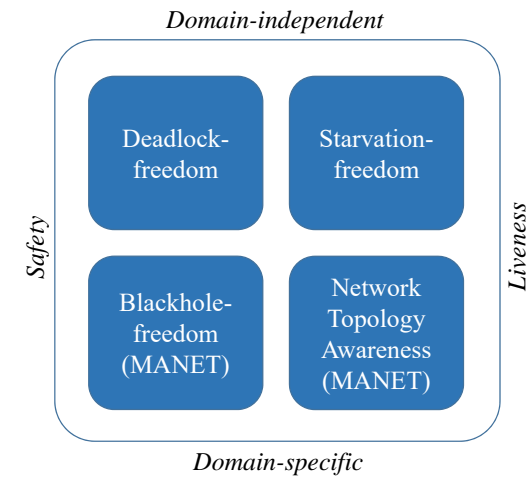


# I protocolli N-AODV e B-AODV per MANET

Gennaro Vessio  
gennaro.vessio@uniba.it

## Classi di proprietà



2

## Le MANET

- Una rete mobile ad-hoc (**MANET**) è una rete wireless caratterizzata da:
  - **Assenza** di infrastruttura fisica fissa
  - Topologia **dinamica**
- Specifici protocolli di routing si basano sulla **cooperazione** fra più host per stabilire route fra coppie di end-point:
  - Proattivi
  - Reattivi
  - Ibridi

3

## Network Topology Awareness

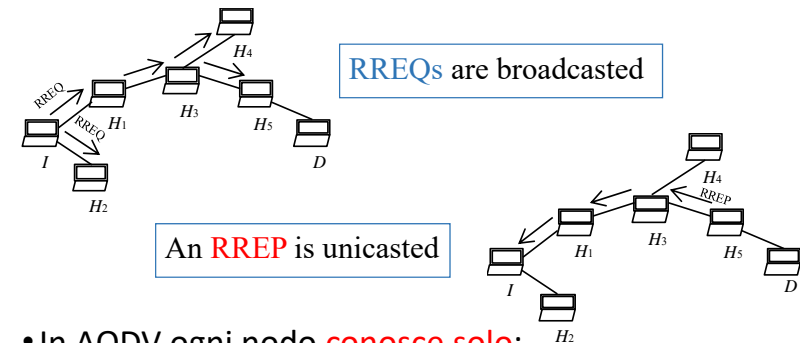
4

## Concetto generale

- La network topology awareness (**NTA**) si riferisce alla **conoscenza** che ciascun host ha:
  - Degli altri host
  - Della loro raggiungibilità attraverso una route
- Tale aspetto è importante in svariate applicazioni:
  - Algoritmi di elezione del leader
  - Sicurezza
  - VANET
  - ...
- *Come varia la NTA in funzione del protocollo di routing adottato?*

5

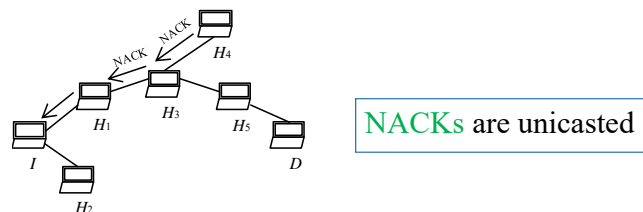
## Limiti di AODV



- In AODV ogni nodo **conosce solo**:
  - I suoi vicini
  - Il prossimo nodo nel percorso verso nodi non vicini
- *Come migliorare tale aspetto?*

6

## NACK-based AODV (N-AODV)



- Ogni nodo intermedio che **ignora** come raggiungere la destinazione comunica all'iniziatore del route discovery che «non sa nulla!»
- *Cos'è cambiato?*

7

## Modellazione in ASM

- Una MANET che adotta N-AODV può essere modellata da una **DASM** composta di  $Hosts = \{h_1, \dots, h_n\}$
- Ogni ASM  $h_i$  è caratterizzata da:
  - *neighb*:  $Hosts \rightarrow PowerSet(Hosts)$
  - *wishToInitiate*:  $Hosts \times Hosts \rightarrow boolean$
  - Tre code di pacchetti
  - Una tabella di routing

8

## Host program

```
HostProgram(hi) =
if ¬isEmpty(requests(self)) then {
  RREQ := top(requests(self))
  previousHop := sender of RREQ
  UpdateRoutingTable(self, RREQ)
  Router(RREQ, previousHop)
  dequeue RREQ from requests(self)
}
if wishToInitiate(self, dest) = true then
  Initiator(dest)

if ¬isEmpty(nacks(self)) then {
  NACK := top(nacks(self))
  if NACK.dest ≠ self then {
    previousHop := select r.nextHop ∈
      routingTable(self) with
        r.dest = NACK.dest
    UpdateRoutingTable(self, NACK)
    enqueue NACK into nacks(previousHop)
    dequeue NACK from nacks(self)
  }
}
```

9

## Predicati sugli stati

- Ciascuna ASM può trovarsi in uno fra più stati computazionali ciascuno caratterizzato da uno o più **predicati sugli stati**:
  - idle
  - routing
  - initiating
  - forwarding
- Per esempio, forwarding =  $isEmpty(replies(self)) = false \vee isEmpty(nacks(self)) = false$

10

## Correttezza

**Theorem.** *The route discovery process always ends*

*Sketch of proof.* Each ASM can enter four different computational branches depending on which rule is executed. Every time a computational branch is entered the computation is reversible to the initial state. Therefore, the thesis holds □

11

## Esperimento

- *Come si comporta N-AODV rispetto ad AODV?*
- **Quesiti di ricerca**:
  - Quale protocollo esibisce maggiore NTA?
  - Qual è il più efficace?
  - E il più efficiente?
- **Metodo**: simulazioni
- **Tool**: CoreASM
- **Modello di mobilità**: topology-based
- **Parametri**:
  - Dimensione della rete: 10, 25, 50 host
  - Livello di mobilità: alta, bassa
  - Numero di run: 500

12

## Metriche

- **Network Topology Awareness:**
  - Routing tables size
  - Routing tables updates
  - Network awareness lag
  - Broadcast activations
- **Efficacia:**
  - Total rate of success
- **Efficienza:**
  - Control overhead

13

## Formulazione delle ipotesi

- Per ogni metrica concernente NTA ed efficienza:
  - $H_0$ : **non c'è differenza** statisticamente significativa fra i due protocolli
  - $H_1$ : **c'è differenza** statisticamente significativa fra i due protocolli
- I valori relativi al total rate of success, invece, sono semplicemente **confrontati**

14

## Risultati: Routing tables size

Scenario	AODV (media)	N-AODV (media)	p-value
10 host/Alta mobilità	13.092	21.13	< 0.0001
10 host/Bassa mobilità	25.772	28.544	< 0.0001
25 host/Alta mobilità	129.948	164.78	< 0.0001
25 host/Bassa mobilità	182.272	225.424	< 0.0001
50 host/Alta mobilità	575.566	809.966	< 0.0001
50 host/Bassa mobilità	1030.224	1135.414	< 0.0001

15

## Risultati: Routing tables updates

Scenario	AODV (media)	N-AODV (media)	p-value
10 host/Alta mobilità	0.912	1.236	0.5661
10 host/Bassa mobilità	0.52	0.878	0.0016
25 host/Alta mobilità	3.404	5.182	0.0106
25 host/Bassa mobilità	2.202	3.46	0.0413
50 host/Alta mobilità	9.598	13.158	0.5598
50 host/Bassa mobilità	6.556	8.056	0.0369

16

## Risultati: Network awareness lag

Scenario	AODV (media)	N-AODV (media)	p-value
10 host/Alta mobilità	6.103	2.6136	< 0.0001
10 host/Bassa mobilità	5.3769	2.8883	< 0.0001
25 host/Alta mobilità	3.8867	2.2163	< 0.0001
25 host/Bassa mobilità	3.3082	1.5847	< 0.0001
50 host/Alta mobilità	2.2764	1.047	< 0.0001
50 host/Bassa mobilità	1.5191	1.1705	< 0.0001

17

## Risultati: Broadcast activations

Scenario	AODV (media)	N-AODV (media)	p-value
10 host/Alta mobilità	1.914	1.446	0.0002
10 host/Bassa mobilità	1.104	1.03	0.5848
25 host/Alta mobilità	4.24	3.052	0.0022
25 host/Bassa mobilità	2.722	2.226	0.0088
50 host/Alta mobilità	11.376	5.496	< 0.0001
50 host/Bassa mobilità	7.972	3.414	< 0.0001

18

## Risultati: Total rate of success

Scenario	AODV (%)	N-AODV (%)
10 host/Alta mobilità	61.6	69
10 host/Bassa mobilità	66.8	71.2
25 host/Alta mobilità	71	76
25 host/Bassa mobilità	71.4	82.4
50 host/Alta mobilità	78.2	80.6
50 host/Bassa mobilità	85.4	90

19

## Risultati: Control overhead

Scenario	AODV (media)	N-AODV (media)	p-value
10 host/Alta mobilità	1.07	1.32	0.6268
10 host/Bassa mobilità	1.05	1.27	0.3857
25 host/Alta mobilità	4.8	5.4	0.6457
25 host/Bassa mobilità	5.16	6.97	0.2728
50 host/Alta mobilità	11.15	10.768	0.0898
50 host/Bassa mobilità	8.846	7.74	0.4286

20

## Analisi

- N-AODV:
  - Fornisce agli host **maggiore** NTA rispetto ad AODV
  - È **più** efficace
  - **Non è meno** efficiente
- L'esperimento condotto soffre di **minacce alla validità**. Quali?

21

## Blackhole-freedom

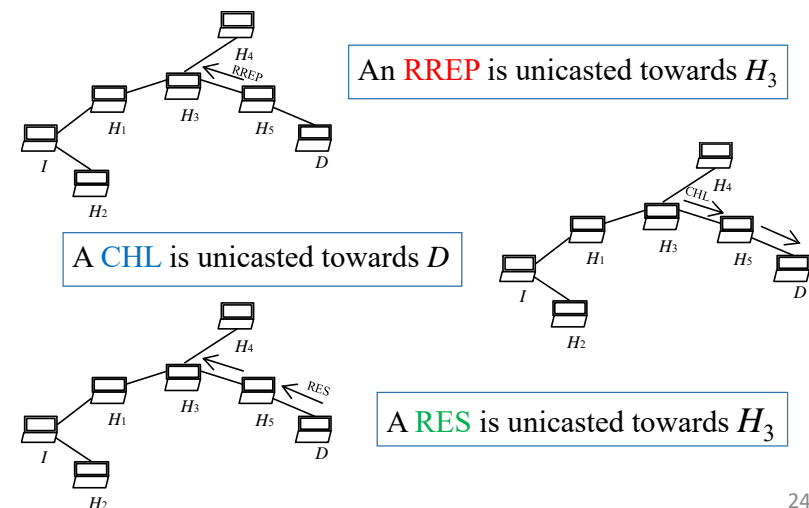
22

## Concetto generale

- Nella sua versione più semplice, un attacco blackhole consiste dei seguenti passi:
  - Il **nodo malevolo** genera un **RREP fasullo**
  - Di solito, l'RREP fasullo ha grande numero di sequenza
  - L'iniziatore sceglie la route passate per il nodo malevolo per instradare pacchetti dati
  - Il nodo malevolo li **scarta** tutti
- Nella sua versione cooperativa, il nodo malevolo è supportato da **complici**

23

## Blackhole-free AODV (B-AODV)



24

## Meccanismi aggiuntivi

- Il protocollo fa uso di **chiavi pubbliche** per cifrare i pacchetti CHL e RES
- Fa inoltre uso di **tabelle di fiducia** e **livelli di fiducia** per stabilire la reputazione di ciascun nodo

25

## Definizione formale di blackhole

- Una definizione formale e univoca di blackhole (e complice) **manca** in letteratura
- La nostra definizione:

**Definition** (forward neighbor). A *forward neighbor*  $f_n$  of a node  $n$  is the next hop of  $n$  in the route to reach destination  $d$ , i.e.  $n, f_n, \dots, d$

**Definition** (blackhole). A *blackhole* is: (i) a main blackhole if it originates fake RREPs; or (ii) a colluder if its forward neighbor is a blackhole

26

## Modellazione in ASM del blackhole

```
BlackholeProgram(hi) =
  if ¬isEmpty(requests(self)) then {
    RREQ = top(requests(self))
    previousHop = sender of RREQ
    UpdateRoutingTable(self, RREQ)
    MaliciousRouter(RREQ, previousHop)
    dequeue RREQ from requests(self)
  }

MaliciousRouter(RREQ, previousHop) =
  create RREP
  enqueue RREP into replies(previousHop)

ColluderProgram(hi) =
  if ¬isEmpty(replies(self)) then {
    RREP = top(replies(self))
    if RREP.init ≠ self then {
      previousHop = select r.nextHop ∈
        routingTable(self) with r.dest = RREP.init
      enqueue RREP into replies(previousHop)
      dequeue RREP from replies(self)
    }
  }
```

27

## Sviluppi futuri

- Implementare concretamente N-AODV su dispositivi mobili e condurre un **esperimento in vivo**
- Implementare un **meccanismo** di diffusione delle identità e delle chiavi pubbliche in una MANET  
–N-AODV può tornare utile?

28

## Riferimenti

1. Bianchi A., Pizzutilo S., Vessio G. Preliminary Description of NACK-based Ad-hoc On-demand Distance Vector Routing Protocol for MANETs. In: 9th International Conference on Software Engineering and Applications (ICSOFT-EA 2014), Vienna, Austria, pp. 500-505, SciTePress, 2014
2. Bianchi A., Pizzutilo S., Vessio G. Suitability of Abstract State Machines for Discussing Mobile Ad-hoc Networks. Global Journal of Advanced Software Engineering, 1, pp. 29-38, Avanti Publishers, 2014
3. Bianchi A., Pizzutilo S., Vessio G. Comparing AODV and N-AODV Routing Protocols for Mobile Ad-hoc Networks. In: 13th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2015), Brussels, Belgium, pp. 159-168, Association for Computing Machinery, 2015
4. Bianchi A., Pizzutilo S., Vessio G. CoreASM-based Evaluation of the N-AODV Protocol for Mobile Ad-hoc Networks. Journal of Mobile Multimedia, 12(1-2), pp. 31-51, Rinton Press, 2016