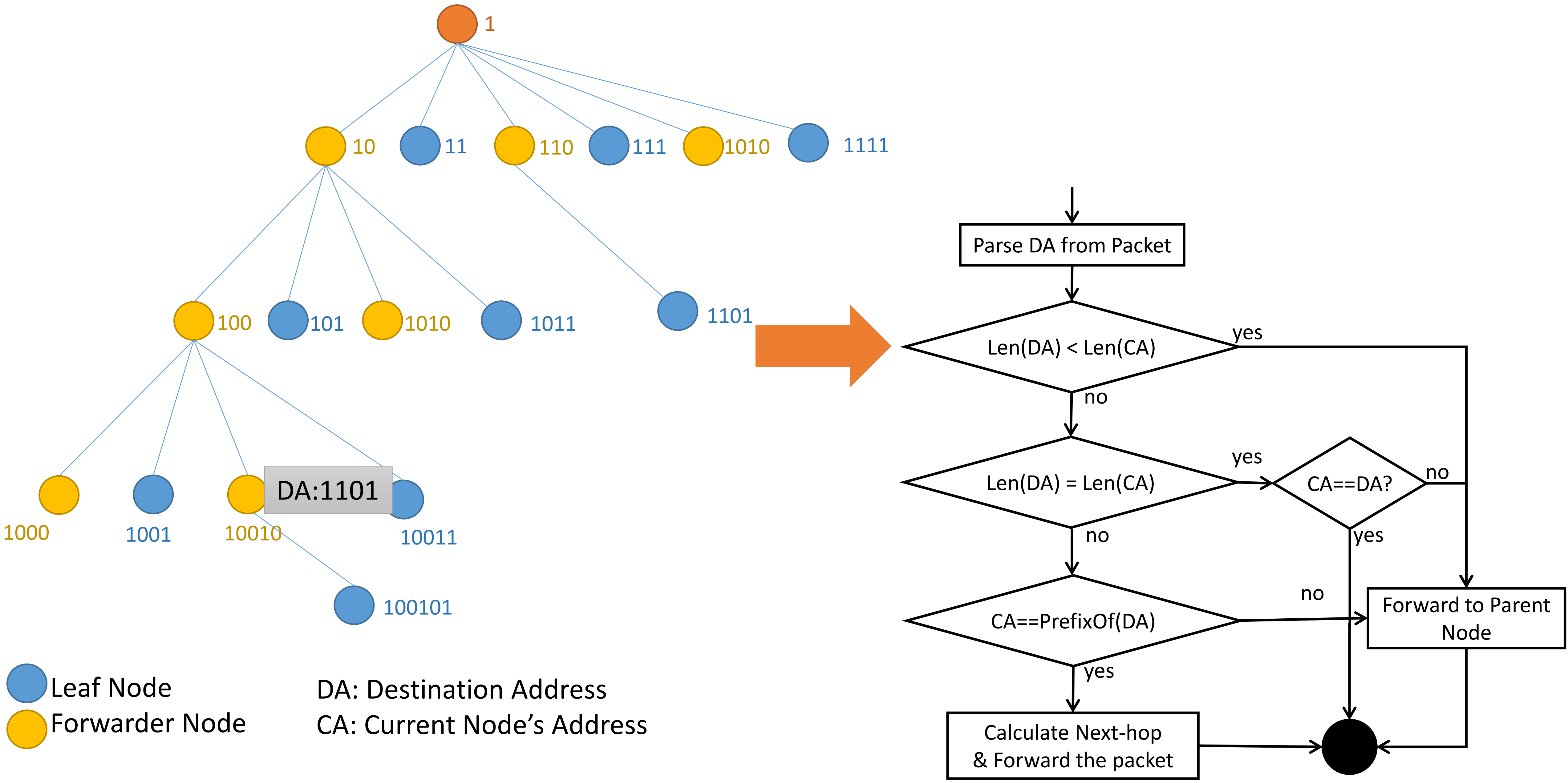
NSA Stateless Forwarding



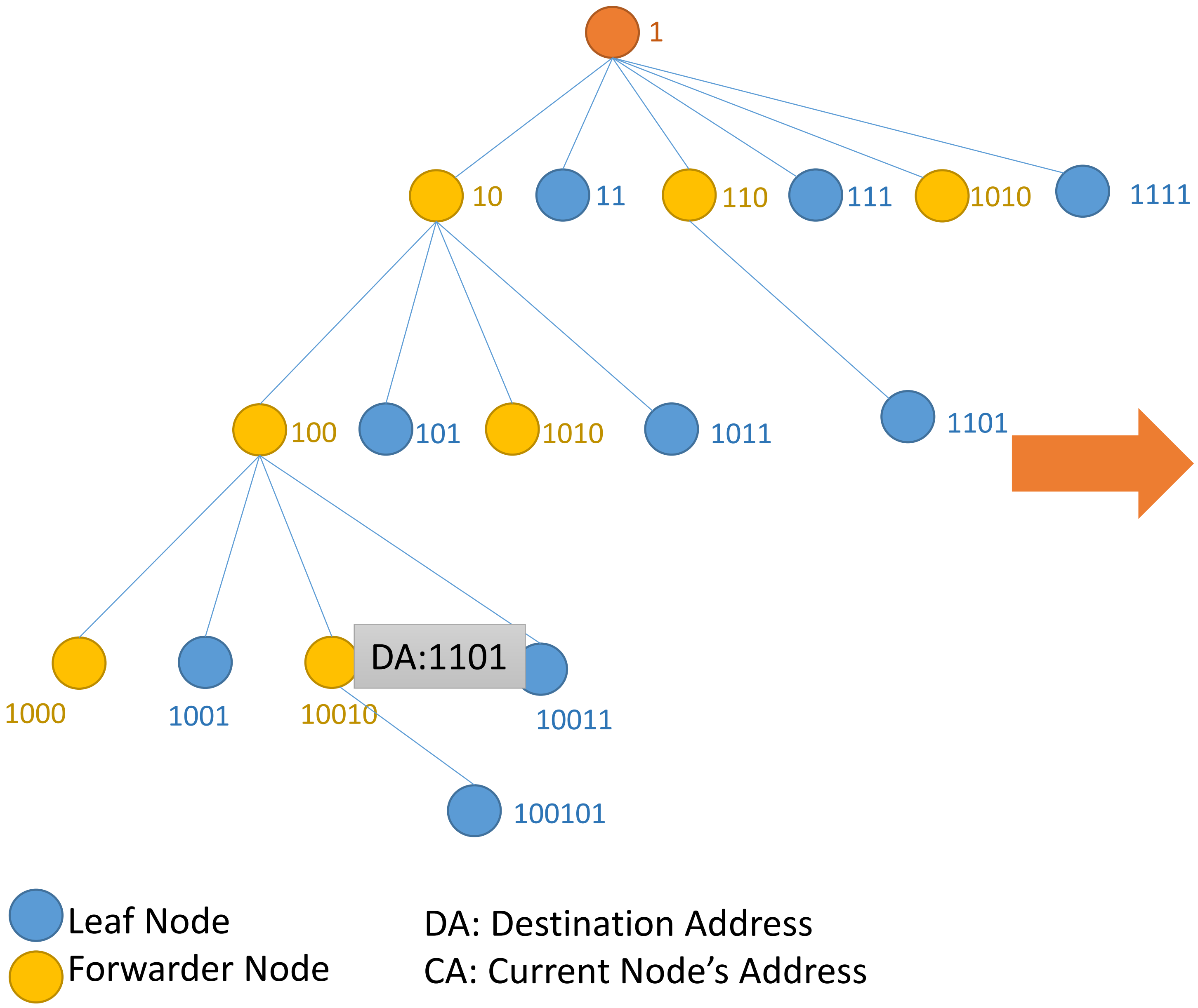
DA: Destination Address
CA: Current Node's Address

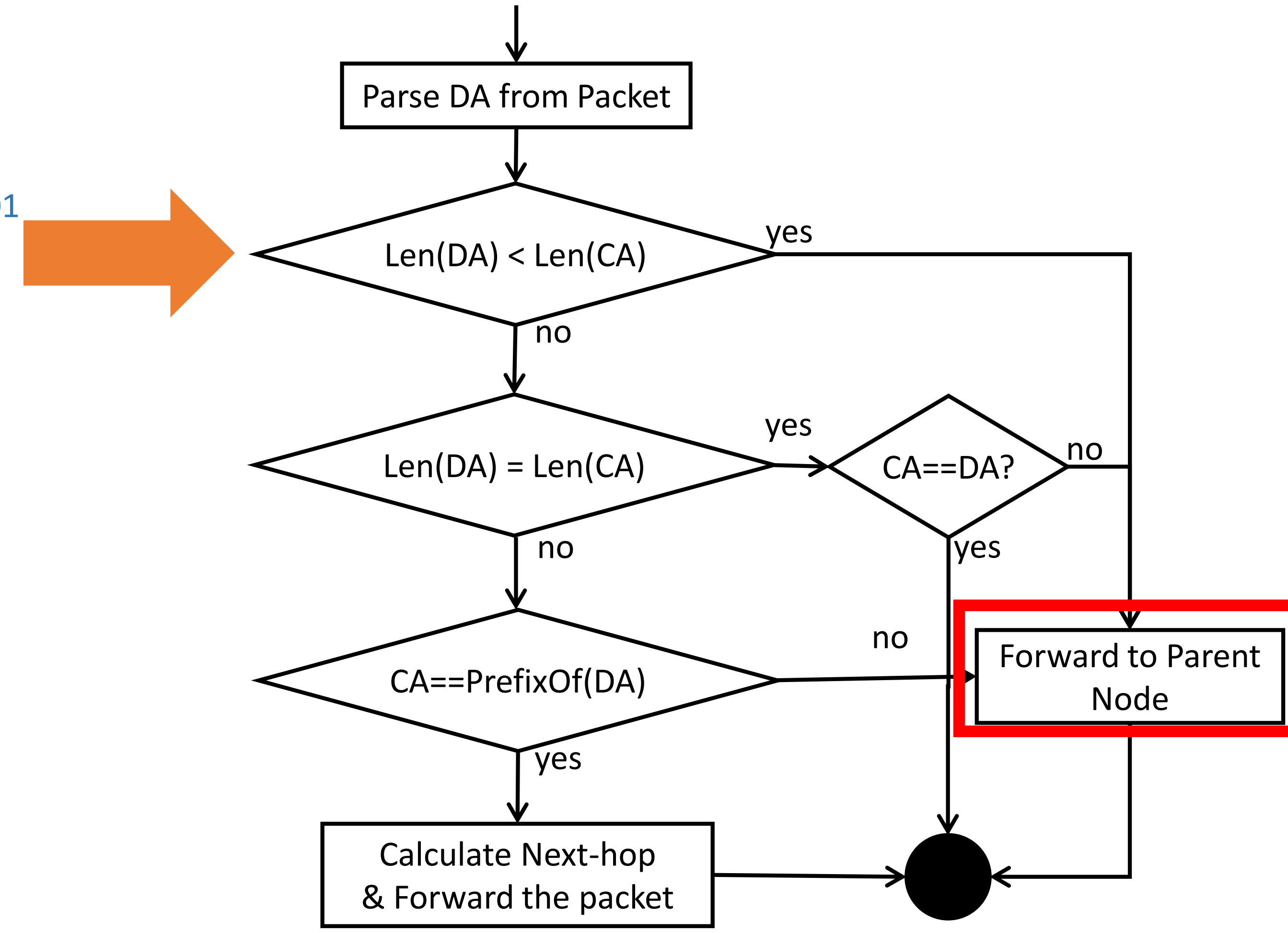


NSA Stateless Forwarding



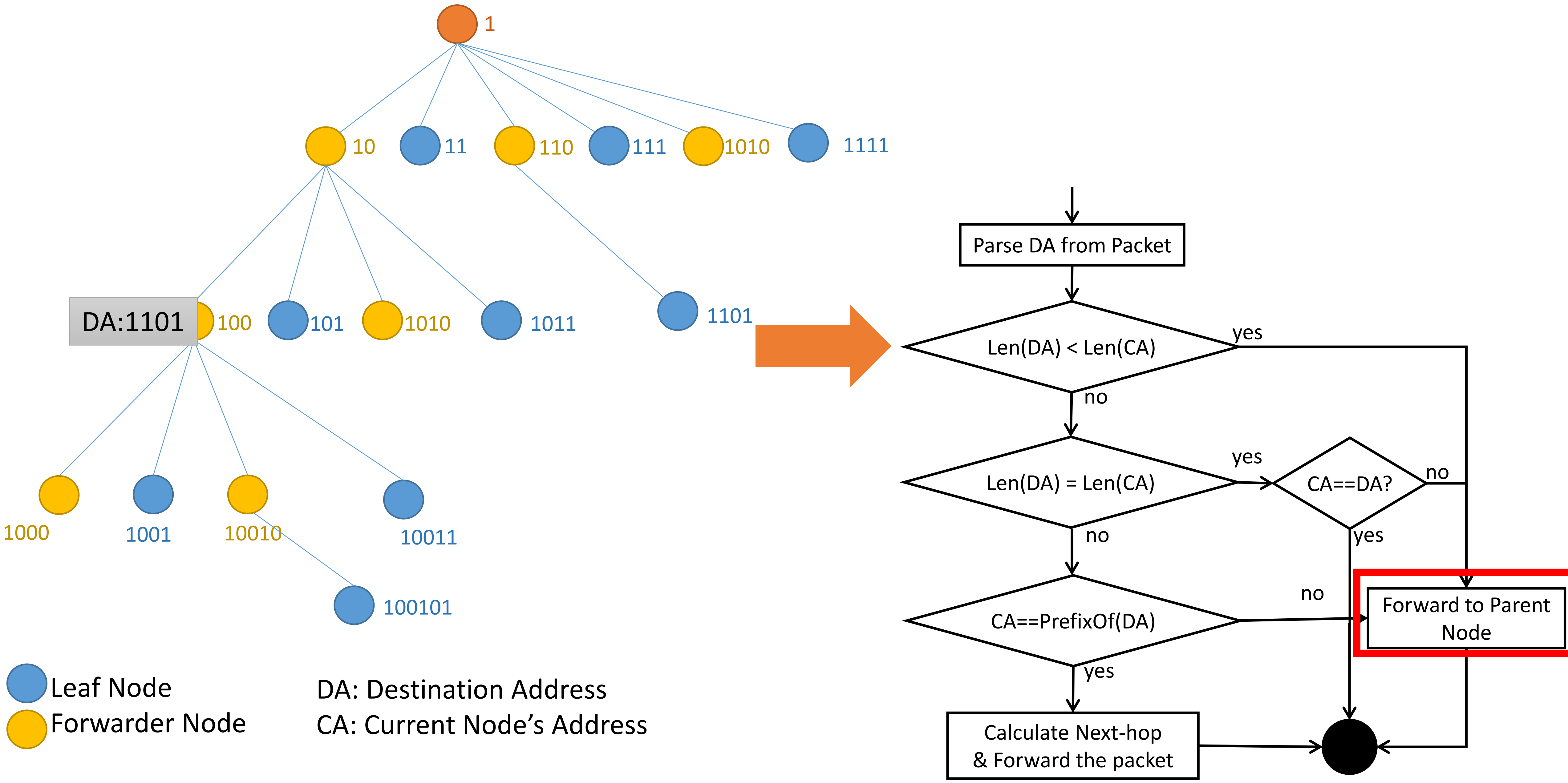
NSA Stateless Forwarding





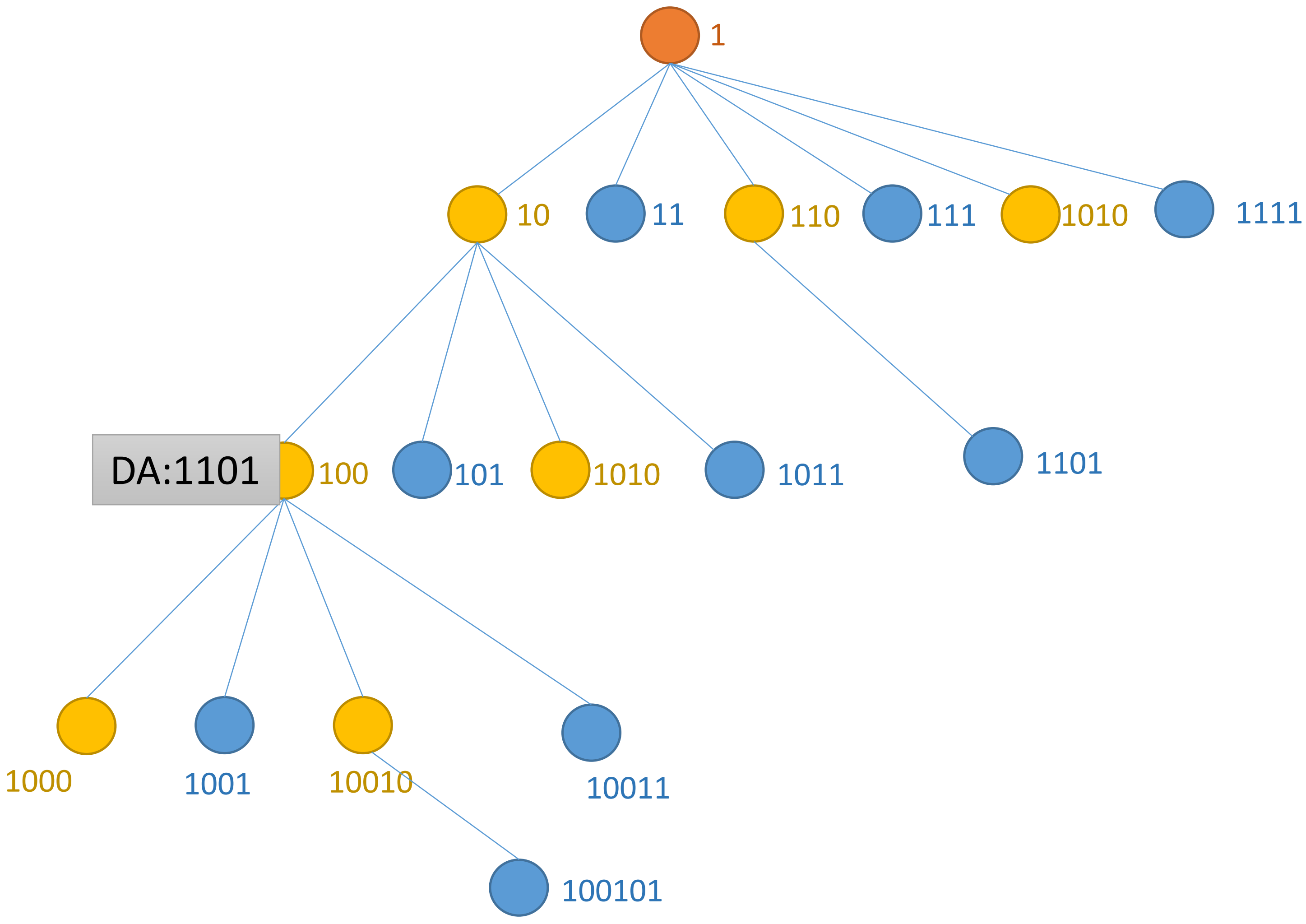
DA: Destination Address
CA: Current Node's Address

NSA Stateless Forwarding



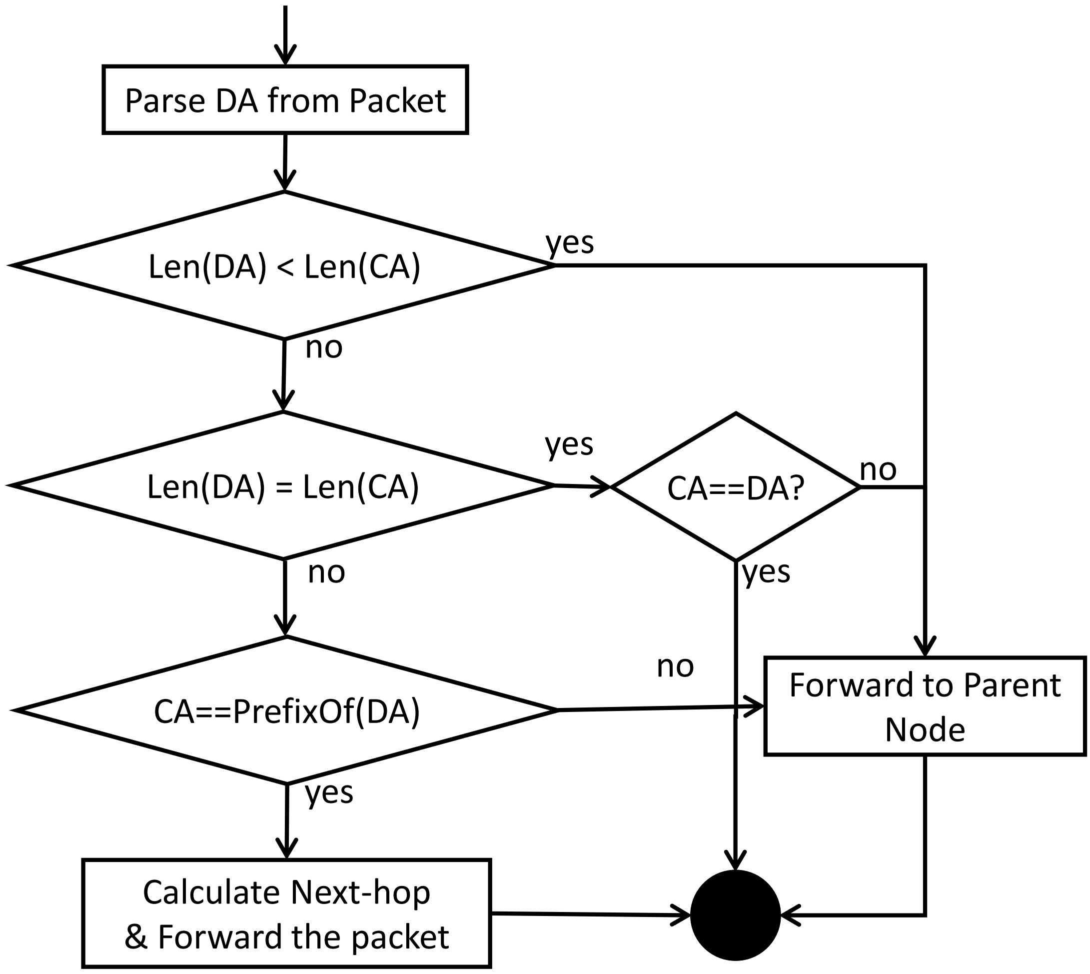
DA: Destination Address
CA: Current Node's Address

NSA Stateless Forwarding

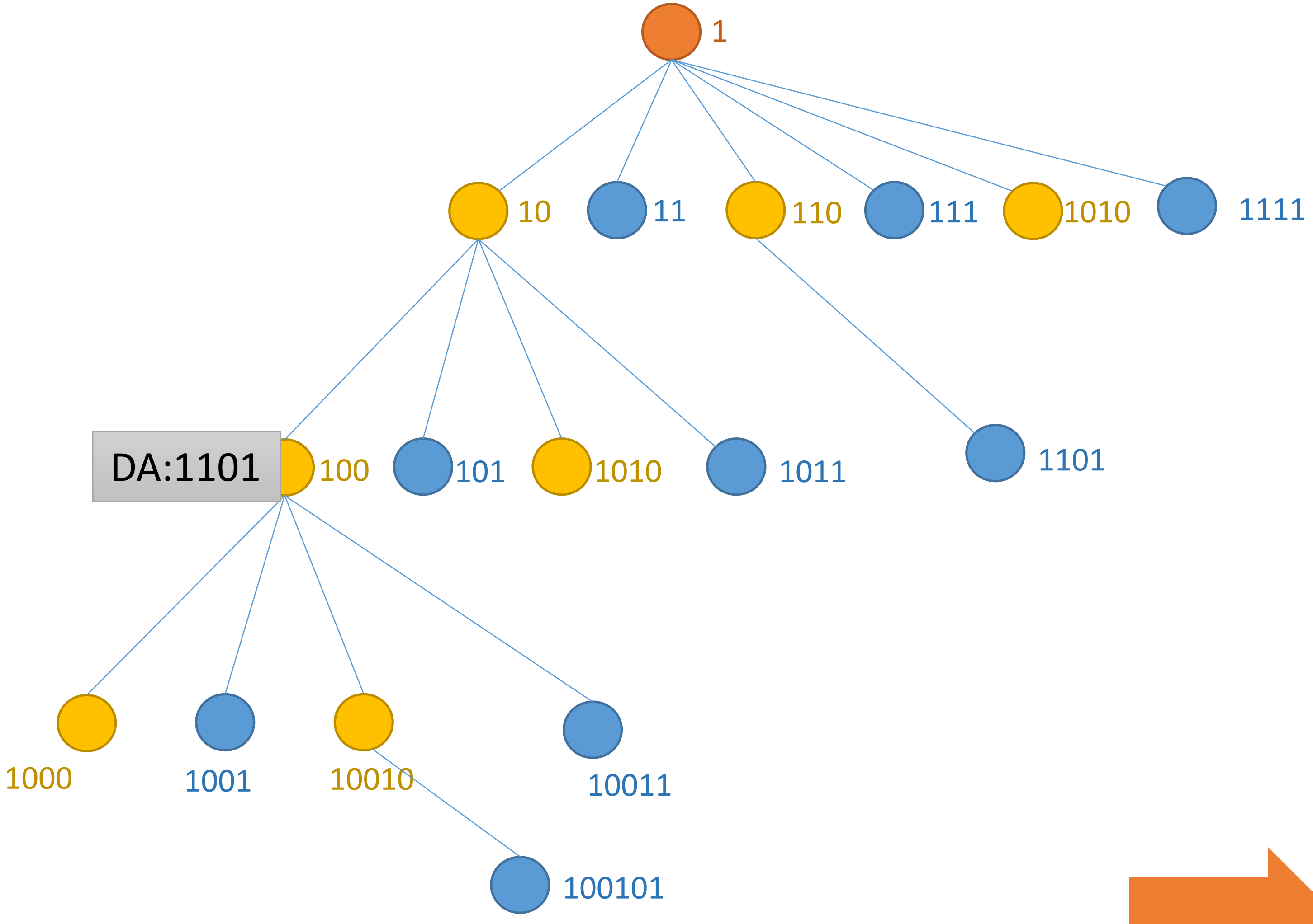


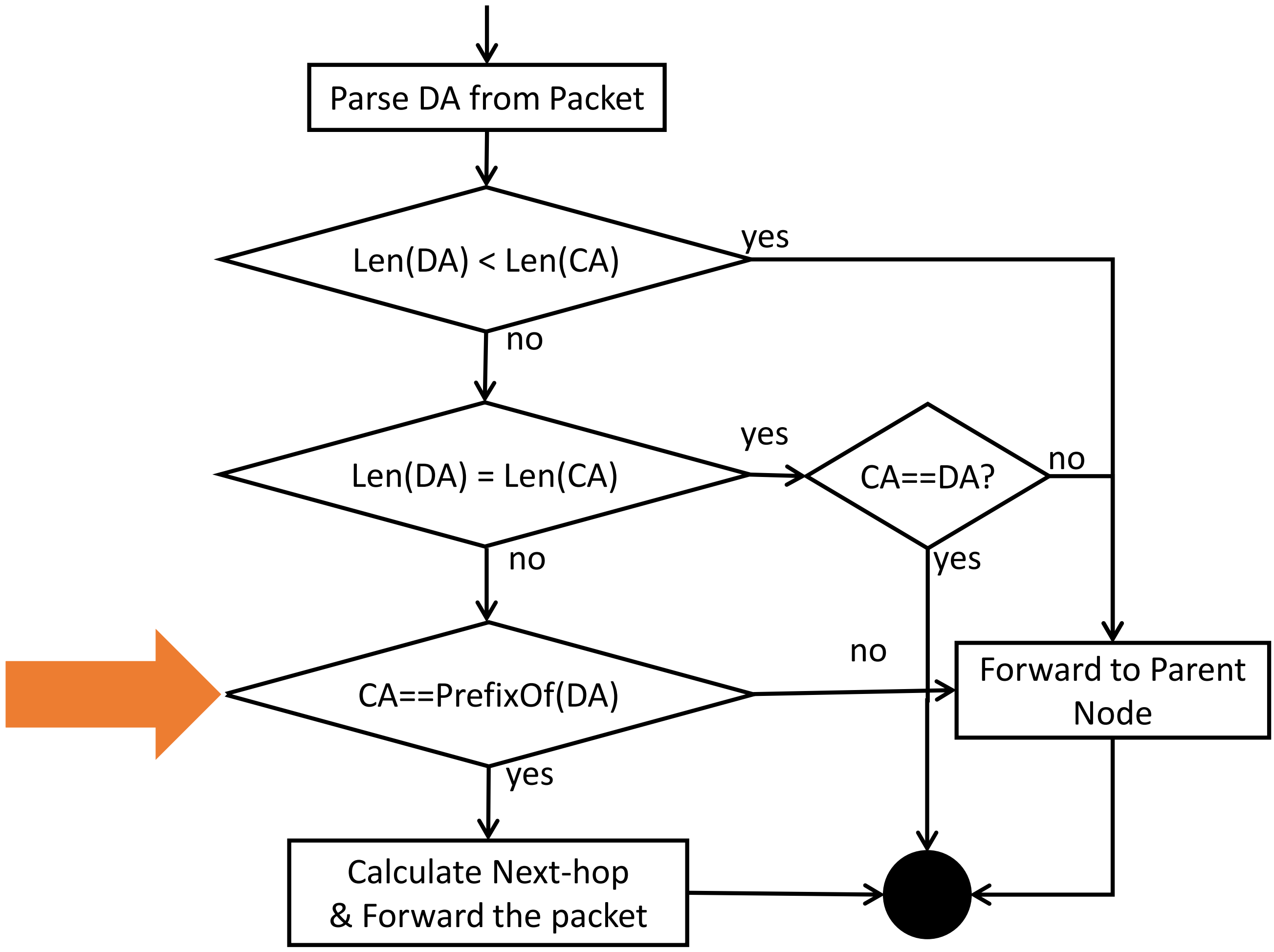
Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node's Address



NSA Stateless Forwarding

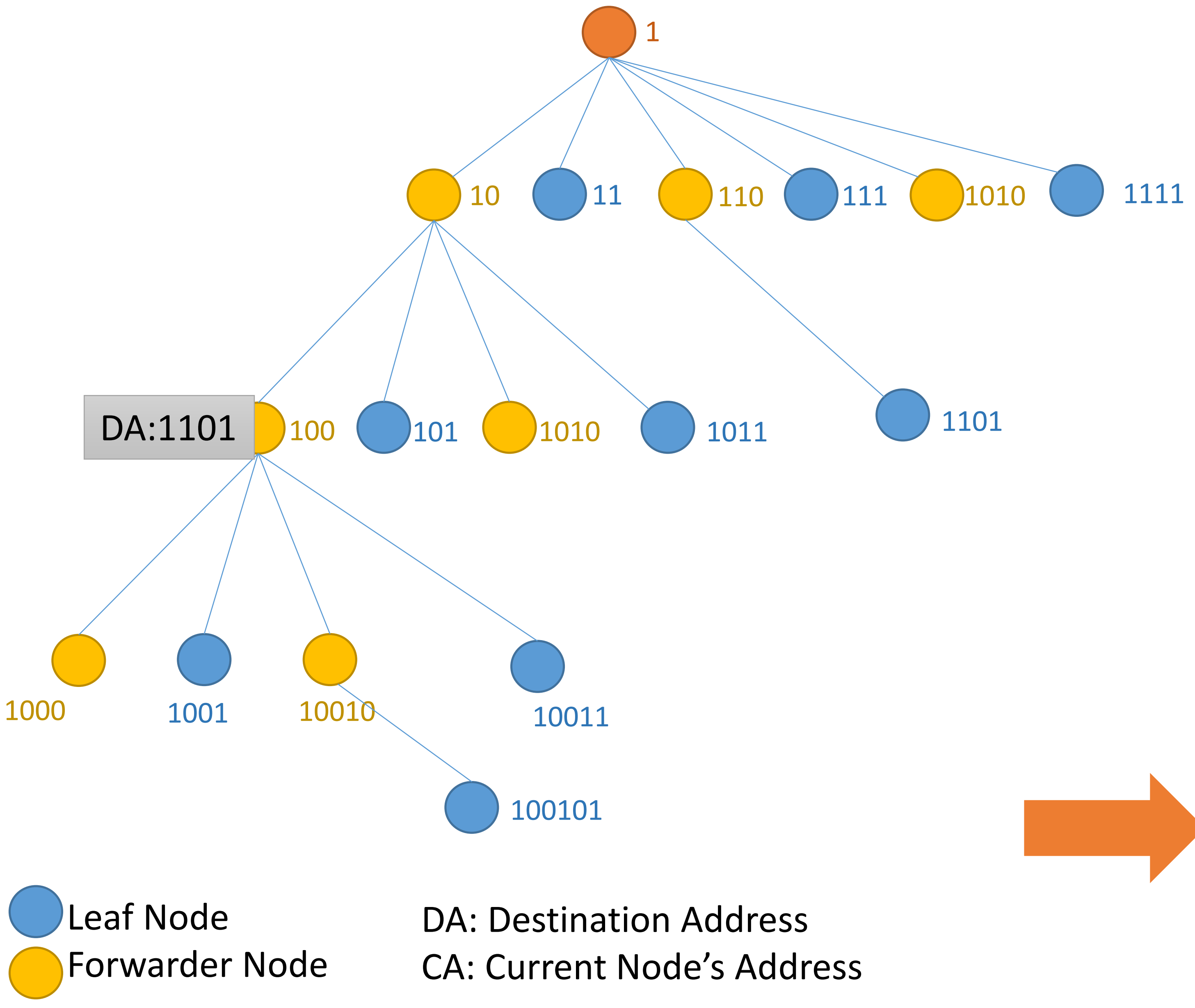


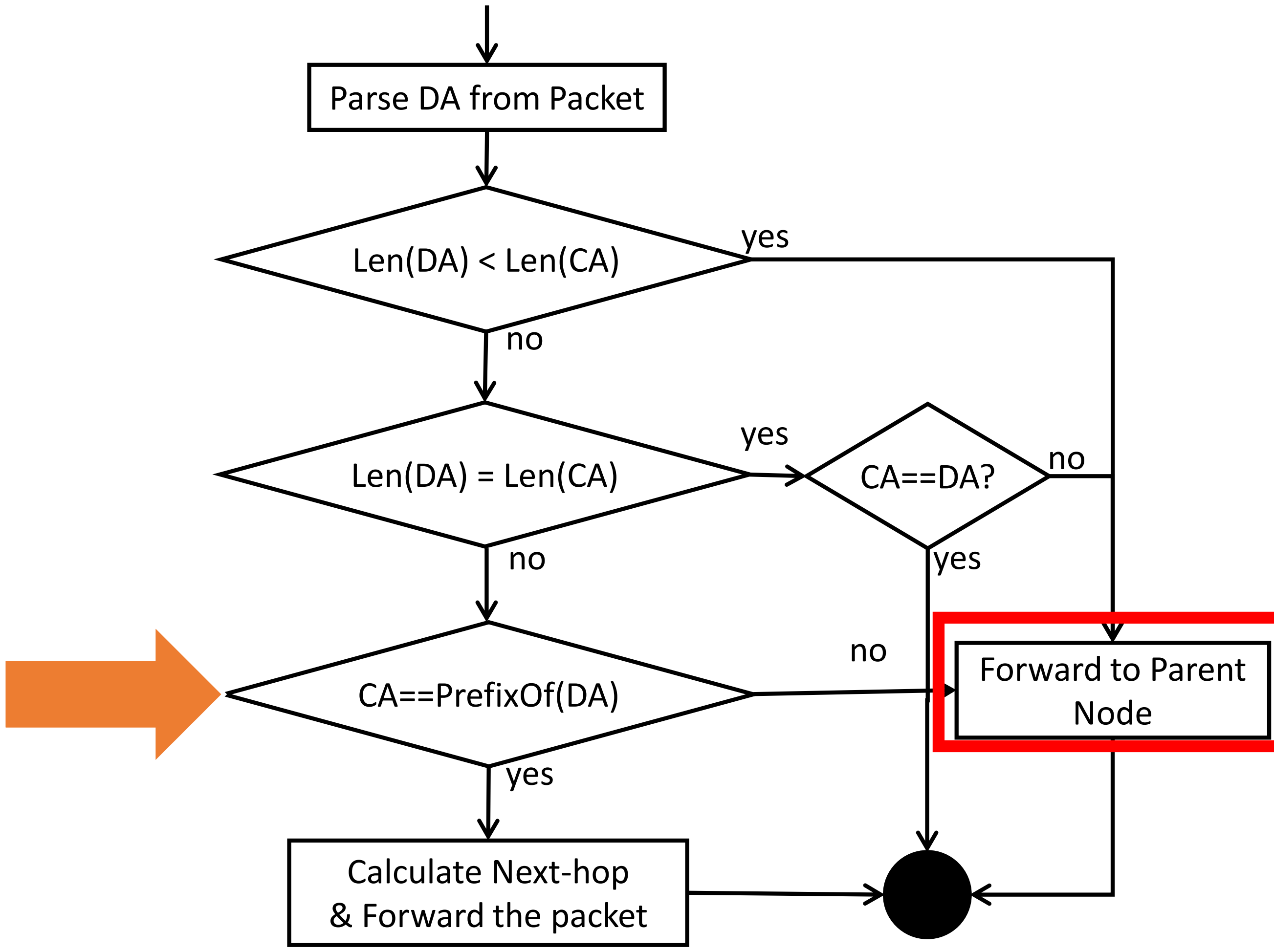


Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node's Address

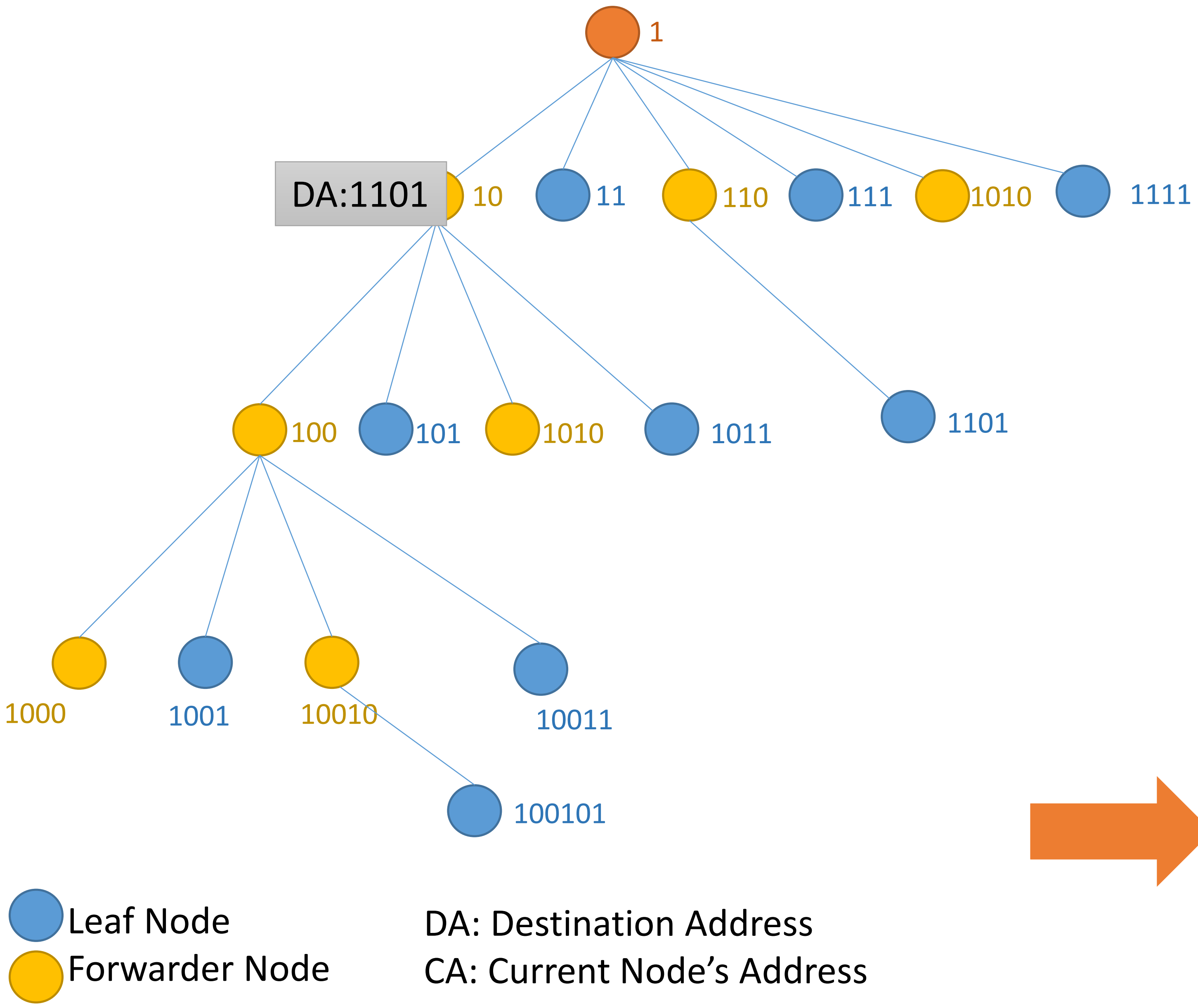
NSA Stateless Forwarding



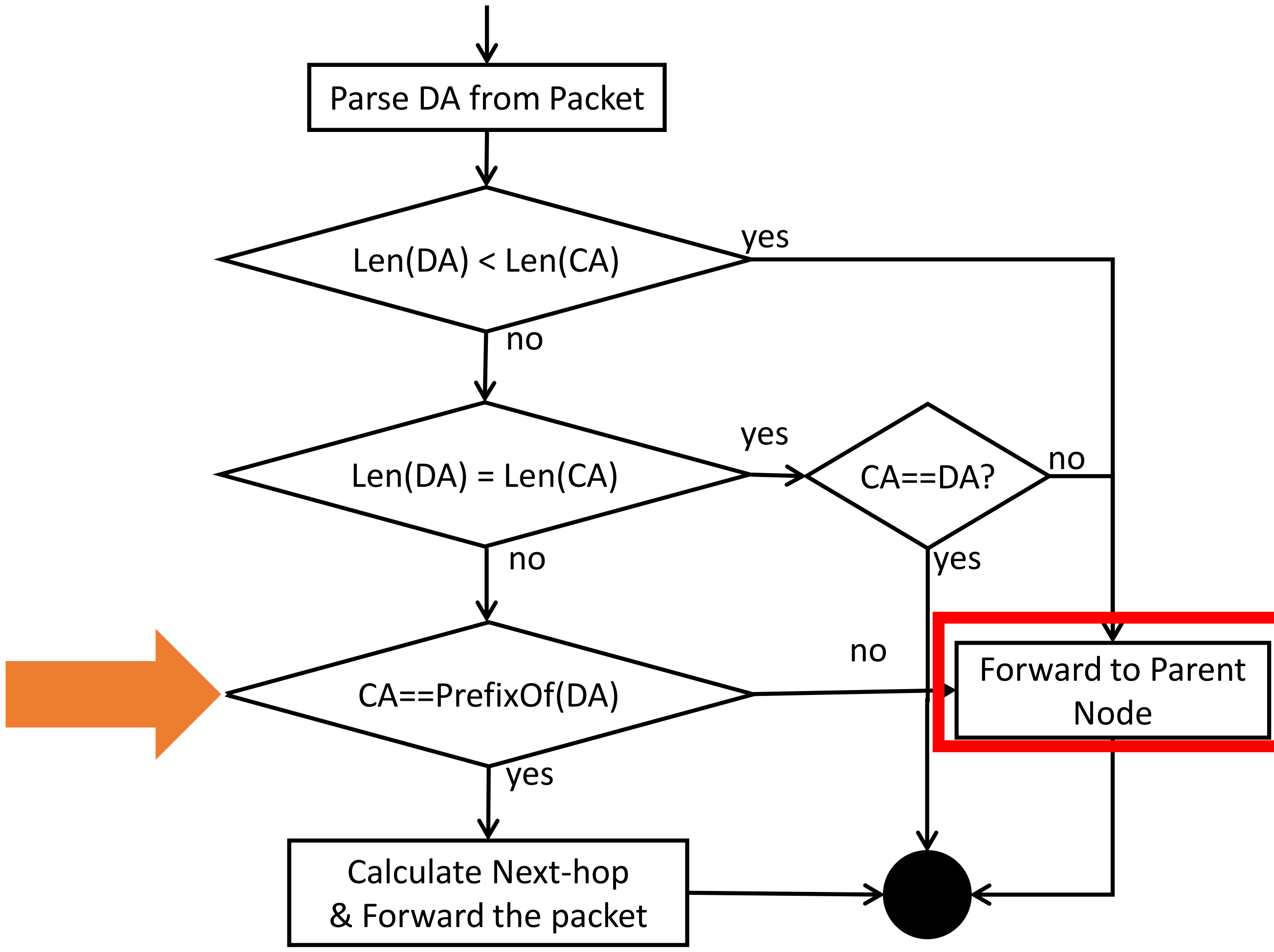


DA: Destination Address
CA: Current Node's Address

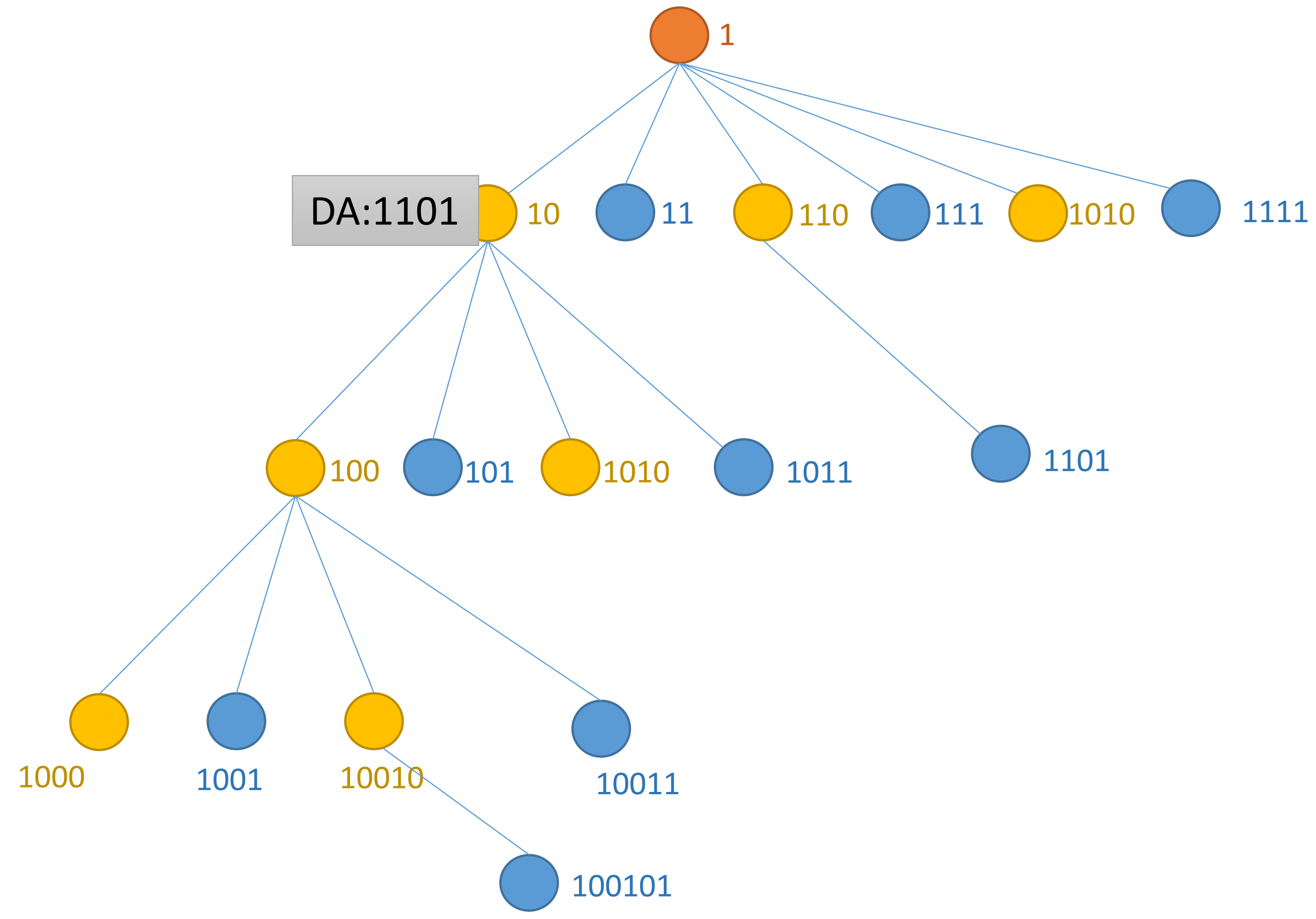
NSA Stateless Forwarding

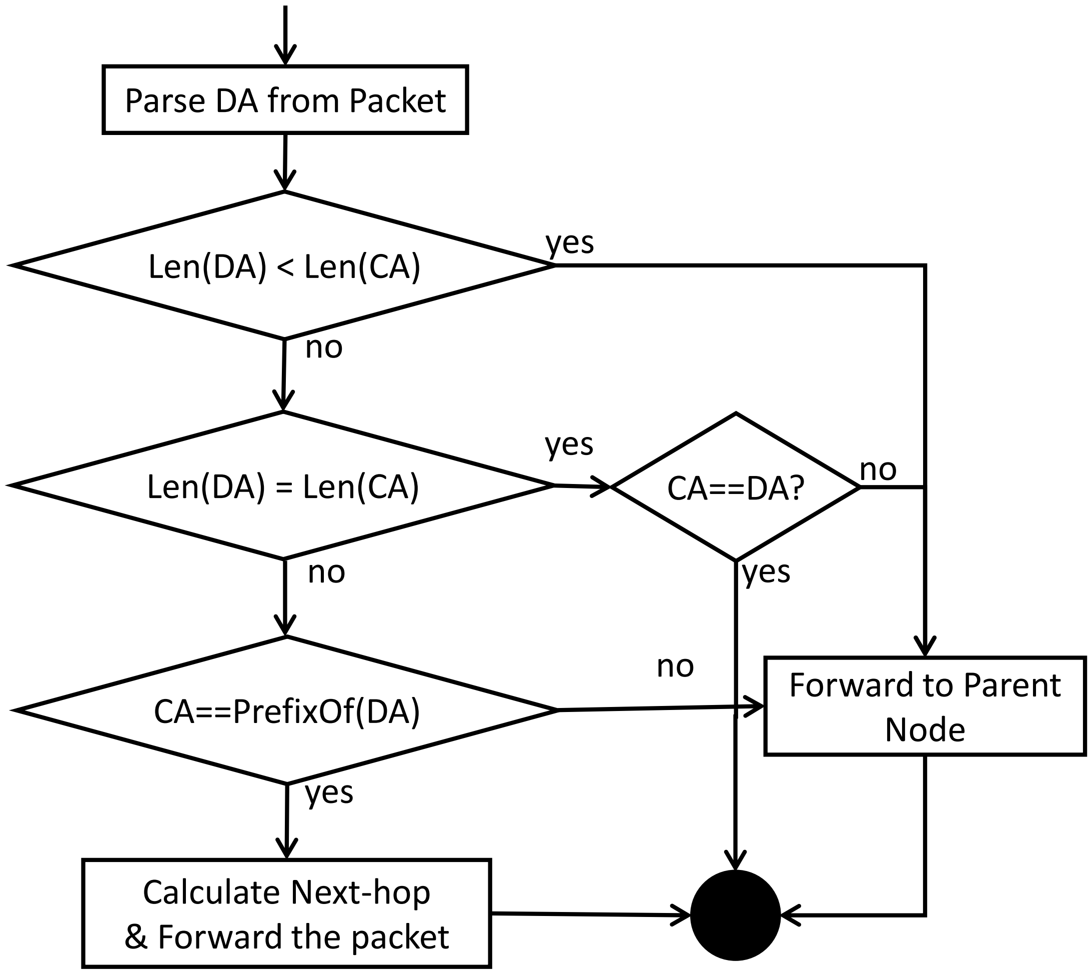


DA: Destination Address
CA: Current Node's Address



NSA Stateless Forwarding

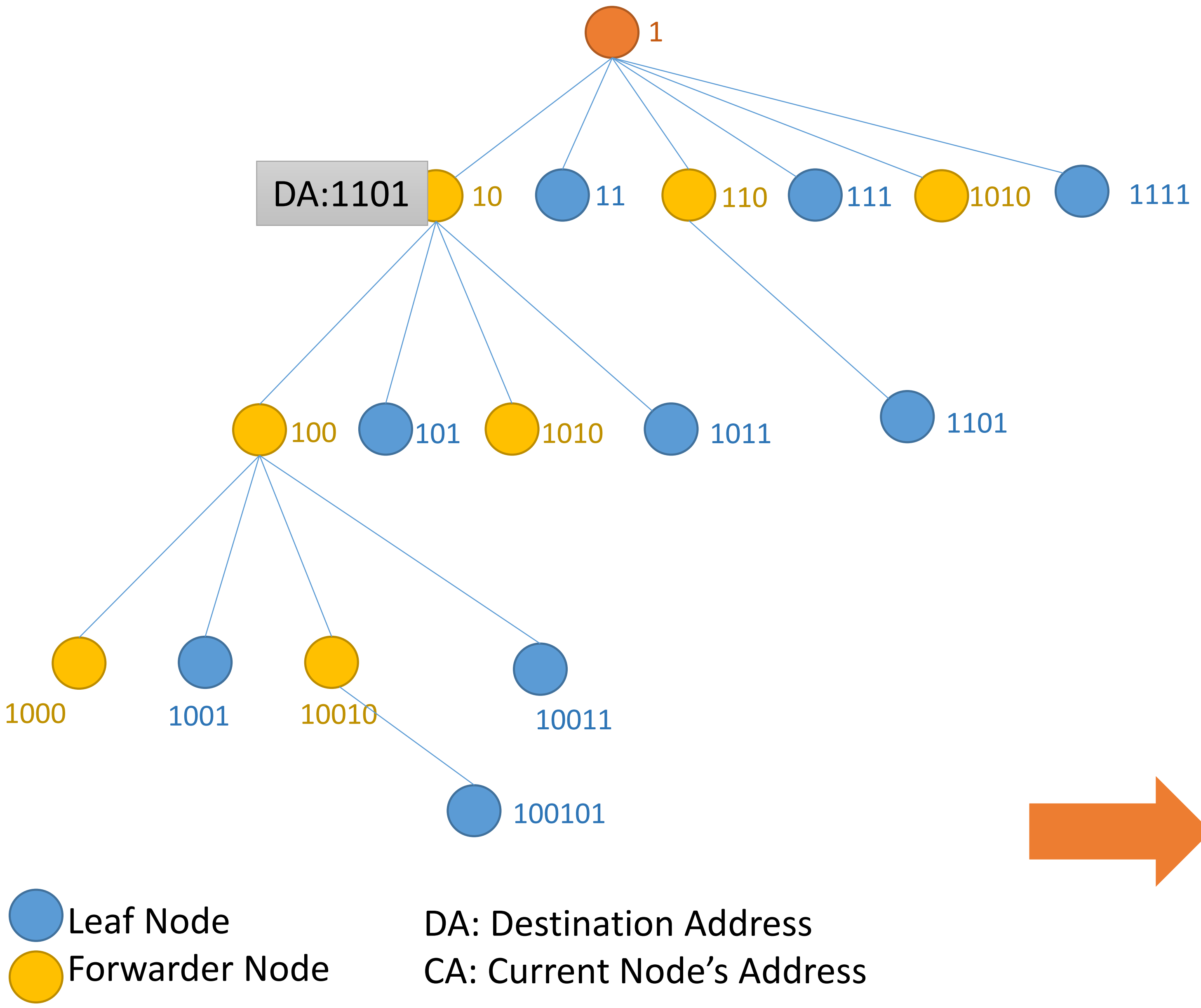




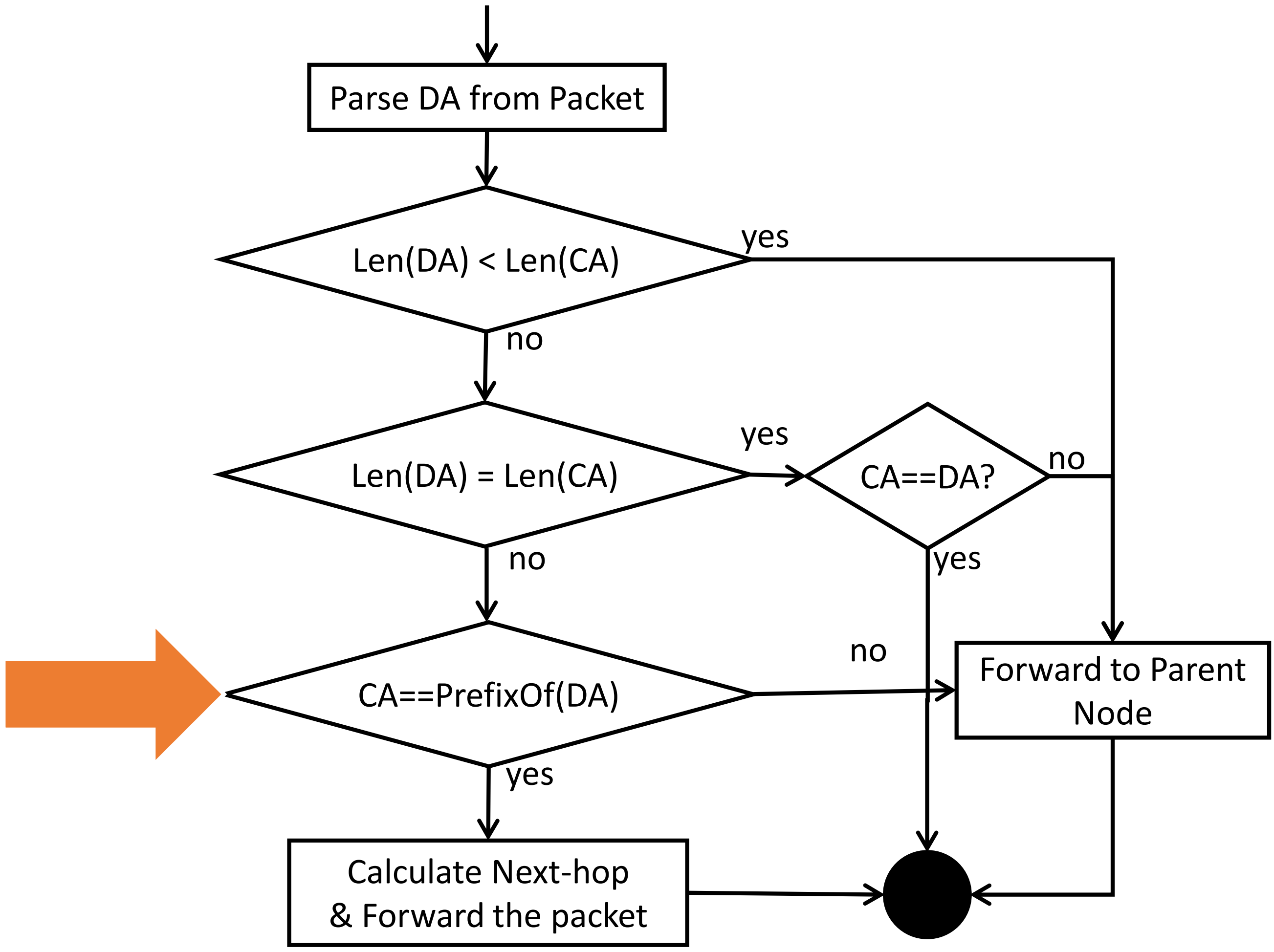
Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node's Address

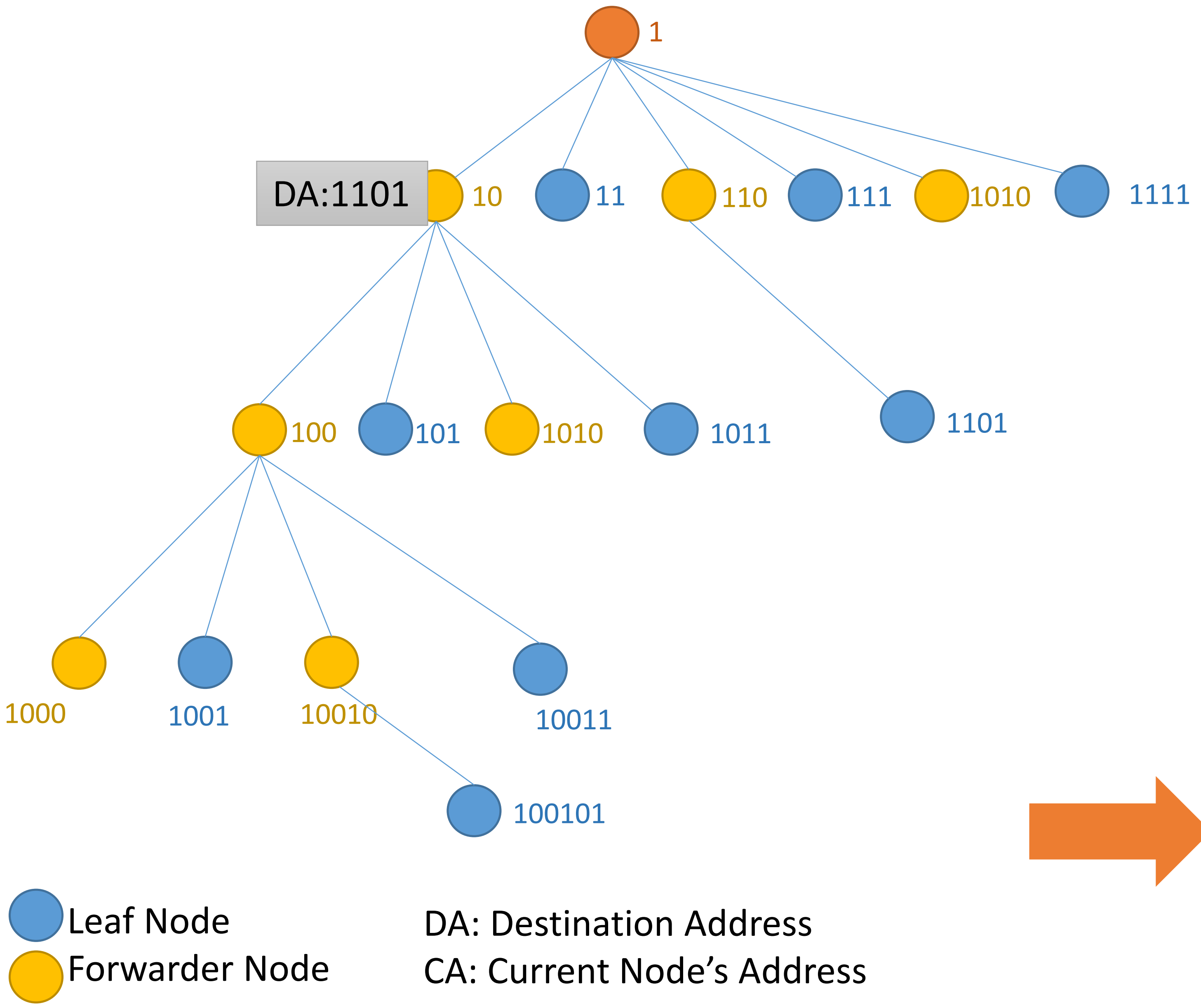
NSA Stateless Forwarding



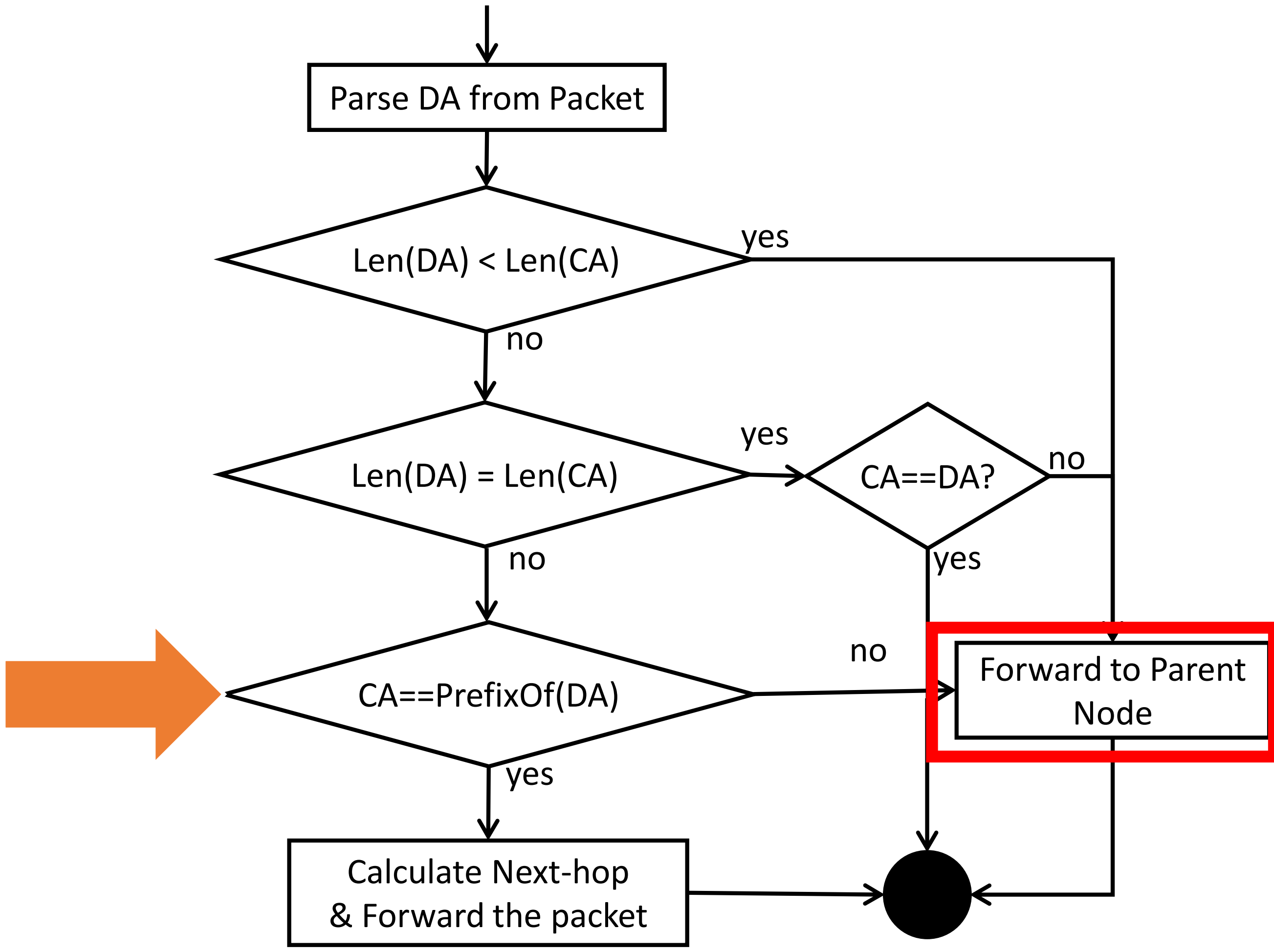
DA: Destination Address
CA: Current Node's Address



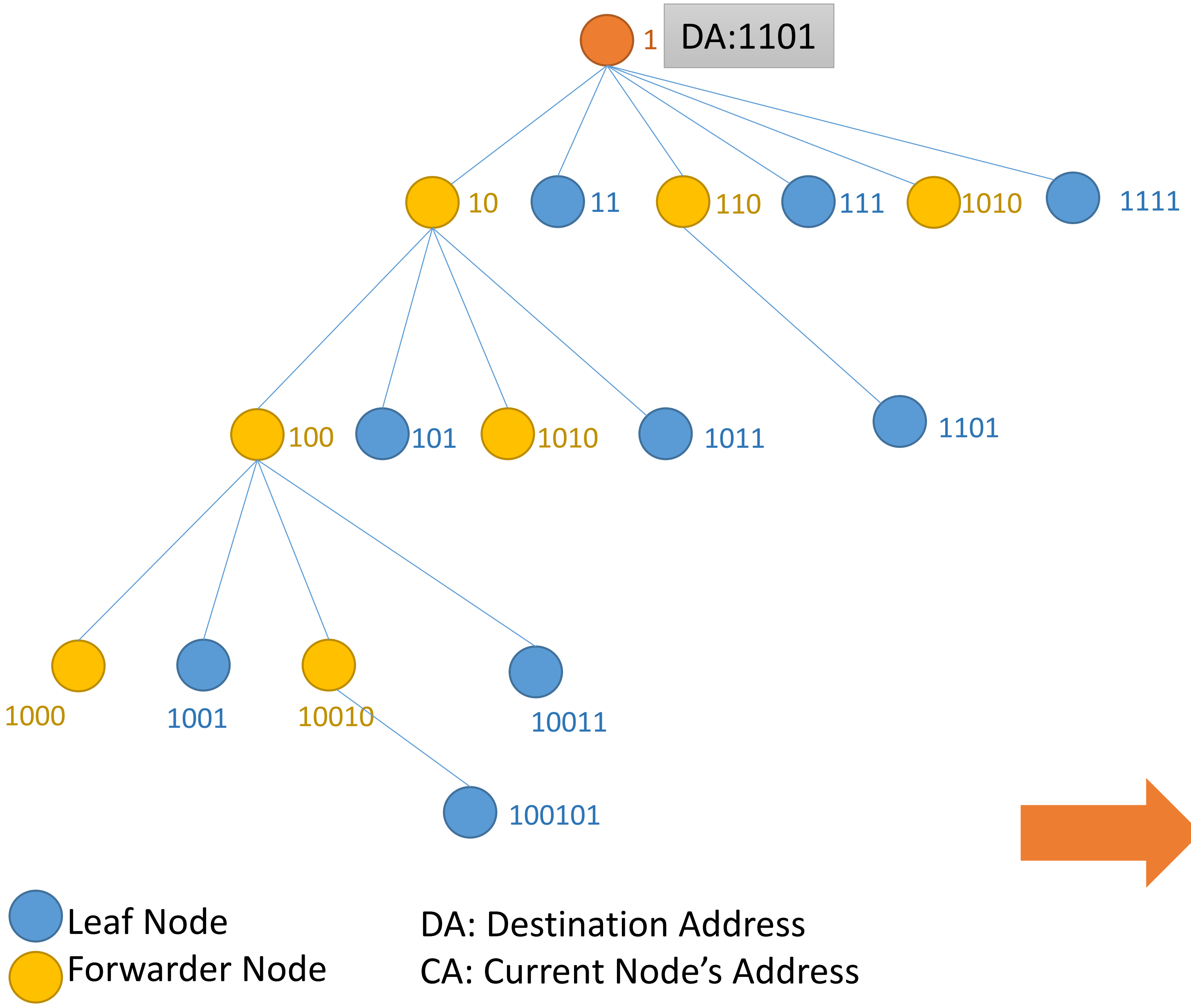
NSA Stateless Forwarding



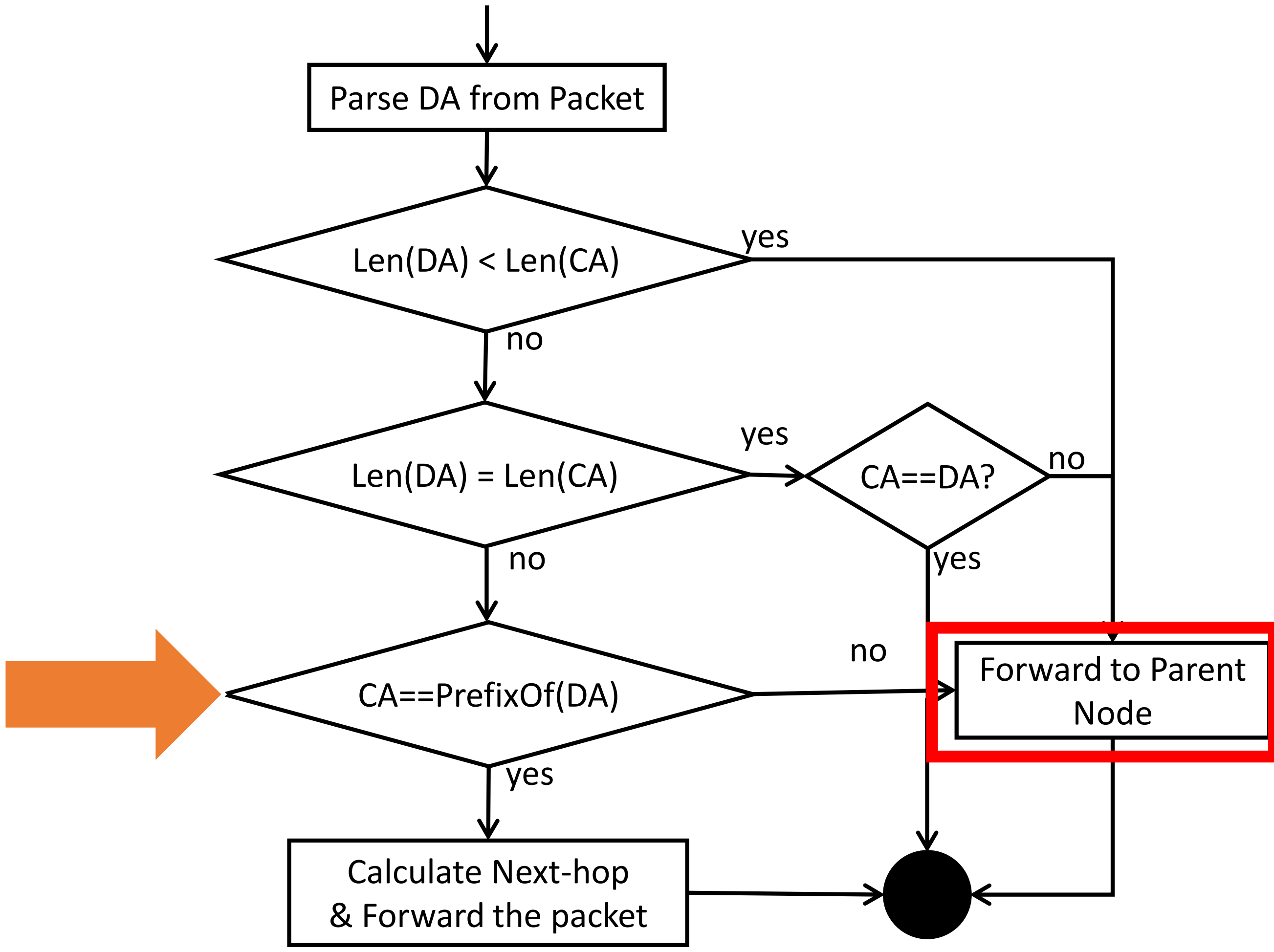
DA: Destination Address
CA: Current Node's Address



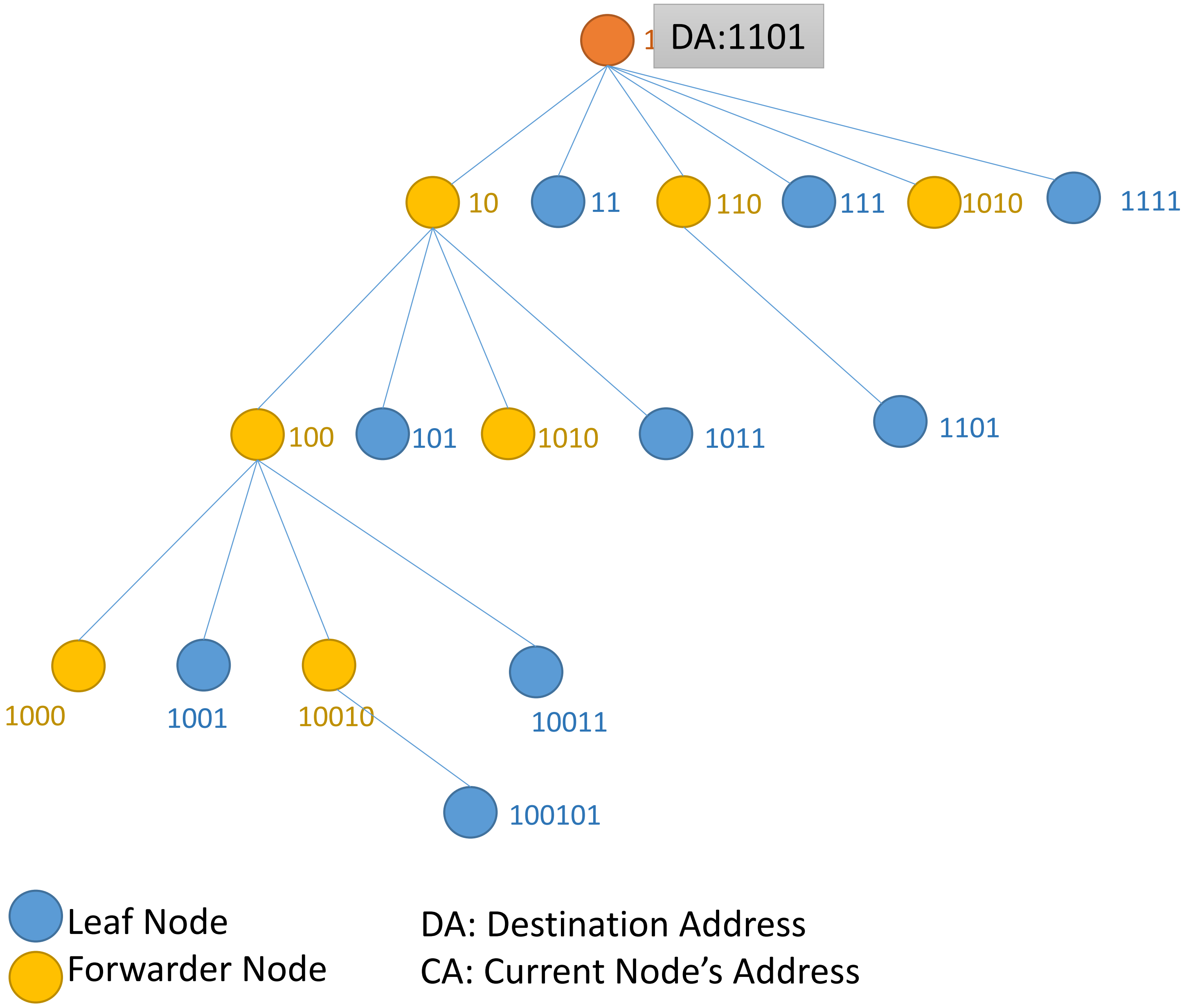
NSA Stateless Forwarding



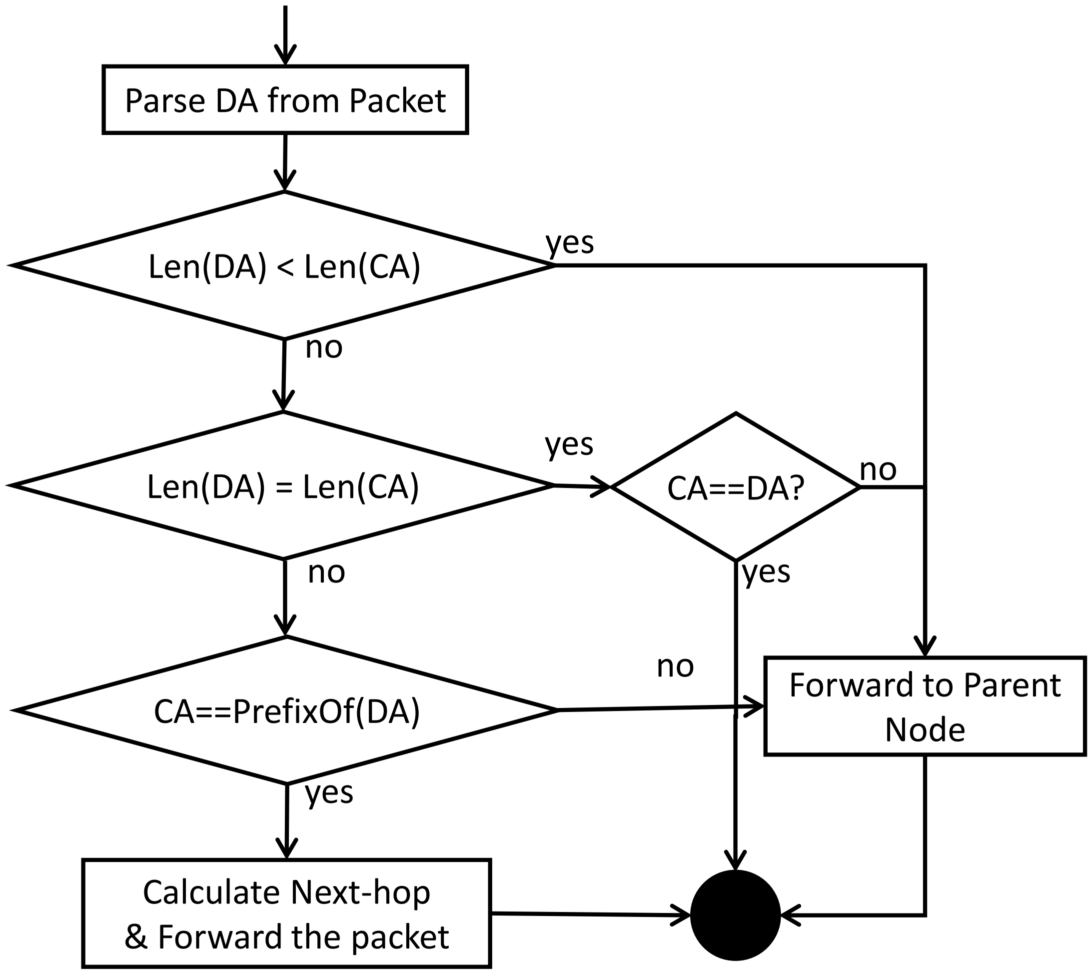
DA: Destination Address
CA: Current Node's Address



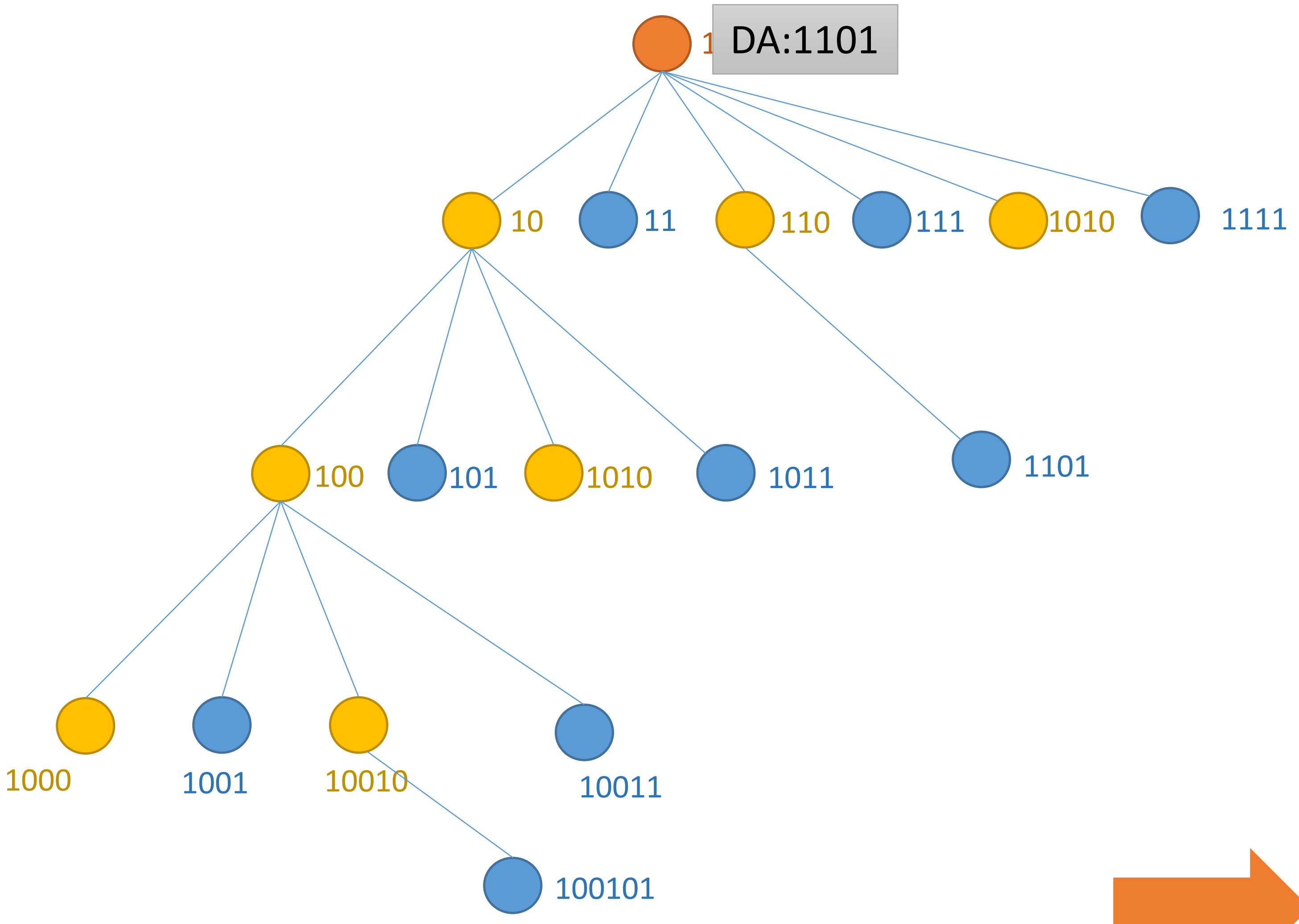
NSA Stateless Forwarding



DA: Destination Address
CA: Current Node's Address

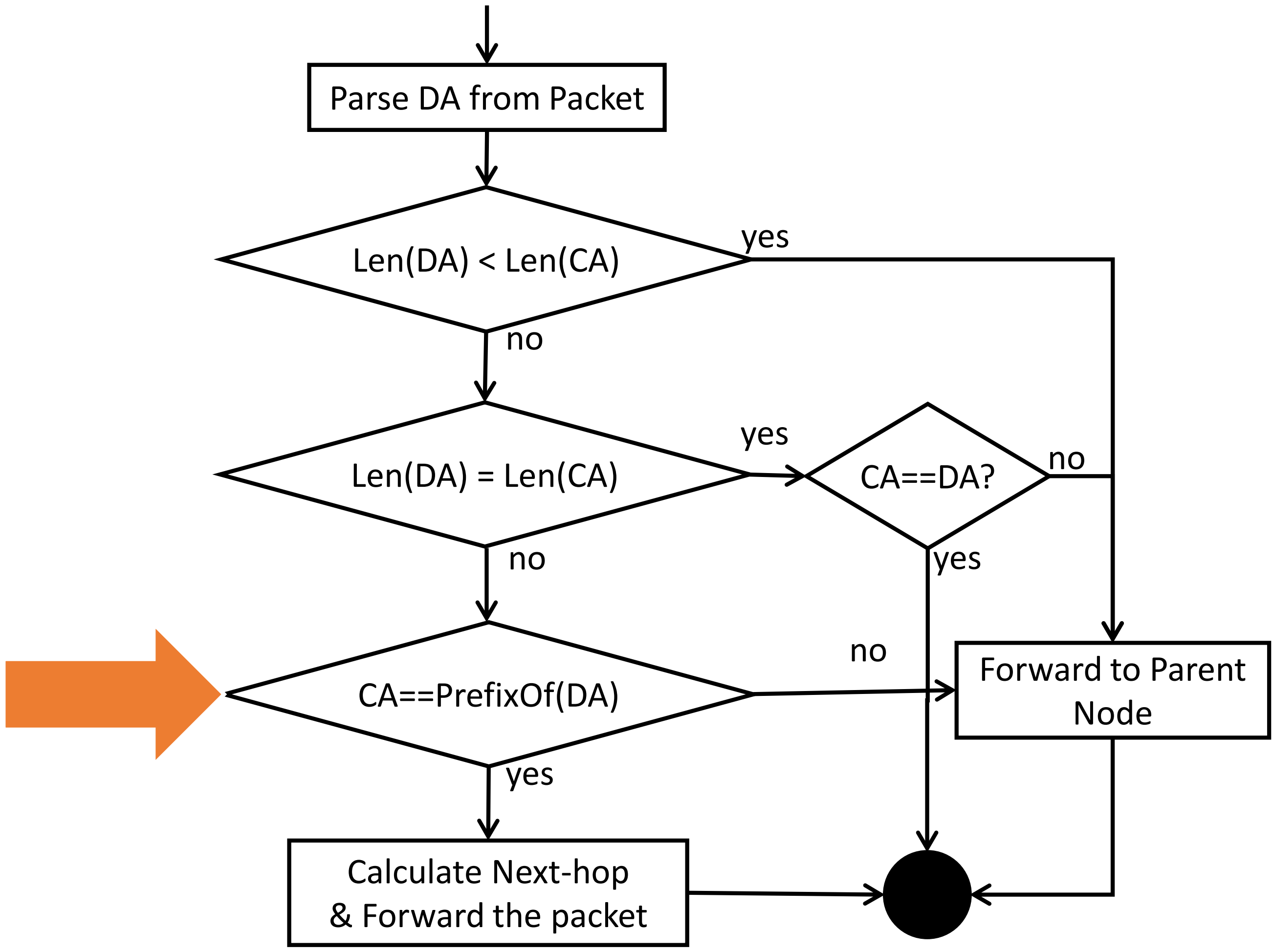


NSA Stateless Forwarding

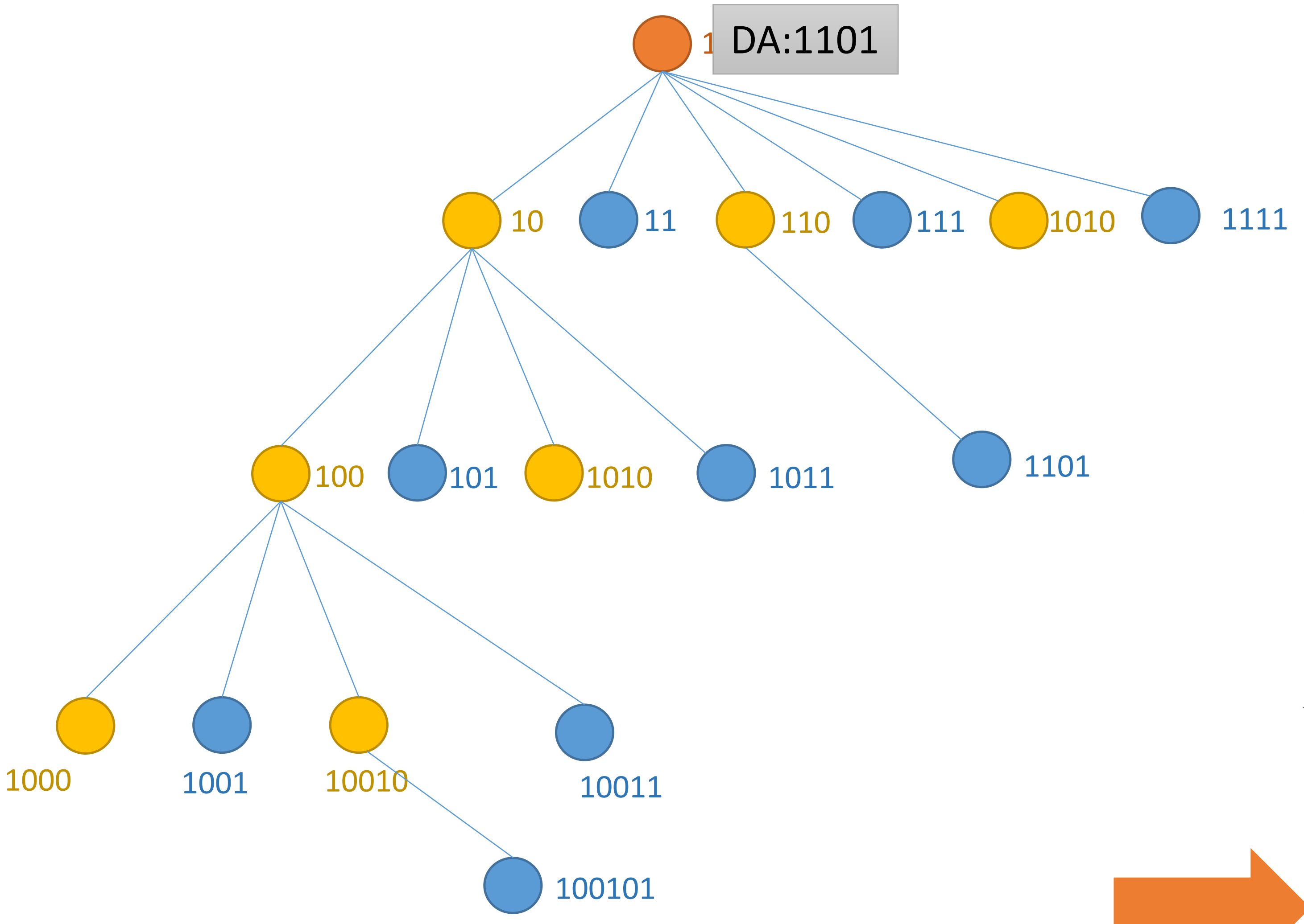


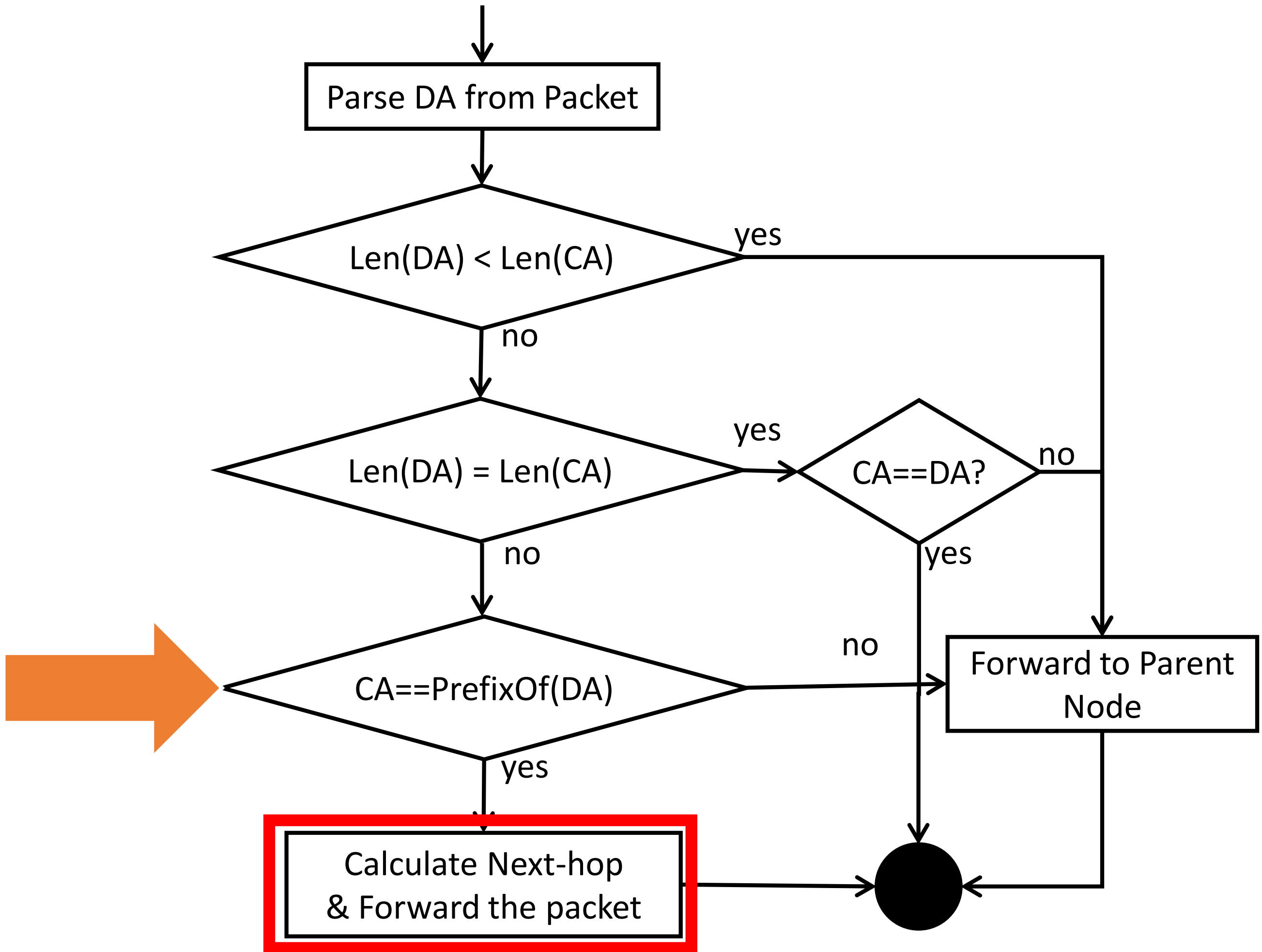
Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node's Address



NSA Stateless Forwarding

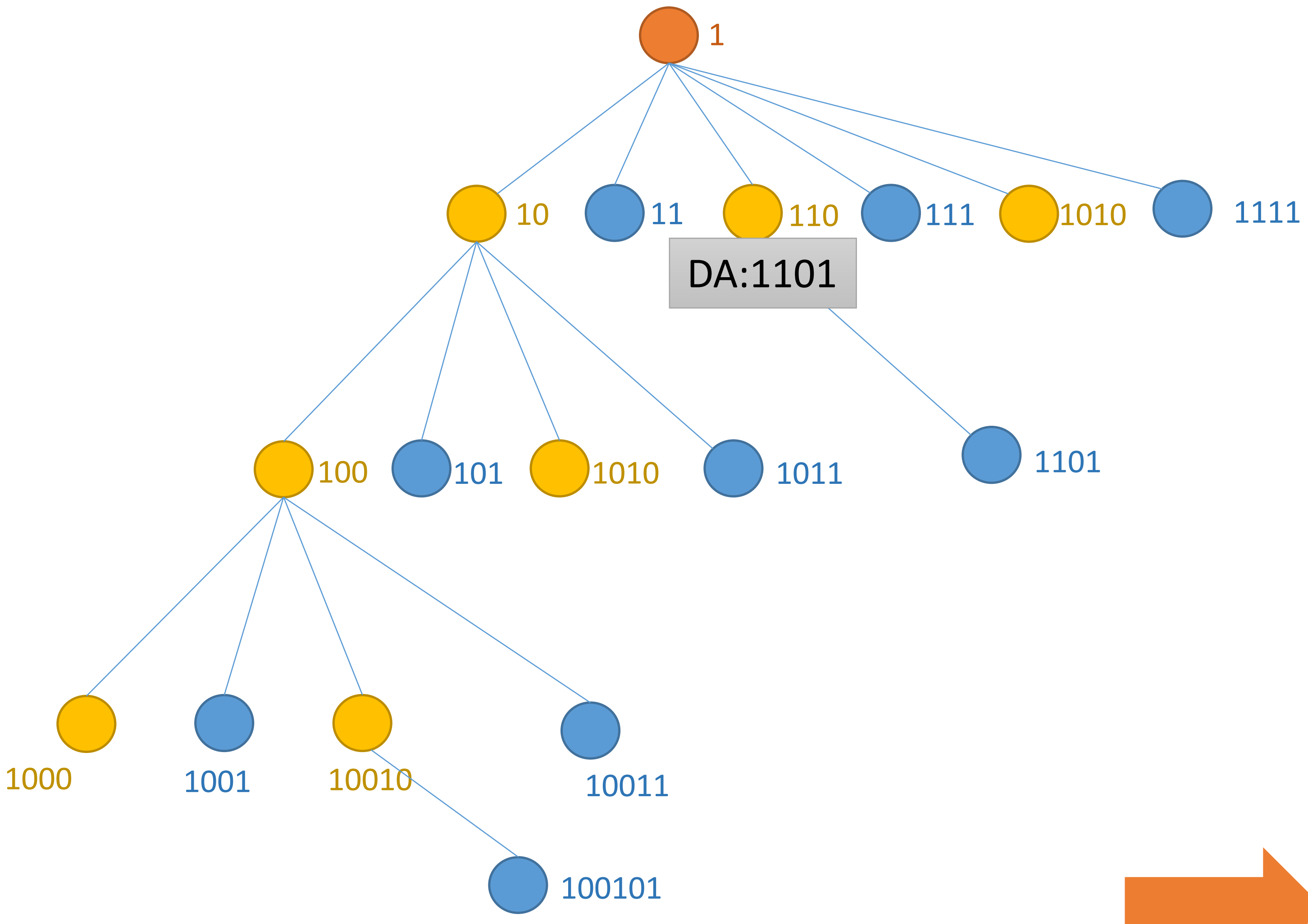


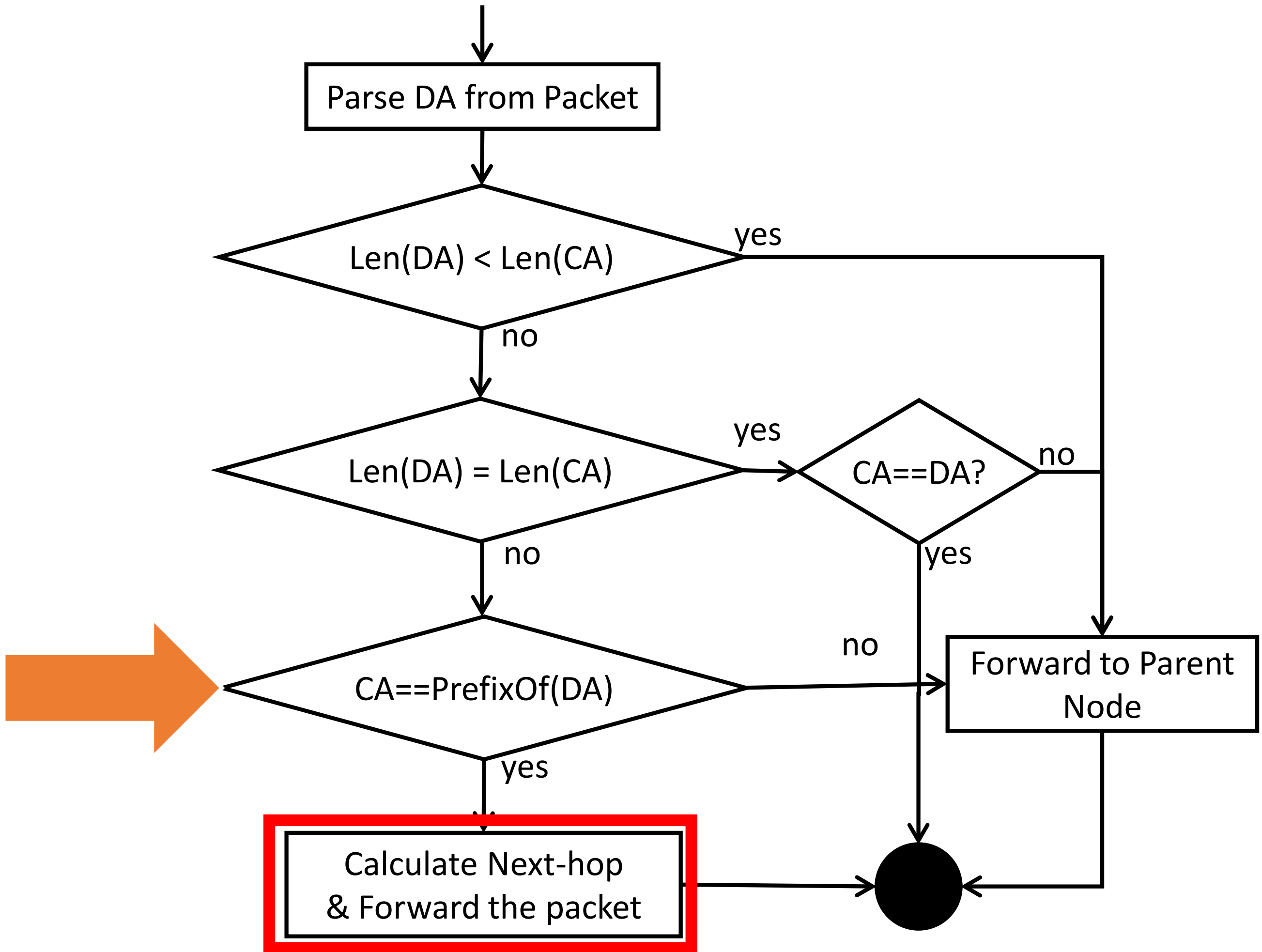


Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node's Address

NSA Stateless Forwarding

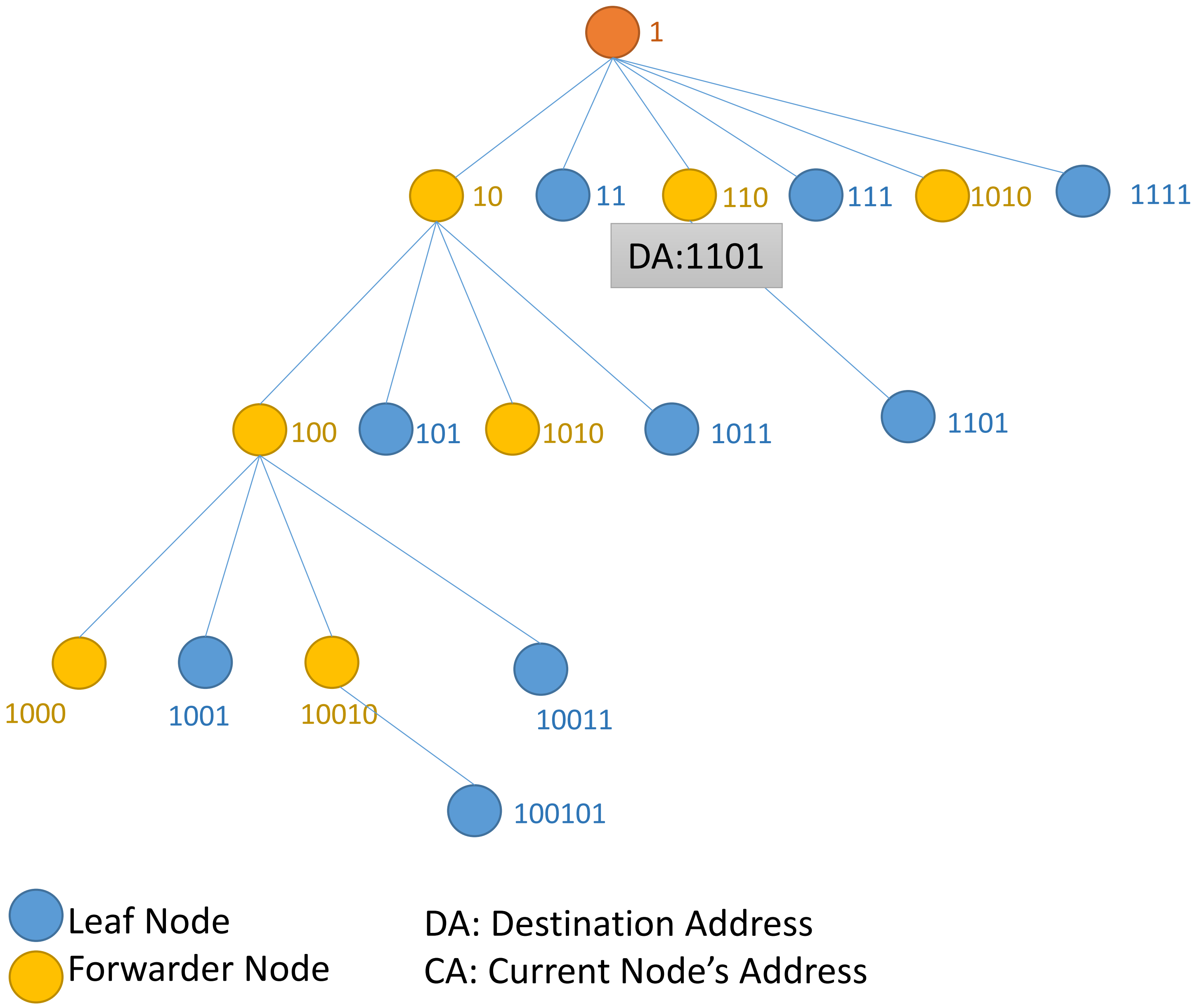




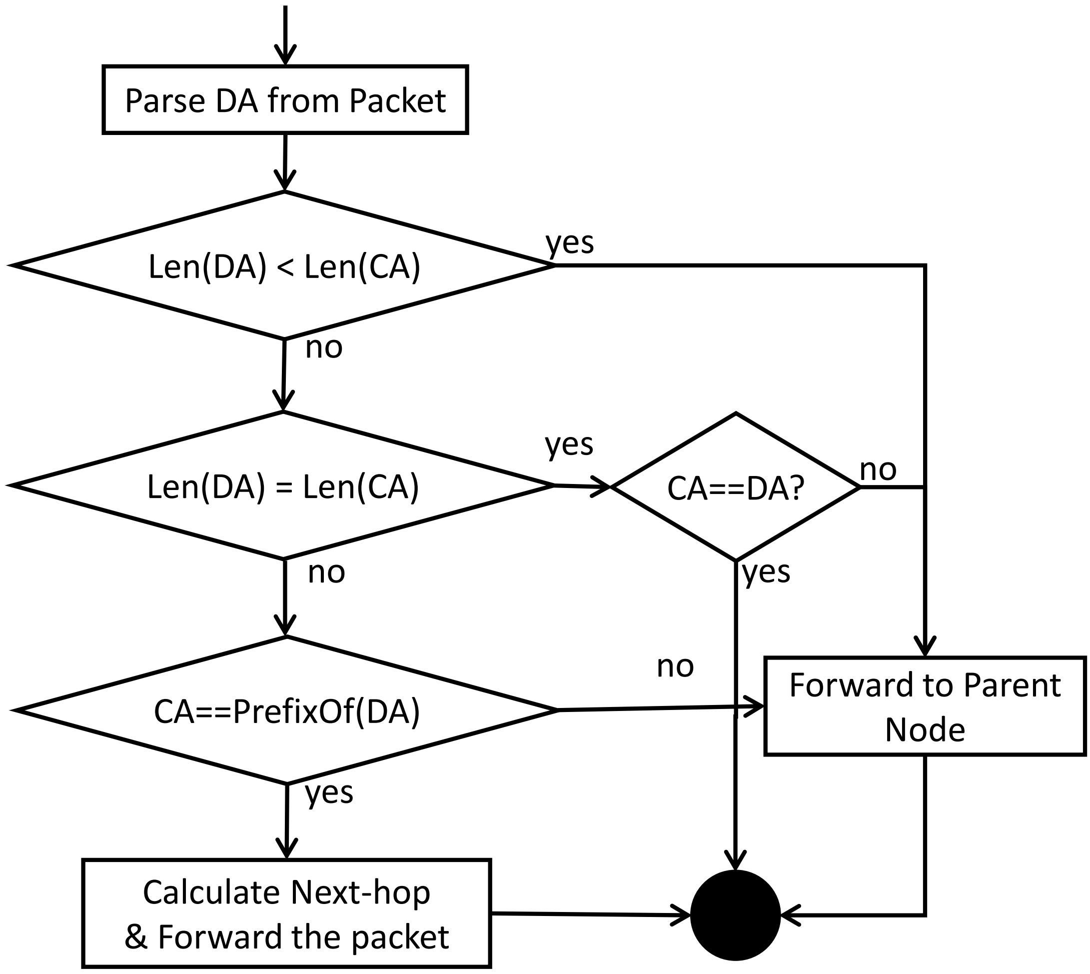
Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node's Address

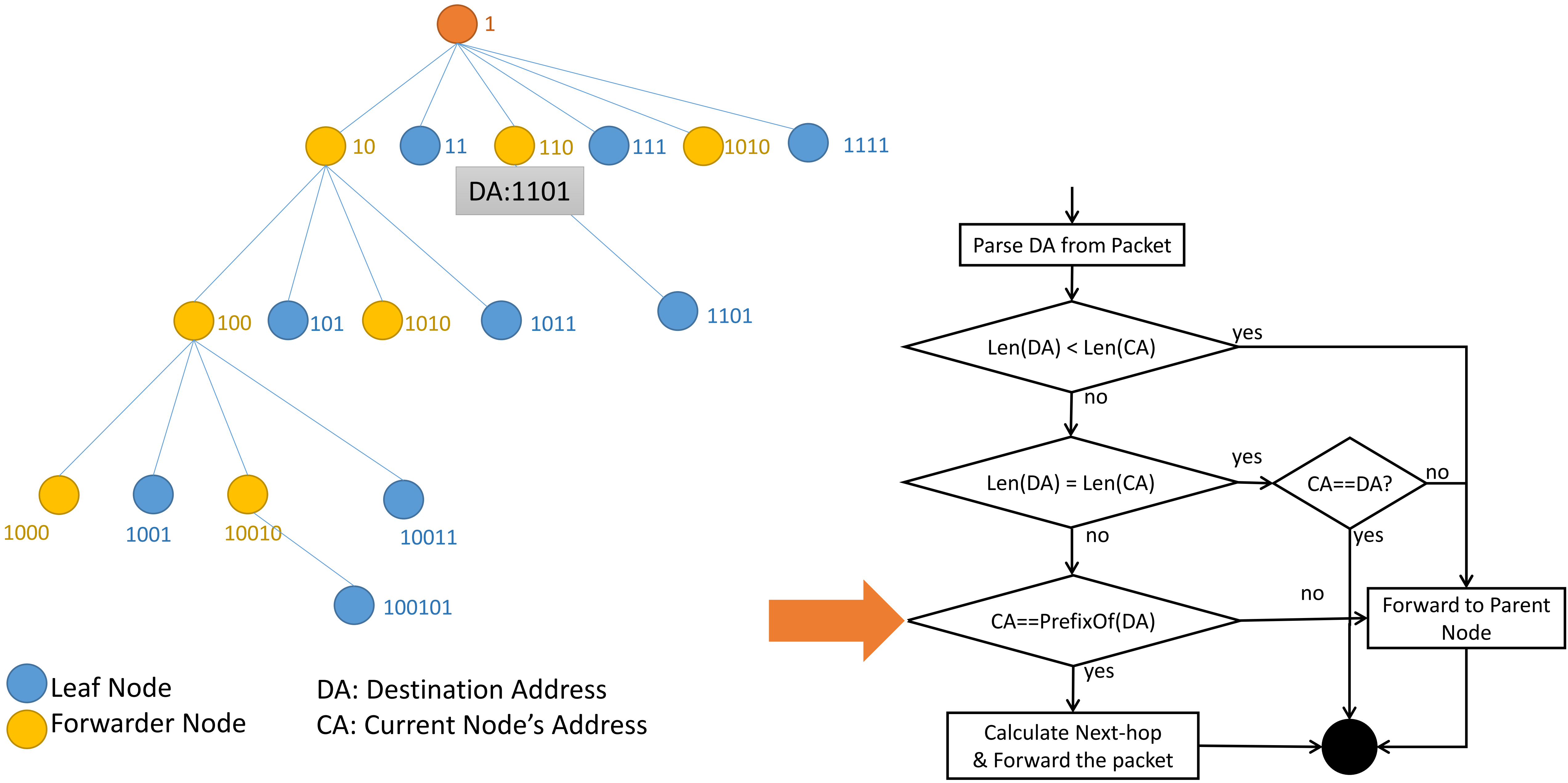
NSA Stateless Forwarding



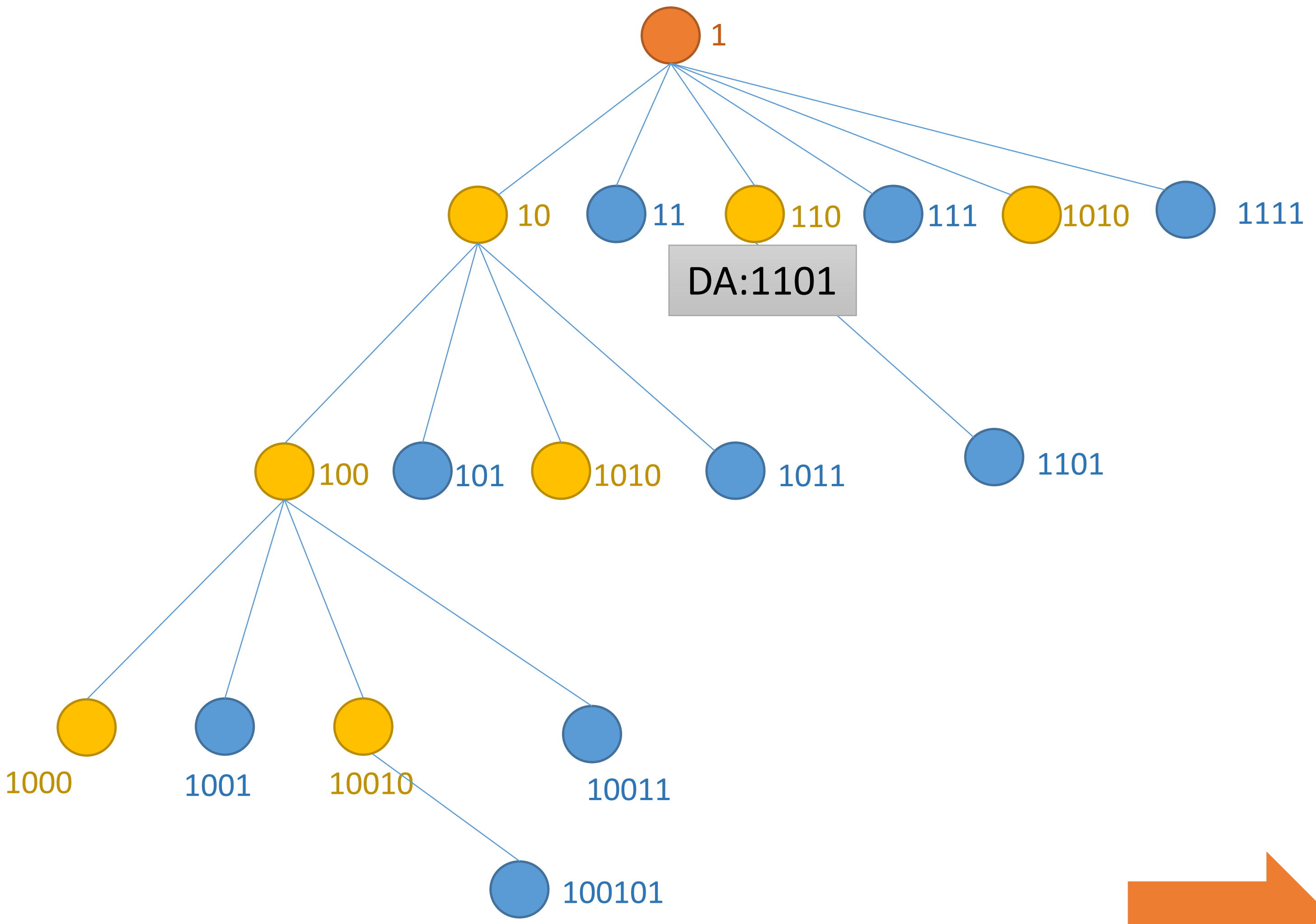
DA: Destination Address
CA: Current Node's Address

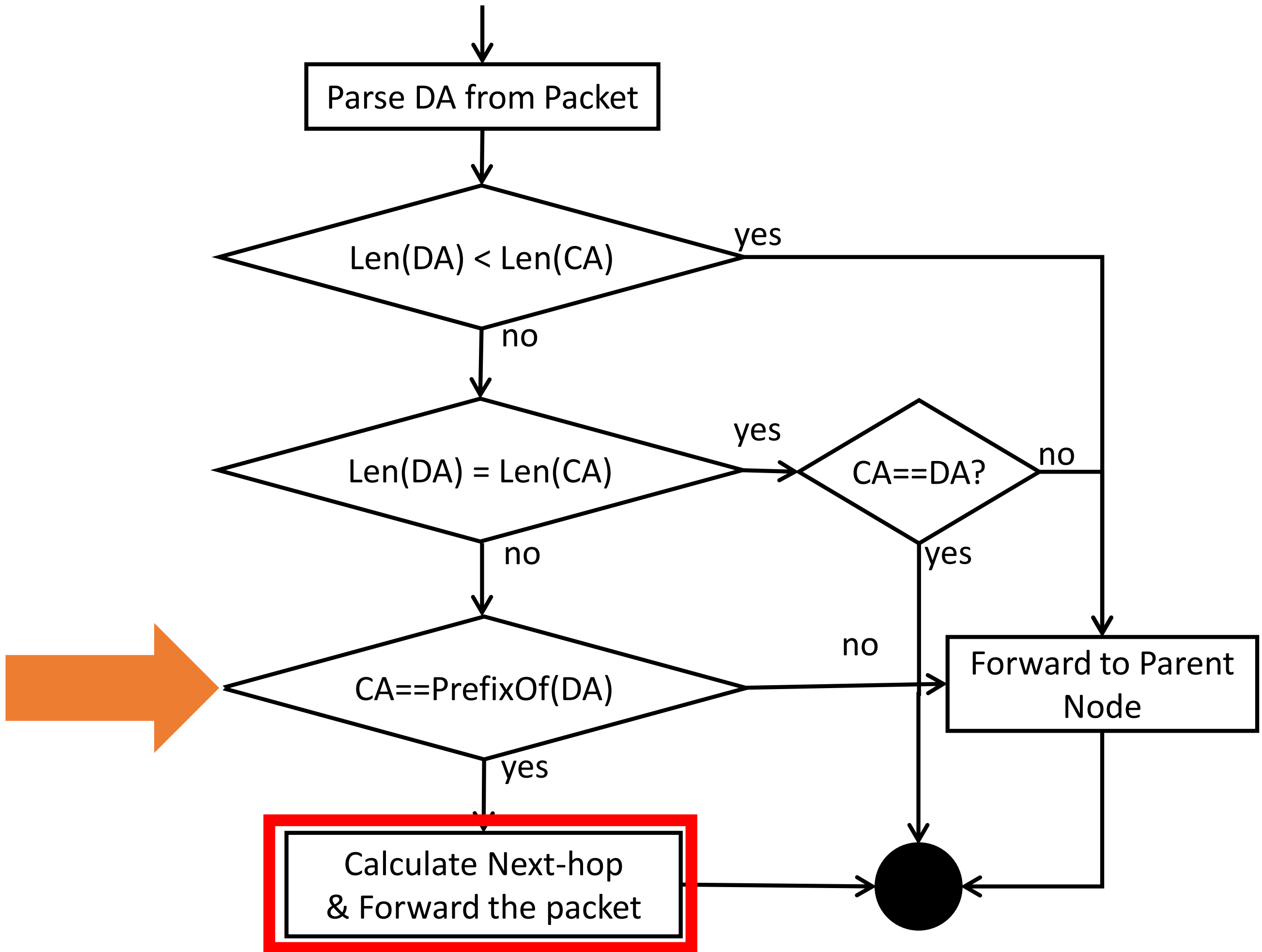


NSA Stateless Forwarding



NSA Stateless Forwarding

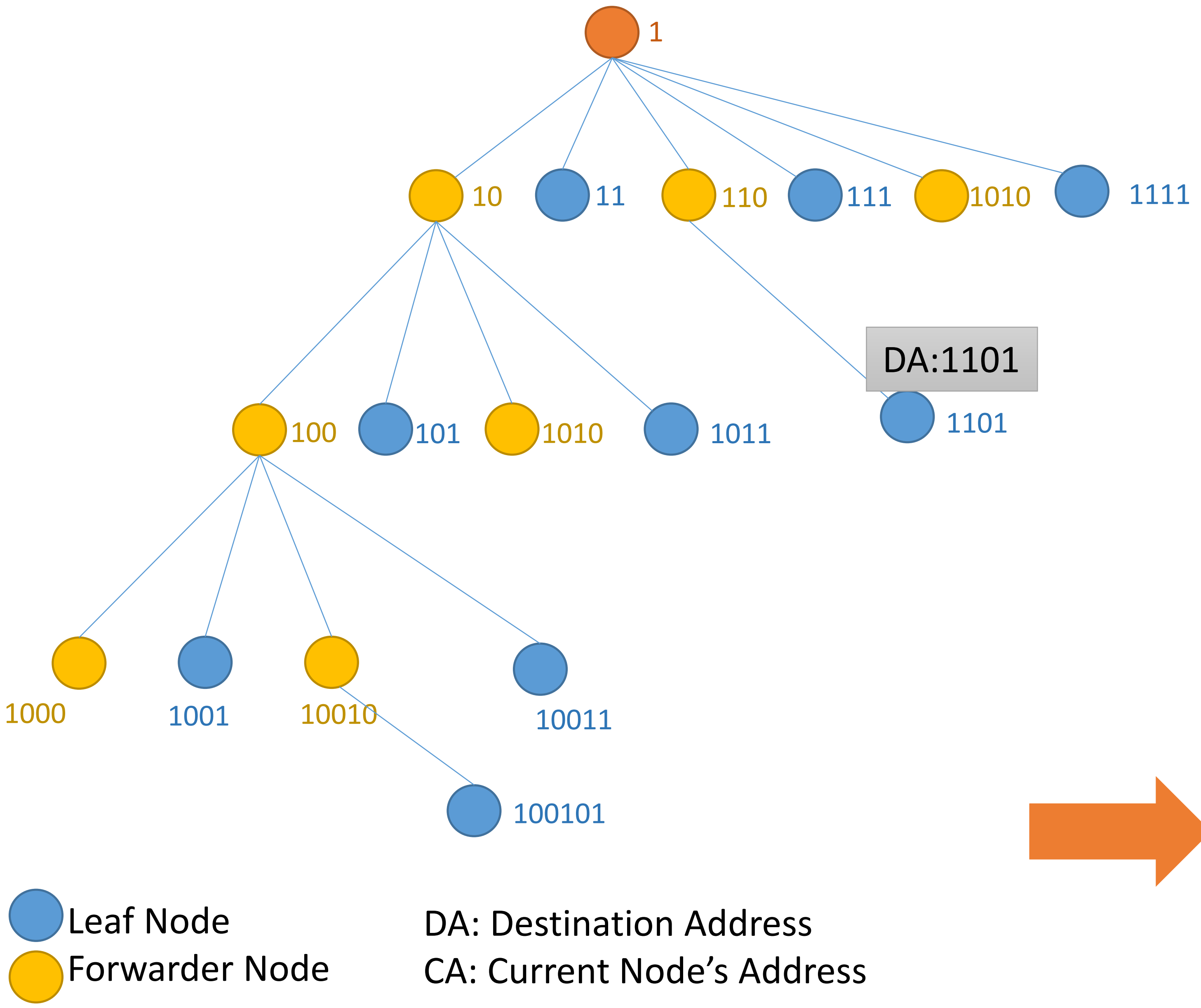




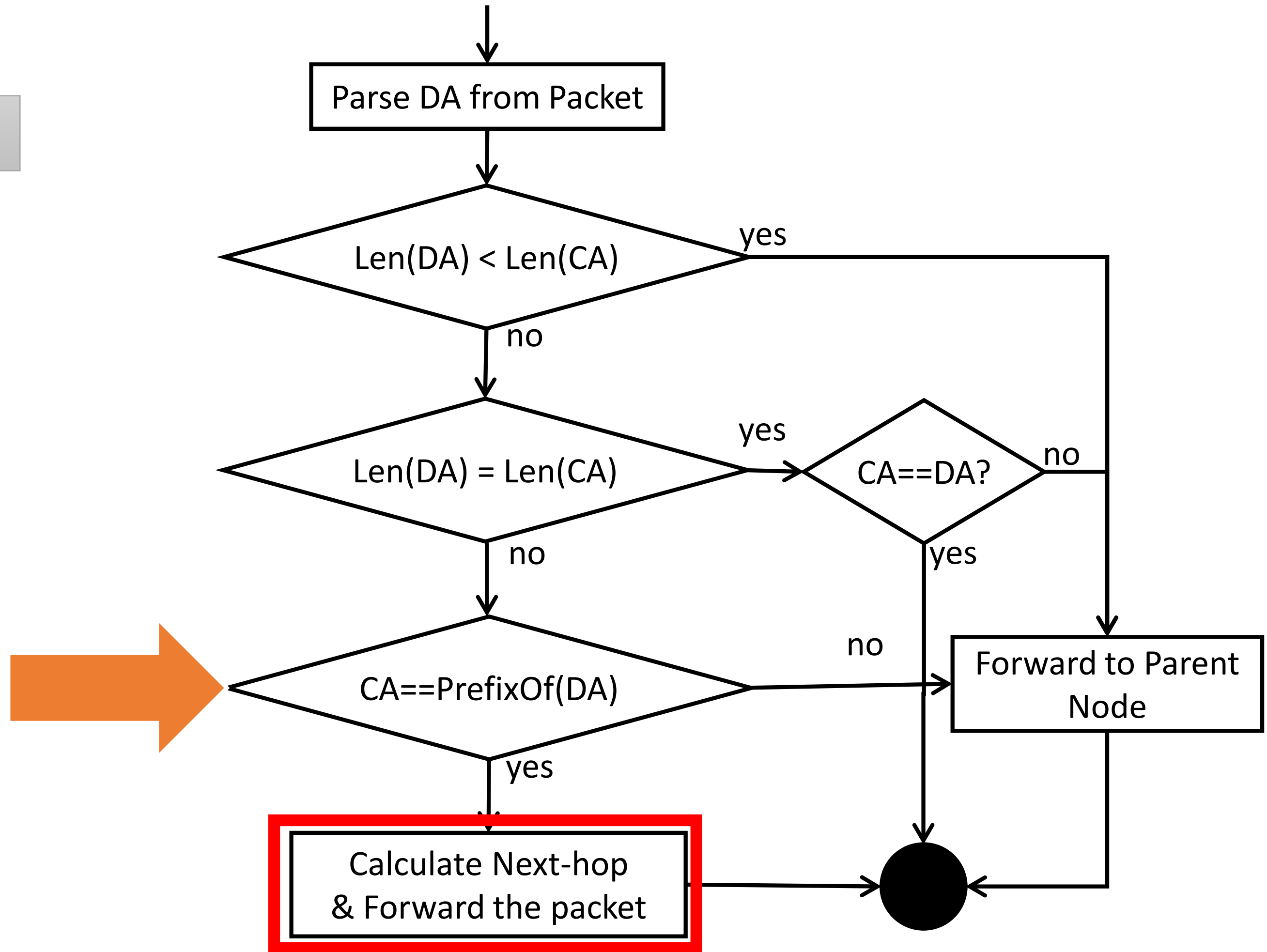
Leaf Node
Forwarder Node

DA: Destination Address
CA: Current Node’s Address

NSA Stateless Forwarding



DA: Destination Address
CA: Current Node's Address



Routing between NSA and IPv6 domains

- For NSA domain
  - The border router(root) can translate to IPv6 addresses
  - Concatenating IPv6 prefix and native short address (padding with 0s).
  - Trimming the IPv6 prefix and suppressing leading 0s in the suffix
- For external IPv6 addresses:
  - Border router maintain a table mapping external IPv6 destinations to a short address
  - Packet carries mapped address when transmitting in the domain
  - Border router will look up real IPv6 destination before sending the packets to IPv6 domain

6lo NSA Routing Header

- **RFC 8138:**
  - **IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header**
  - **Allows to define special type of header prepended to the 6lo-compressed IPv6 header**

- **6lo NSA Routing Header Format:**

```
 0                   1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-          ...          -+
|1|0|0|  size |0|     Type      | Address 1 to 16 bytes    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-          ...          -+
                                 <-- Length implied by Type/TSE -->
```

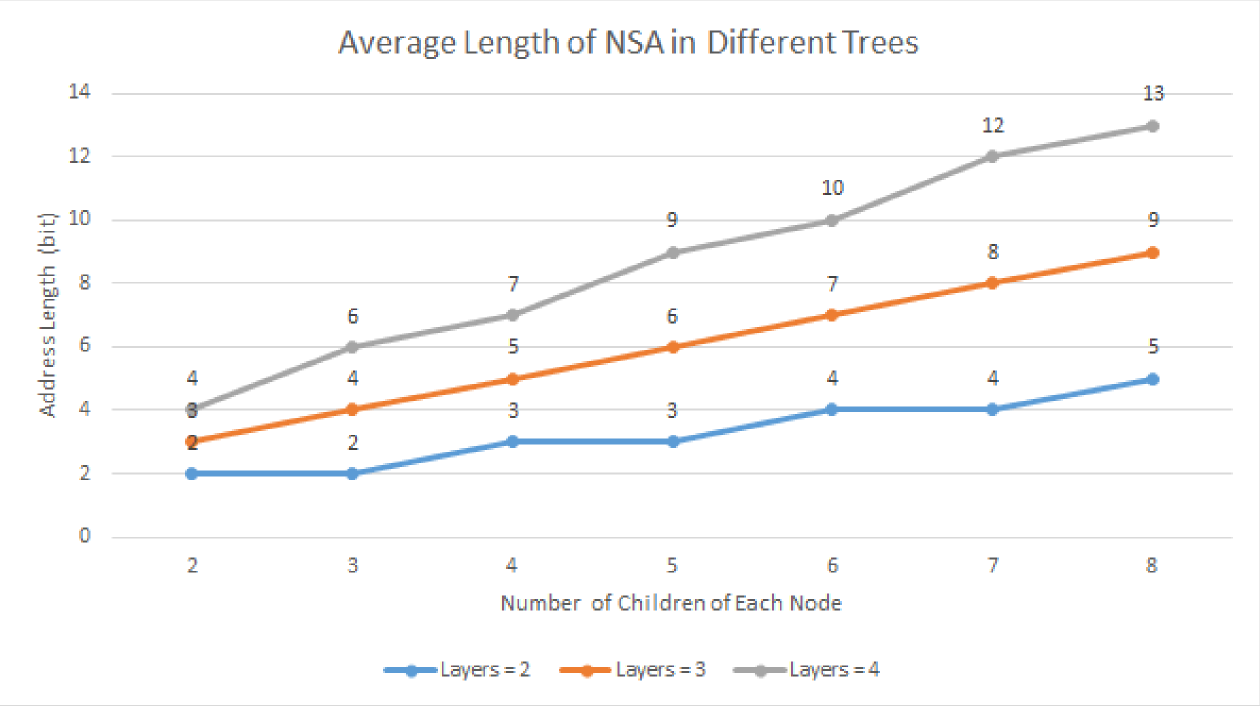
Size: 4 bits
0: unused
Type: <NSA_Type, I/O, MA>
I/O: Inward/Outward
MA: Mapped Address

I/O = 0, MA = 0: Destination address not NSA, neither a mapped Address (aka external full IPv6 address)
I/O = 0, MA = 1: Destination address is an external mapped address
I/O = 1, MA = 0: Destination and source addresses are NSA
I/O = 1, MA = 1: Destination address is NSA, source address is an external mapped address

Take away

Topological addressing scheme

- **Suitable for IoT networks with relative stable connections**
- **Can achieve smaller average address length (see figure)**
  - even if it is not the main target
- **Enables stateless forwarding**
  - **Enabling greener solution**
  - **Working on very constrained nodes**



Future Research direction

- **Reliability**
- **Quantification of gain w.r.t. routing**
  - **e.g. RPL**

Any question/comment welcome!

THANKS!