Ad-hoc On Demand Distance Vector (AODV)

Summary done by: M.S. Francisco Eduardo Balart Sánchez
Original Author: Pankaj

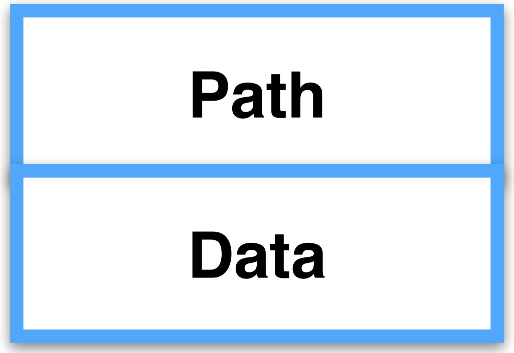
Ad-hoc On Demand Distance Vector (AODV)

- In this presentation, AODV will be explained.
- **AODV** stands for: **A**d-hoc **O**n **D**emand **D**istance Vector, routing protocol
- There are three kinds of routing protocols
    - Reactive
    - Proactive
    - Hybrid
- AODV falls under reactive routing protocol
- This means that whenever any node wants to send some data to another node, at that time, they will start searching for the path.
- And so if before if they don't want to send any data, they'll want to go for any route searching.
- So at the time of when it's needed, at that time, they'll start looking for, if there is any, ¿what is the physical path?

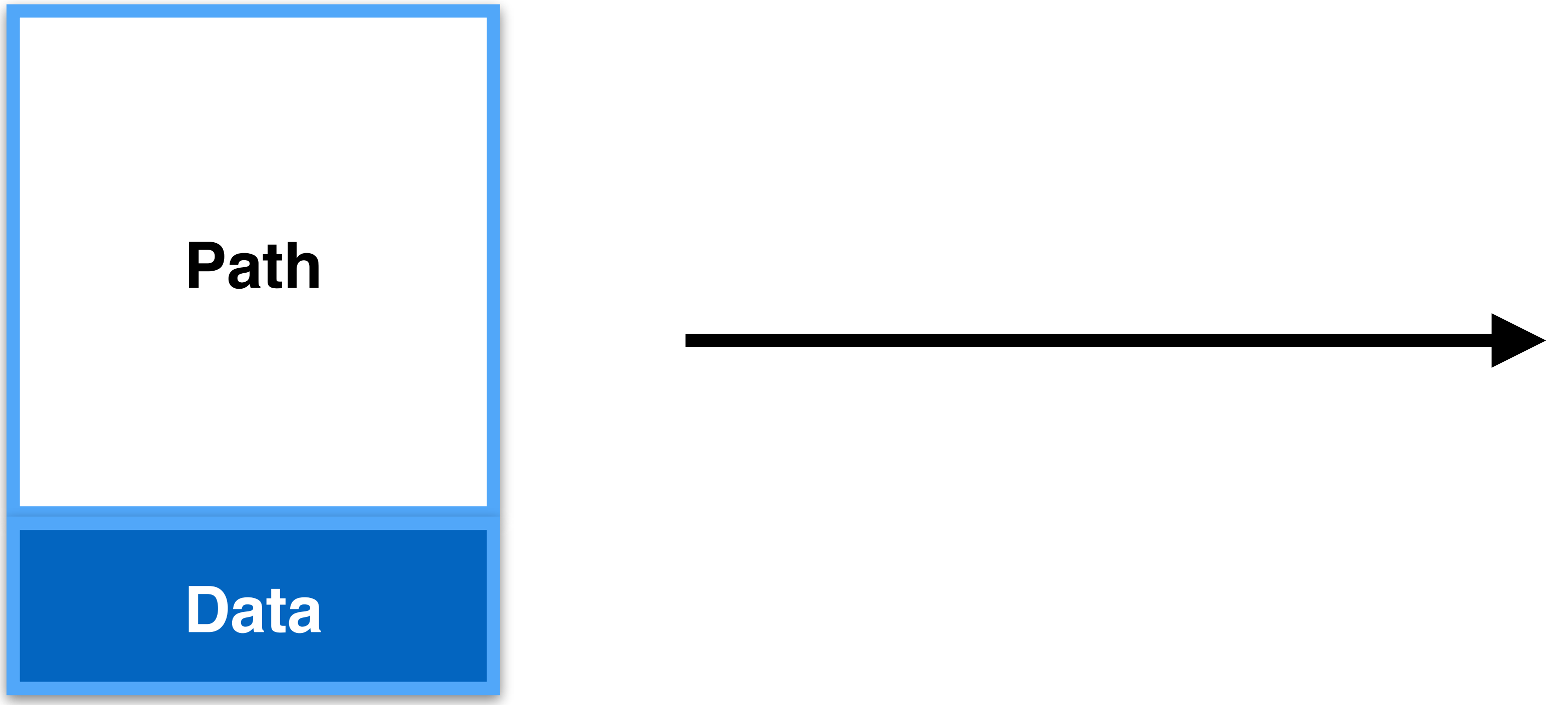
Ad-hoc On Demand Distance Vector (AODV)

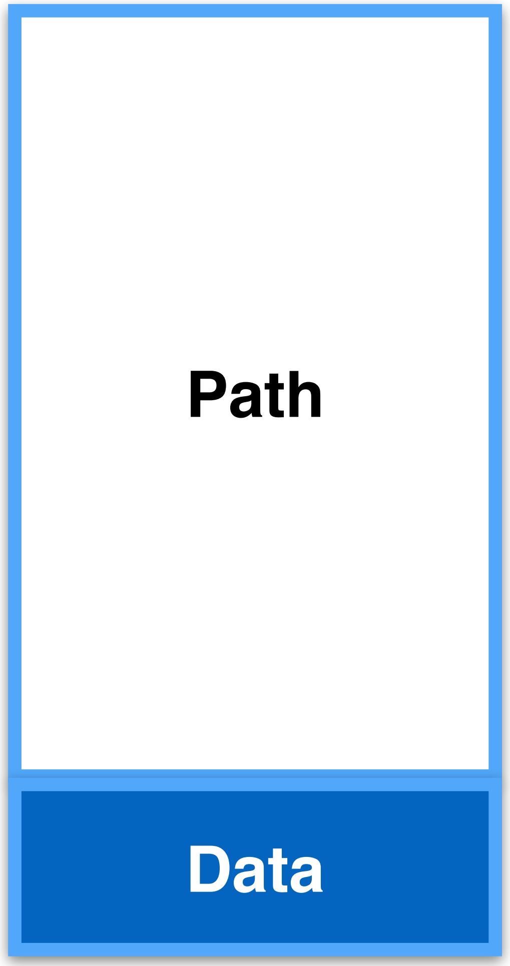
- AODV is an extension of **DSR**
- **DSR** stands for **D**ynamic **S**ource **R**outing
- DSR Disadvantages:
  - Whenever a node finds out a route to the destination, the path will be included in the data itself, the data packet carries the entire route, so for example:
    - If network is not large, the path will be small
    - If network size grow, the bytes required to store path in data packet will also increase.
    - Most of the time in this case network bandwidth will be not used or not utilized fully because most of the time bandwidth is used to send the path across the nodes.

Ad-hoc On Demand Distance Vector (AODV)
DSR-Disadvantage



As the network size grow length of path also increase





As the network size grows, route path also increased.

Data packet’s header also increase

Ad-hoc On Demand Distance Vector (AODV)

- DSR Disadvantage (cont.)
- For DSR, for one destination, there are multiple routes possible.
    - One node can cache multiple routes for the same destination.

AODV-Concept

Reactive routing

- Pure on-demand route acquisition system
- The routes are created when needed, so called “on-demand”

A broadcast route

- RREQ (**R**oute **Req**uest packet) broadcasting to find a route
- RREP (**R**oute **Rep**ly packet) is used to set up forward path

Dynamic establishment of route table entries

- Nodes lie on active paths only maintain routing information

Maintenance of timer-based states

- A routing table entry is expired if not used recently

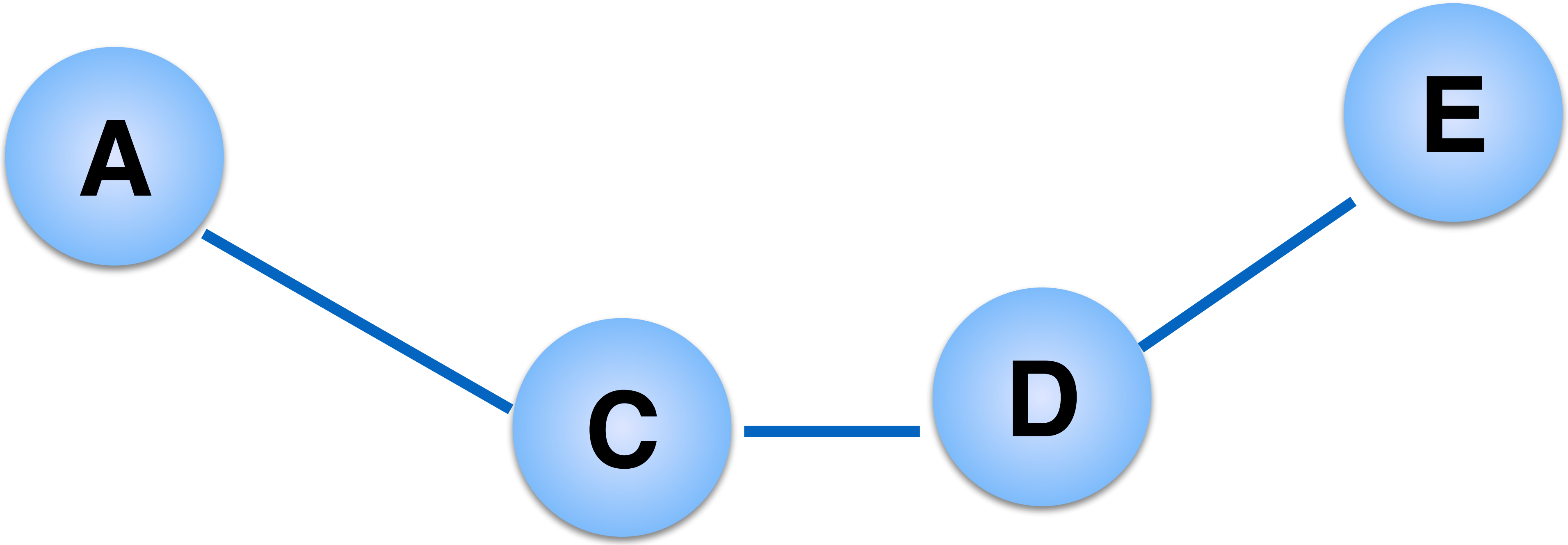
Destination sequence number

- Prevention of routing loops
- Avoidance of of and broken routes

AODV-Concept

- In AODV, the route discovery mechanism is similar as DSR.
- In order to discover a route, first, it will send a Route Request packet.
- When the destinations receive that packet, they will reply by sending a Route Reply back packet.
- AODV maintains all the routes in the form of a table, at the contrary of DSR which doesn’t use table.
- When any entry is made in the table, a time will be associated with that entry, that timer will specify at what time that entry should be removed from the table.
- The destination sequence number can be considered as a timestamp. In this case with the destination sequence number we can know if the coming information is recent (fresh) or not.
- Also, the inclusion of this sequence number also avoids routing loops

Route Discovery



- **Objective**: A wants to send data to E.
- Initially node A doesn’t have any information about where is node D, and if it exist in the network.
- Node A will start a route discovery mechanism

Route Discovery

- Node A will broadcast a Route Request Packet (RREQ)
  - The sequence numbers will work as a time stamp
  - Unique Broadcast ID, if the node send out a new Route Request Packet it will increase the Broadcast ID, so in this way this new RREQ will have another unique ID.
- Format of the RREQ of AODV (numbered by entry)
  1. Source address
  2. Source Sequence Number
  3. Broadcast ID
  4. Destination Address
  5. Destination sequence number (empty at the beginning)
  6. Hop count (hops required to send data from Tx to Rx)

Route Discovery

1. Every Node mantain two counters

Sequence Number

Broadcast_id: increments whenever the source issues a new RREQ

2. Source broadcast RREQ data packet for searching route

<source_addr, source_sequence#,broadcast_id, dest_addr, dest_sequence#, hop_cnt>

It is a time stamp. When destination send RREP it increase number when it send RREQ

Route Discovery

- When destination receive a Route Request, it will check the destination address, and will find out who’s this route discovery process is for.
- Then destination will reply by using doubt reply packets
- Format of the Route Reply request packet
  1. Source Address
  2. Destination Address
  3. Destination Sequence Number (time stamp)
  4. Hop Count
  5. Lifetime
     - Specify for how long this entry, this path, can be used

Route Discovery

- Intermediate Node
- Whenever they hear a duplicate packet they will discard it
- If sender is requesting for its rout they will be broadcasting the Route Request Packet
- In case the Intermediate node has any updated information, instead of broadcasting a RREQ they will send a reply to the sender

Route Discovery

3. Destination replies using RREP (Route Reply) unicasting

<source_addr, dest_addr, dest_sequence#, hop_cnt, lifetime>

RREP contains the current sequence number, hop count=0, full lifetime

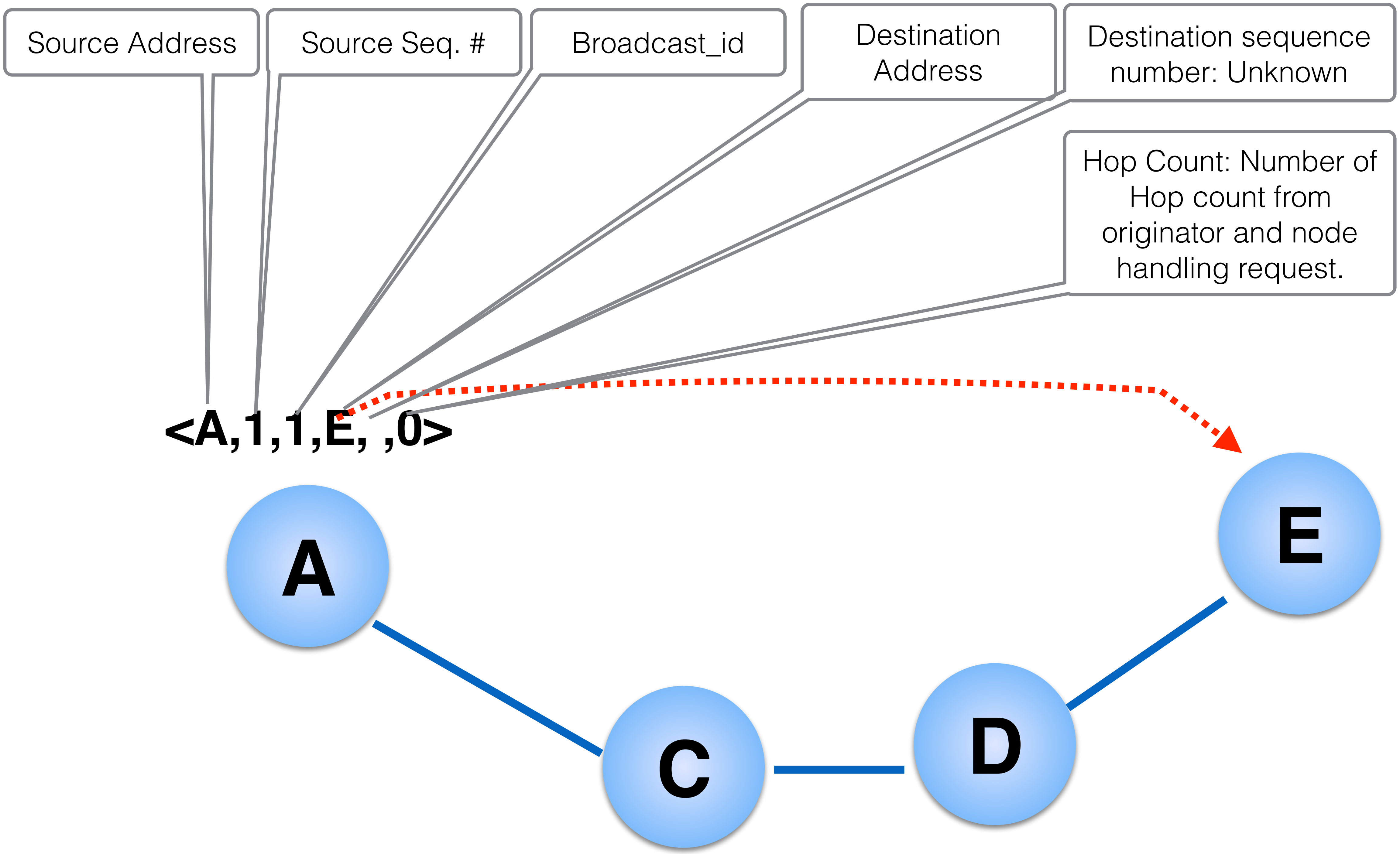
4. Intermediate Node

Discard duplicate packet

Send RREP if it has active route with higher sequence number

Otherwise broadcast packet

AODV-Route Discovery [RREQ]



Source Address

Source Seq. #

Broadcast_id

Destination Address

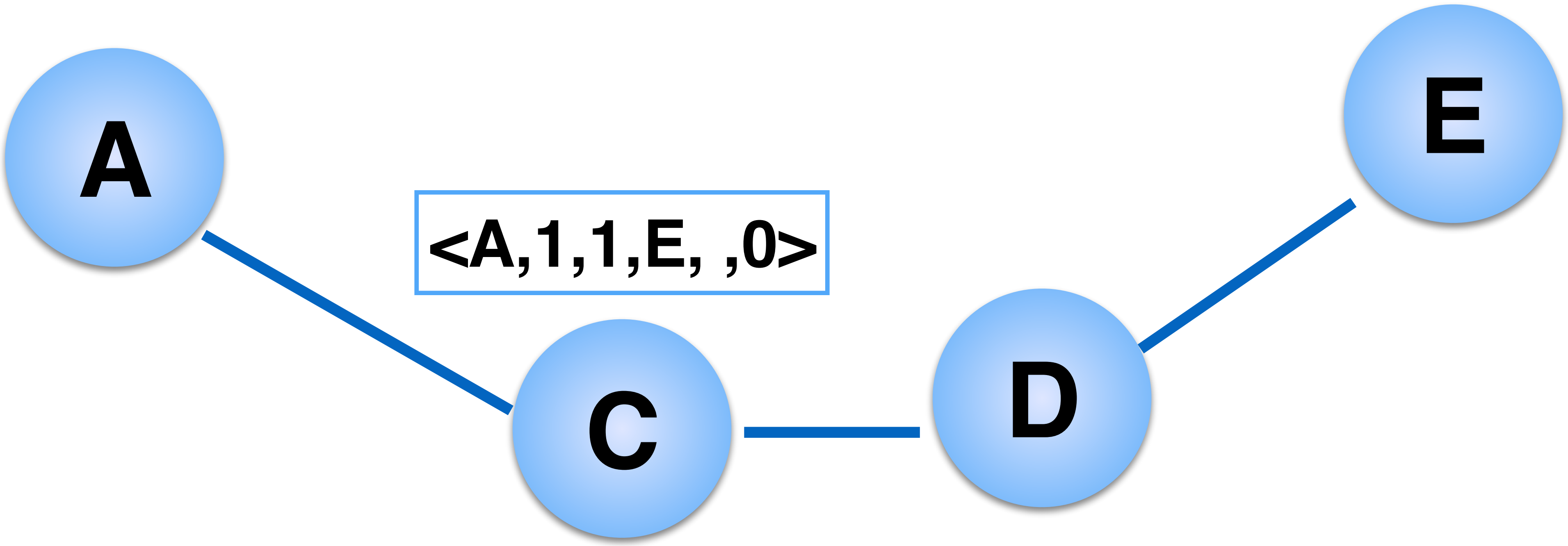
Destination sequence number: Unknown

Hop Count: Number of Hop count from originator and node handling request.

<A,1,1,E, ,0>

- A will send RREQ packet to all it’s neighbors

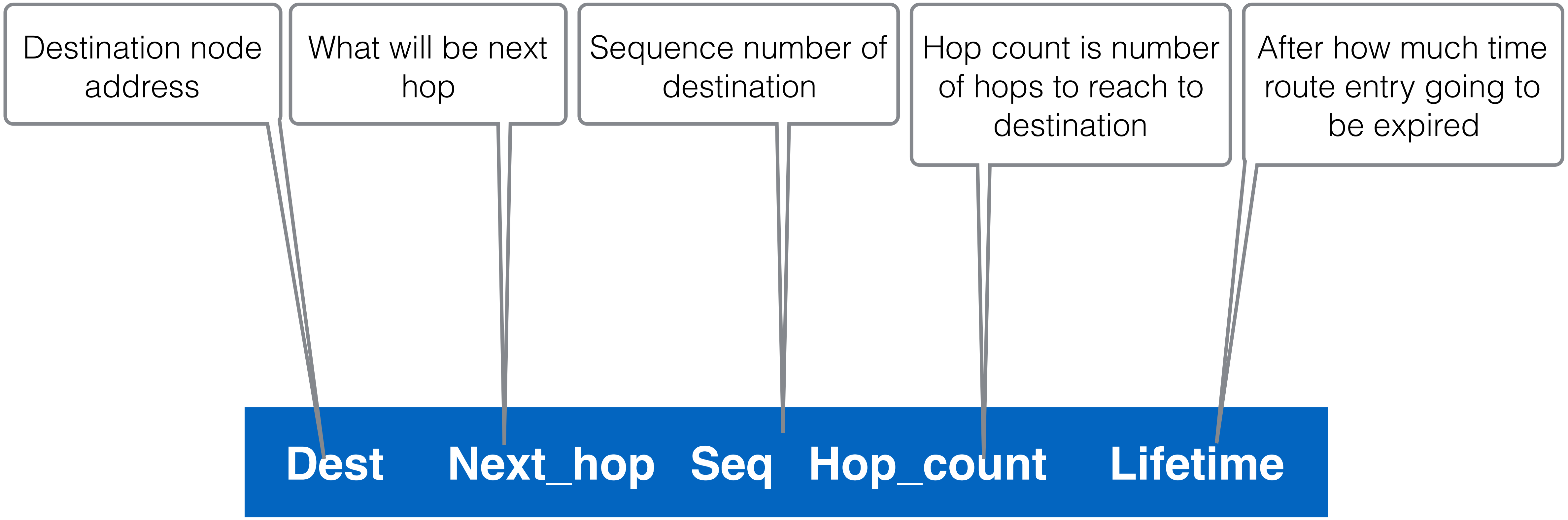
AODV-Route Discovery [RREQ]



C will set a reverse path to node from which it is receiving RREQ.
C will store information in table

- **A** having only 1 neighbor will send the packet to C
- C does not have any information about node E, so C will broadcast this packet again
- Before broadcasting this packet, C will update this hop count

AODV-Routing Table Entry



Destination node address

What will be next hop

Sequence number of destination

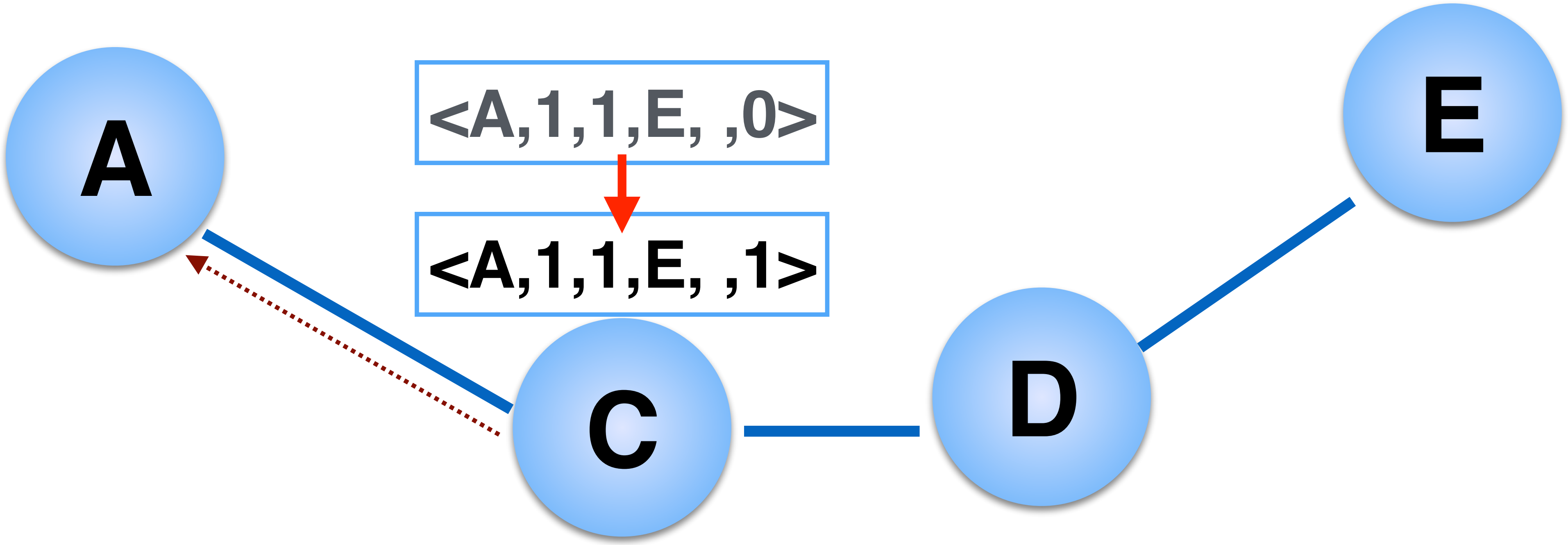
Hop count is number of hops to reach to destination

After how much time route entry going to be expired

Dest Next_hop Seq Hop_count Lifetime

AODV-Route Discovery [RREQ]

C increase hop count in RREQ
C check it’s routing table if it has a valid route to E it reply.

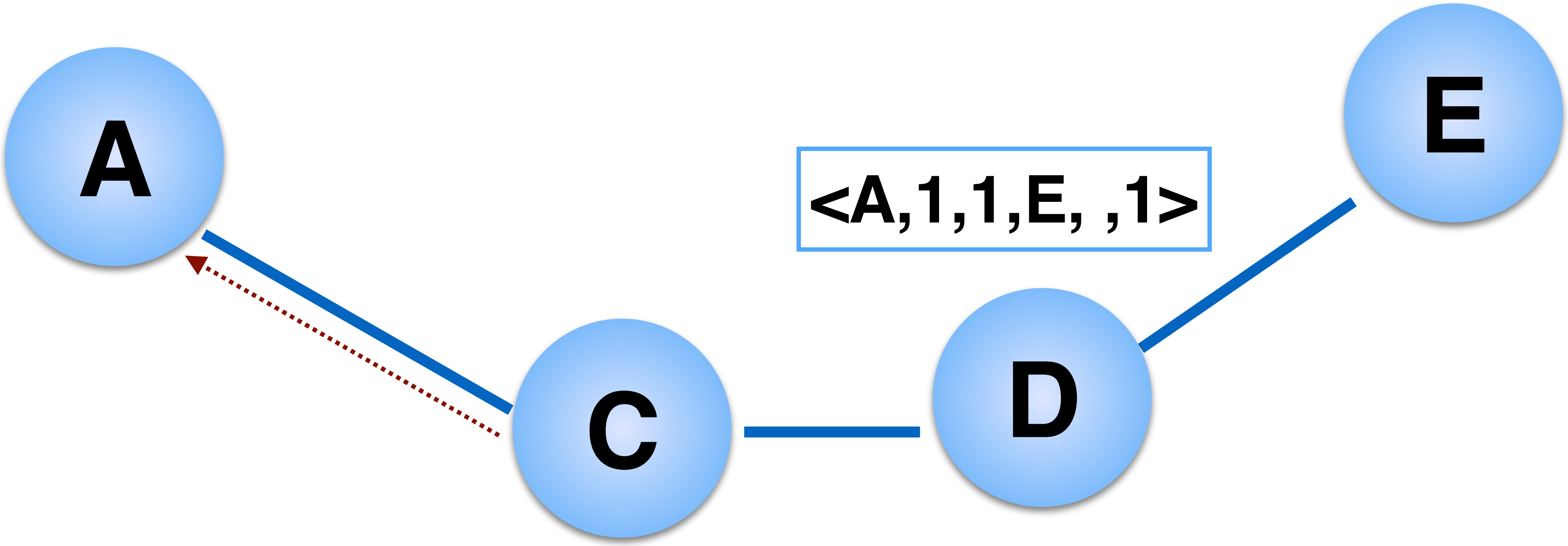


Dest	Next	Hop	Seq
A	A	1	1

C check it’s routing table and find nothing.
C add a entry for reverse path to source

- C will know this packet is actually coming from A

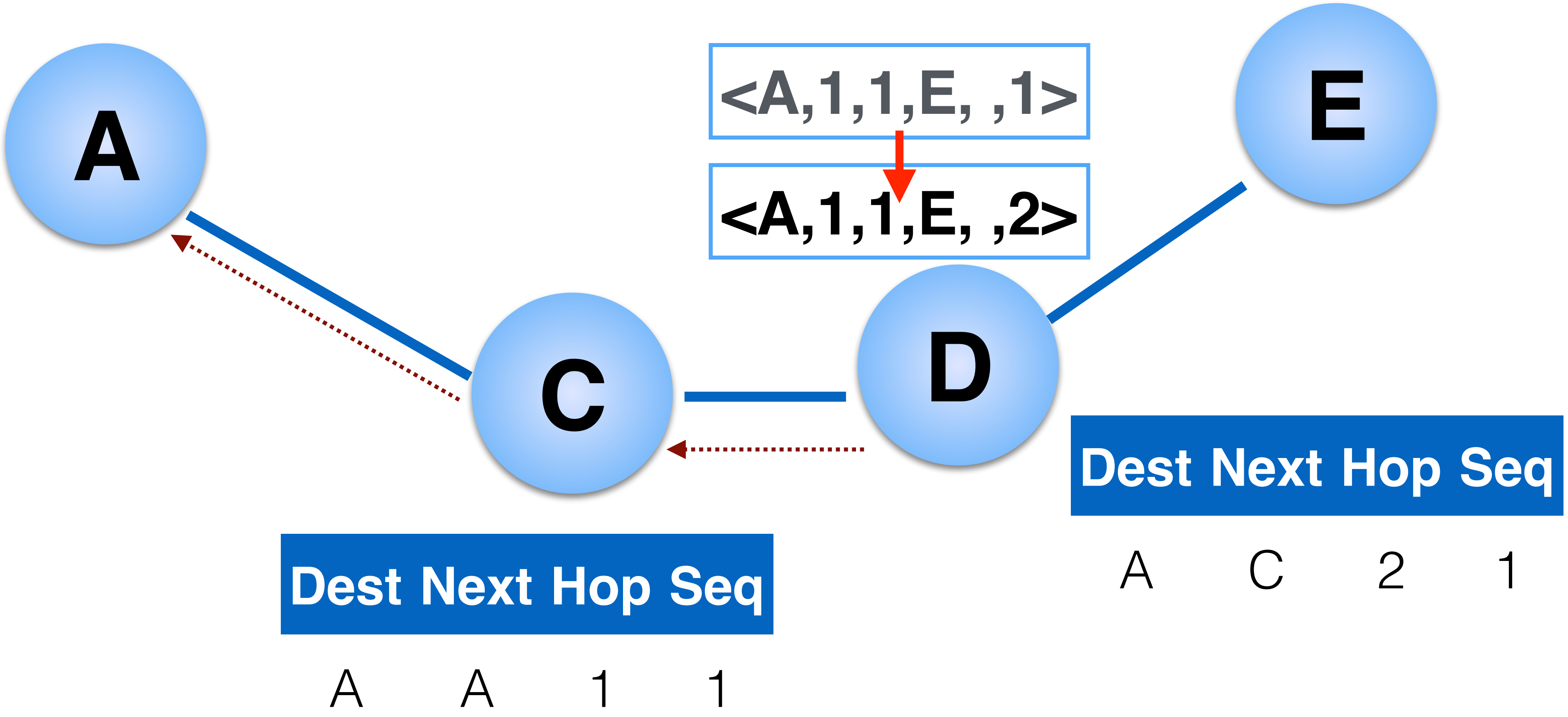
AODV-Route Discovery [RREQ]



Dest	Next	Hop	Seq
A	A	1	1

- C increase the hope count and broadcast the packet again

AODV-Route Discovery [RREQ]



<A,1,1,E, ,1>

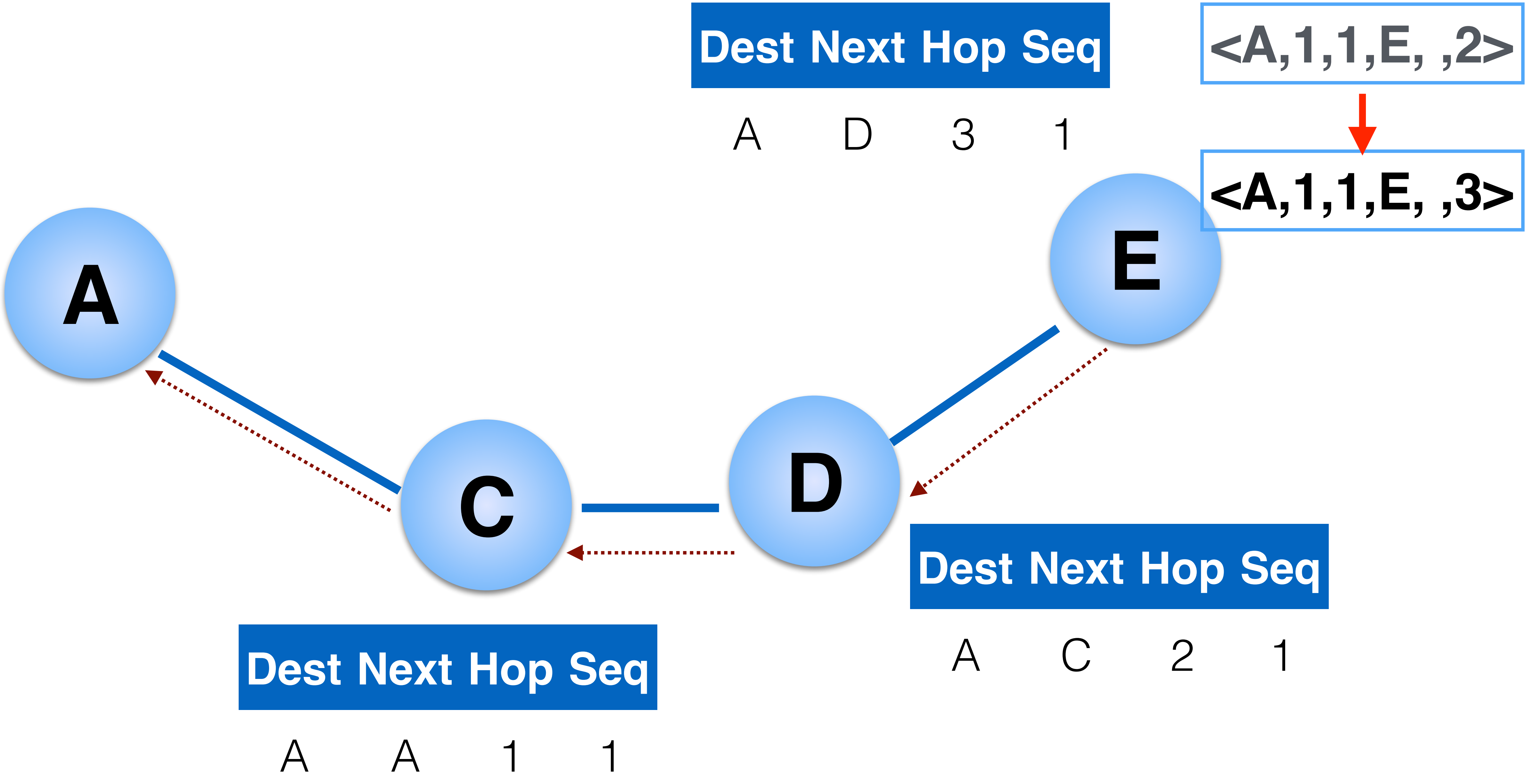
<A,1,1,E, ,2>

Dest	Next	Hop	Seq
A	A	1	1

Dest	Next	Hop	Seq
A	C	2	1

- D will check that the destination is E, so D will also broadcast this packet.
- But before broadcast the packet, D will make an entry in its table
- it will make the entry if this packet is coming from A
- Node D will know that in order to reach destination A it will have to go to C
- It will also increase the Hop Count

AODV-Route Discovery [RREQ]



<A,1,1,E, ,2>

<A,1,1,E, ,3>

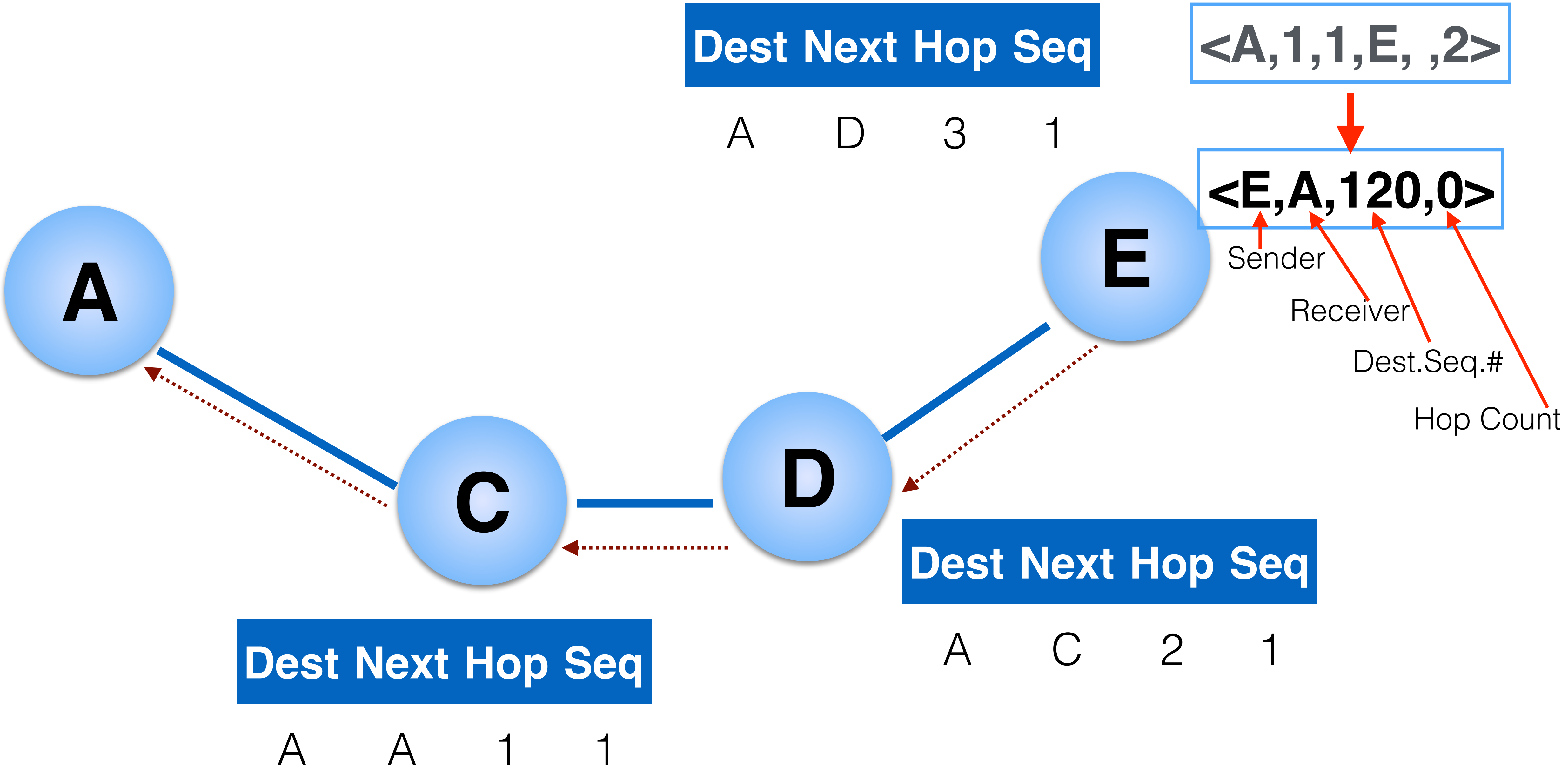
Dest	Next Hop	Hop	Seq
A	D	3	1

Dest	Next Hop	Hop	Seq
A	C	2	1

Dest	Next Hop	Hop	Seq
A	A	1	1

- E will check that the RREQ is exactly for itself
- Therefore it will prepare a Route Reply packet
- And before that will also create an entry in the routing table

AODV-Route Reply [RREP]



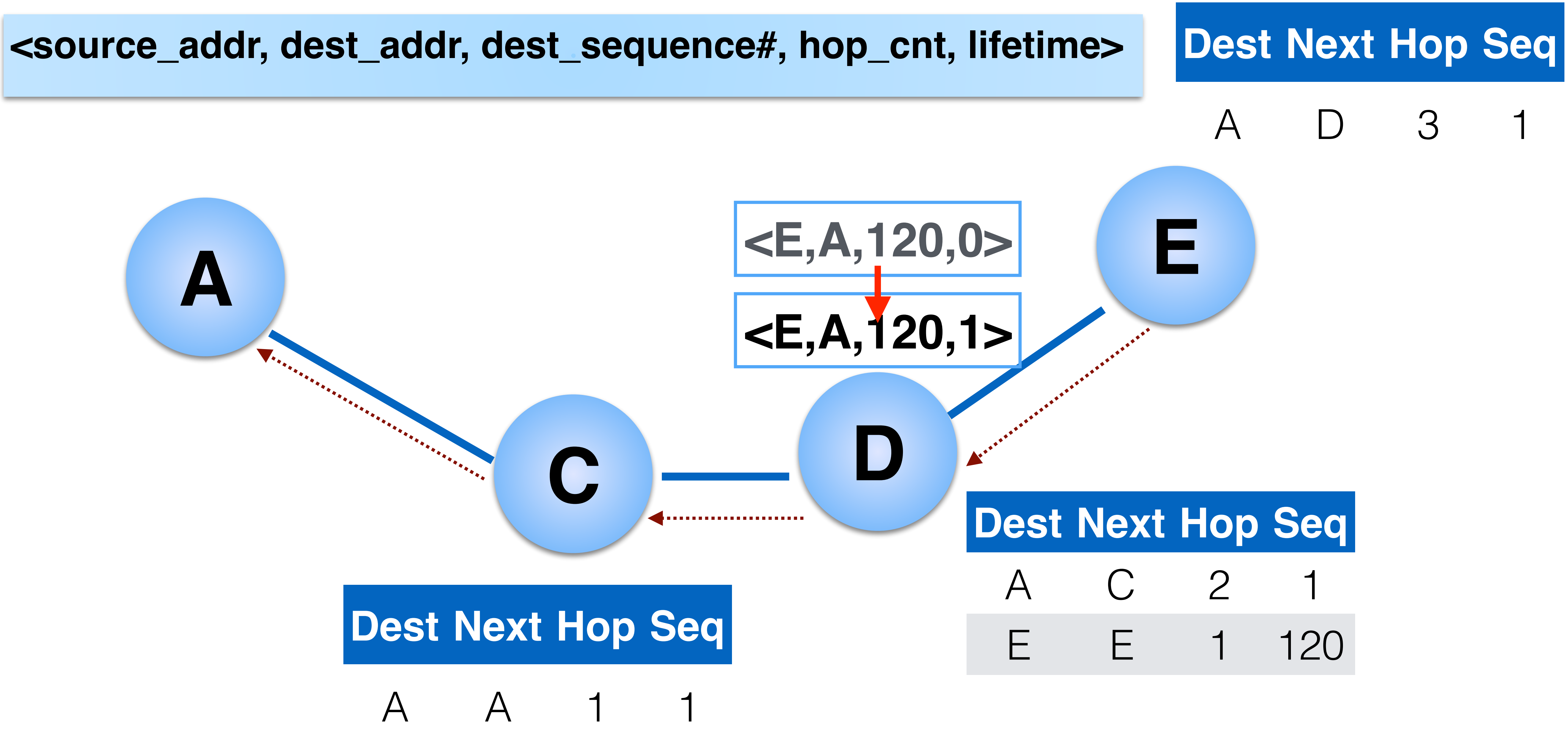
Dest	Next	Hop	Seq
A	D	3	1

Dest	Next	Hop	Seq
A	C	2	1

Dest	Next	Hop	Seq
A	A	1	1

- E will check that the RREQ is exactly for itself
- Therefore it will prepare a Route Reply packet
- And before that will also create an entry in the routing table

AODV-Route Reply [RREP]



<source_addr, dest_addr, dest_sequence#, hop_cnt, lifetime>

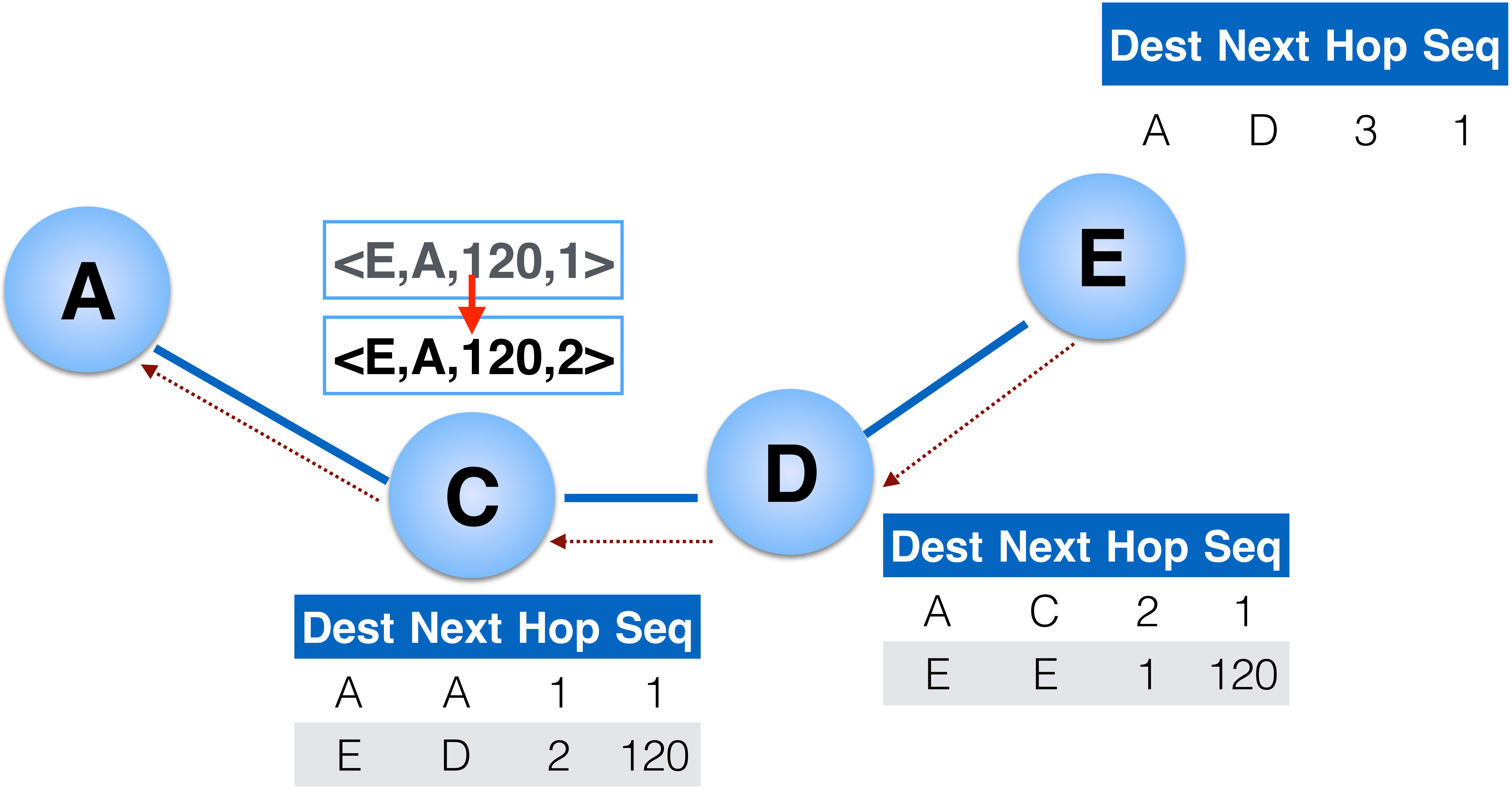
Dest	Next	Hop	Seq
A	D	3	1

Dest	Next	Hop	Seq
A	C	2	1
E	E	1	120

Dest	Next	Hop	Seq
A	A	1	1

- E will send this value to D
- E make an entry in its table in order to know how to reach E
- It will also increase the hop count and send the packet to C

AODV-Route Reply [RREP]



Dest	Next	Hop	Seq
A	D	3	1

<E,A,120,1>

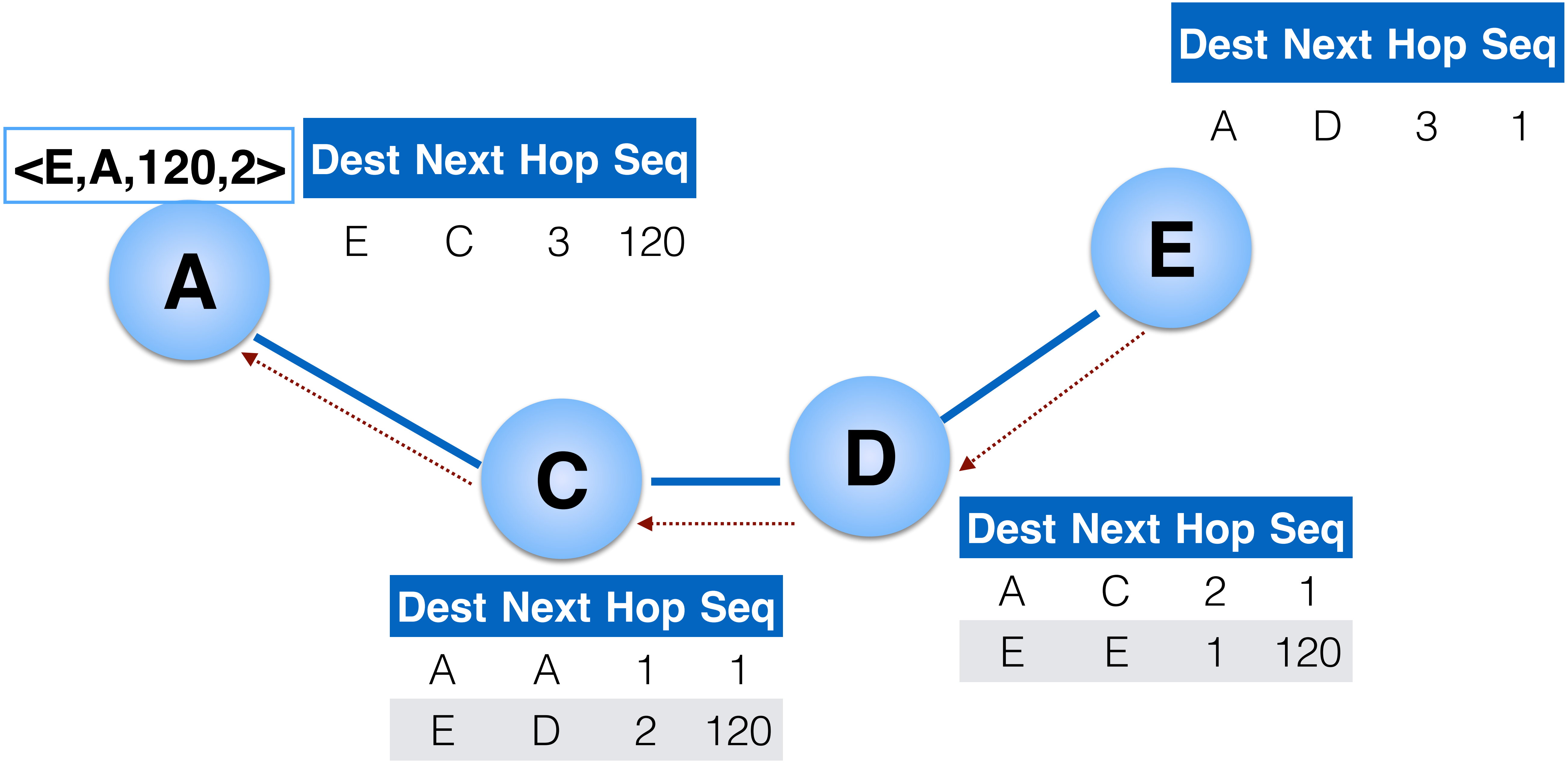
<E,A,120,2>

Dest	Next	Hop	Seq
A	A	1	1
E	D	2	120

Dest	Next	Hop	Seq
A	C	2	1
E	E	1	120

- C will make an entry in order to know how to reach E
- It will also increase the hop count and send the packet to A

AODV-Route Reply [RREP]



Dest	Next Hop	Seq	
E	C	3	120

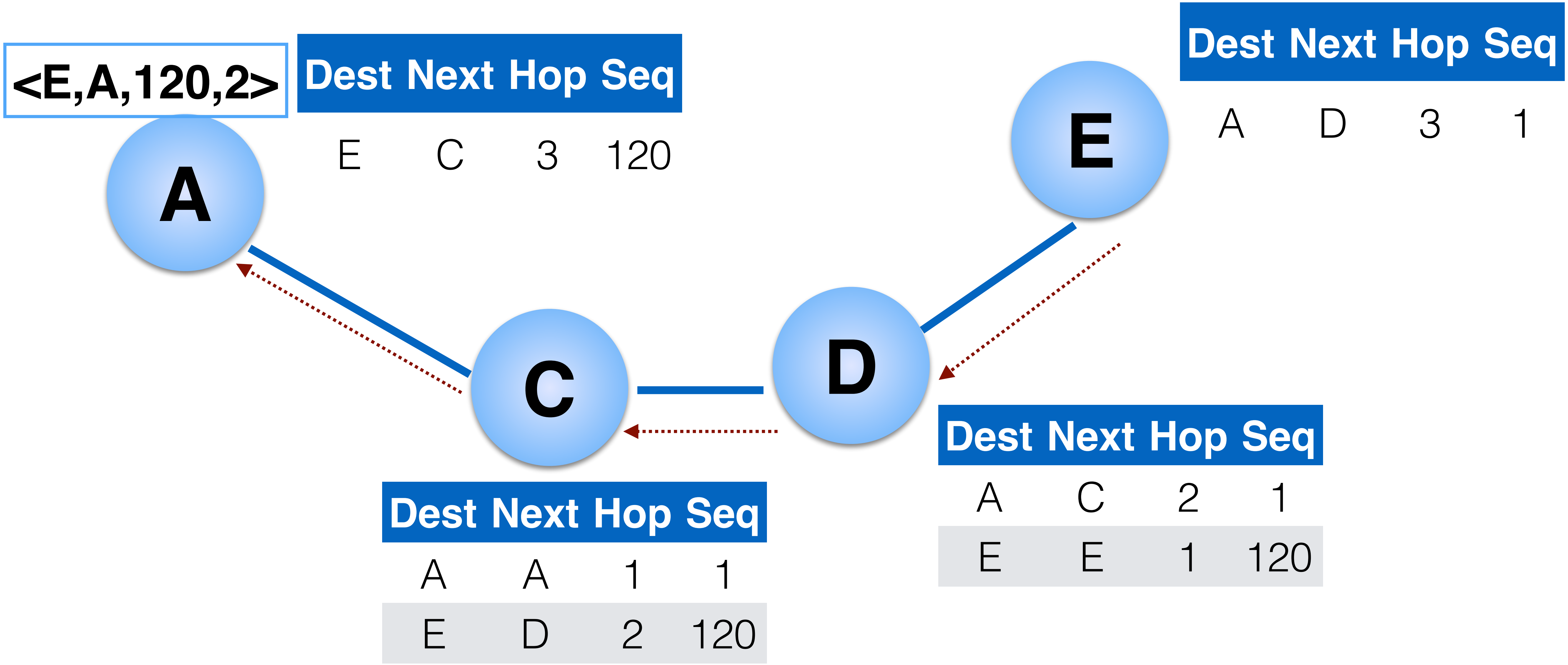
Dest	Next Hop	Seq	
A	D	3	1

Dest	Next Hop	Seq	
A	A	1	1
E	D	2	120

Dest	Next Hop	Seq	
A	C	2	1
E	E	1	120

- A will create an entry in order to know how to reach E
- Now all nodes have updated their routing table
- All nodes have the information to reach node E

AODV-Route Reply [RREP]



Dest	Next	Hop	Seq
E	C	3	120

Dest	Next	Hop	Seq
A	D	3	1

Dest	Next	Hop	Seq
A	C	2	1
E	E	1	120

Dest	Next	Hop	Seq
A	A	1	1
E	D	2	120

- Now the data will not carry the path information nor route information, cause every node has their routing table.
- Whenever they receive any data packet they’ll consult their routing table, on the basis of that routing table they’ll forward that received data packet to other nodes.