Mobile Computing non è SmartPhone: Sicurezza nei Sistemi Mobili

Contesto

- Sistemi di calcolo mobile sono tutti quei sistemi in cui qualche componente hw/sw è in grado di muoversi in uno specifico spazio
- È una definizione molto ampia che comprende sistemi molto eterogenei
- Ci concentreremo solo sui sistemi MANET

Definizione

- MANET (Mobile Ad-hoc NETwork) indica una tipologia di reti wireless che possono operare senza la necessità di una infrastruttura fisica fissa
- Permettono la comunicazione wireless tra host mobili

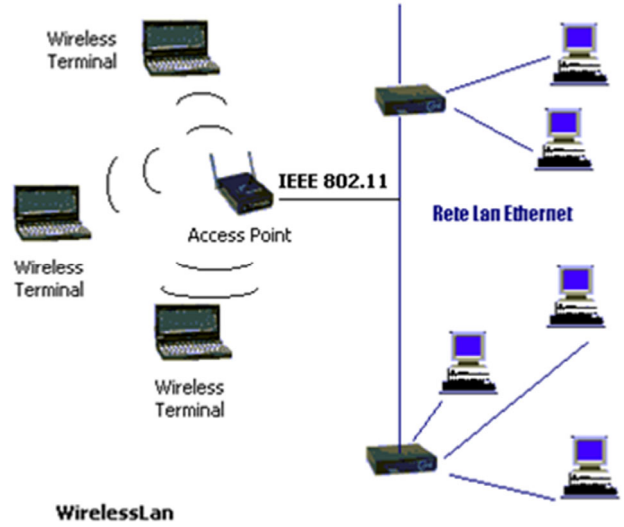
Host Mobili (1)

- In una MANET
  - le comunicazioni tra una sorgente e una destinazione sono stabilite e mantenute dalla cooperazione tra i vari host presenti nella rete
  - ogni host può agire
    - sia come end-point di una comunicazione (mittente/destinatario di msg)
    - che come router di pacchetti

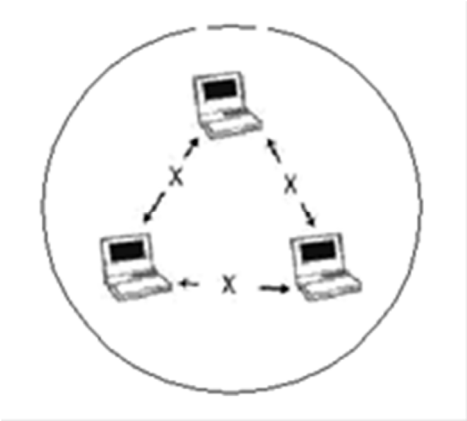
Host Mobili (2)

- Ogni host può muoversi a diverse velocità, in ogni direzione nello spazio della MANET
  - Durante la "vita" di ogni host in una MANET sia direzione che velocità cambiano continuamente
- Ogni host può comunicare direttamente solo con altri che sono all'interno di un predefinito radio range
  - Radio range è specifico per ogni host

Reti Wireless con Infrastruttura



Reti Wireless senza Infrastruttura



Dinamicità in una MANET

- Le MANET sono sistemi altamente dinamici, a causa
  - del duplice ruolo ricoperto da ogni host
  - del continuo cambiamento nella topologia della rete,
- Due aspetti della dinamica:
  - dinamica della rete: cambiamento della posizione degli host
  - dinamica del comportamento computazionale di ogni host

Applicazioni

- Le MANET sono applicate per permettere la comunicazione quando non sono disponibili infrastrutture fisiche fisse
- Ad esempio
  - squadre di soccorso nel caso di disastri
  - navi durante traversate oceaniche
  - sistemi spaziali
  - reti di sensori per il monitoraggio (fauna selvatica, frane, robot, …)
  - ...

Problemi

Problemi Tradizionali

- Le MANET richiedono di affrontare vari problemi tradizionali:
  - Il sw per l'uso e la gestione è molto complesso, spesso costituito da migliaia di moduli: sono tutti davvero necessari?
  - Esistono condizioni per eseguire tutte le componenti logiche di ogni algoritmo?
  - Esistono condizioni di deadlock/livelock?
  - Quali e quante risorse sono necessarie?
  - Cosa succede in caso di guasto?
  - Il sistema è completo? Quanto è complesso?
  - …

Problemi Specifici

- Le caratteristiche della MANET pongono vari problemi specifici:
  - Definizione e analisi delle prestazioni di protocolli di routing
  - Sincronizzazione, cooperazione e concorrenza tra le diverse componenti del sistema MANET
  - Quante comunicazioni possono essere contemporaneamente servite da un host inteso come router? Quante da un host inteso come end-point?
  - …

Soluzioni in Letteratura

- Tanti lavori in letteratura affrontano questi problemi
  - usando strumenti concettuali e computazionali derivati da altre discipline o specifici per questo caso
- Nella maggior parte dei casi lo studio fa uso di simulatori per valutare le prestazioni offerte dalla MANET

Simulatori

- Esistono diversi strumenti per simulare MANET
  - ns-2 MONARCH, OLSR, …
- La maggior parte dei quali:
  - È basata su una logica event-driven
  - La sincronizzazione è imposta artificialmente da un clock esterno
  - Non permette di descrivere formalmente il sistema

Altre Soluzioni (1)

- L'approccio seguito da altri consiste nell'analisi della formalizzazione delle MANET allo scopo di indagarne il comportamento

Altre Soluzioni (2)

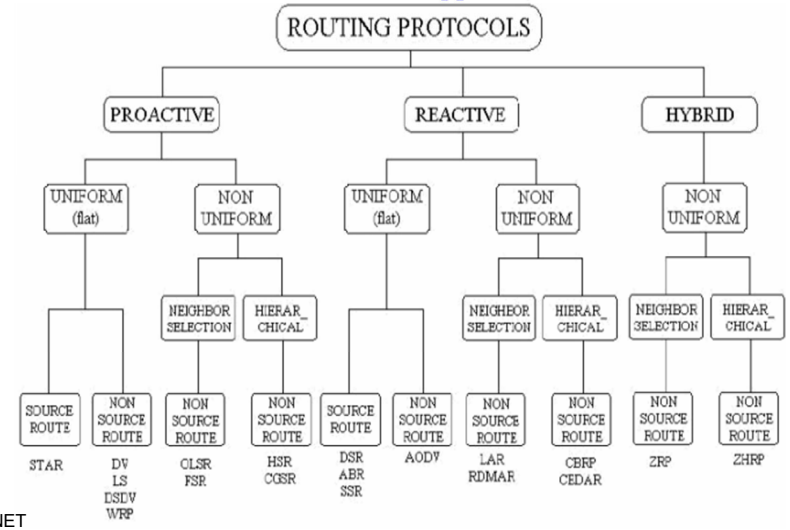
- Il gruppo di ricerca di Metodi Formali per Sistemi Distribuiti del dib affronta I problemi con un approccio ibrido
- Ha realizzato un ambiente che permette sia la simulazione che la modellazione formale delle MANET
  - in cui la sincronizzazione è stabilita dal comportamento degli host interni al sistema MANET
  - I modelli usati sono le PN (DEMONE) e le ASM (MOTION)

Protocolli di Routing per MANET

Premessa

- Esistono parecchi protocolli di routing specifici per MANET
- La principale caratteristica è la cooperazione dei vari nodi nell'instradamento di pacchetti verso la destinazione voluta
- Caratteristiche richieste ai protocolli:
  - Minimizzazione del numero di pacchetti inviati
  - Semplificazione delle attività computazionali
  - Conoscenza il più possibile aggiornata della topologia corrente della rete
  - Capacità di evitare cicli nell'invio dei pacchetti

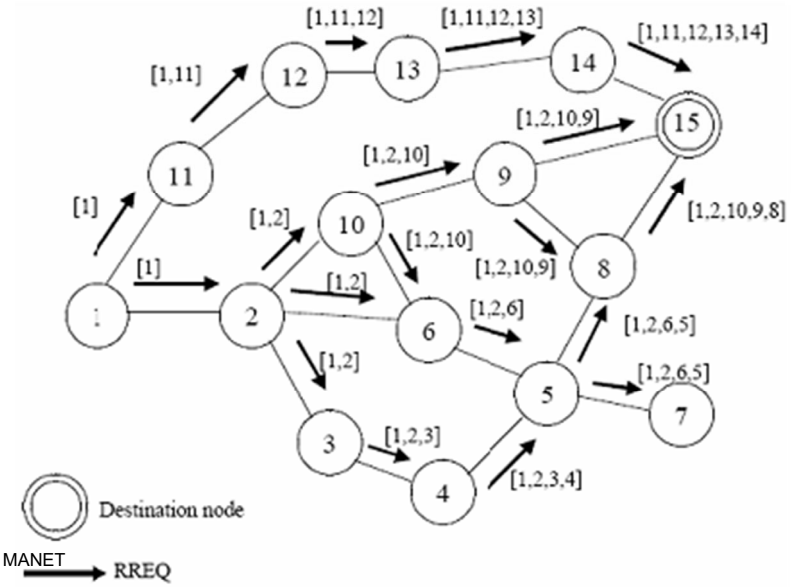
Tassonomia dei Protocolli di Routing



Dynamic Source Routing

- Initiator wants communicate with Destination
- If (Dest is a neighbour of Init) OR (a route to Dest is in Init's cache)
  - Communication can start
  - End Protocol
- Init broadcasts RREQ pck to neighbours
- Algorithm reiterated until route is found
  - RREP pck is sent back to Init

Route Discovery in DSR



Formato del Pacchetto RREQ

- Init address
- Dest address
- Request ID: integer number generated by Init – uniquely identifies RREQ
- Route record: list of visited hosts

Ad-hoc On-demand Distance Vector

- If a route to Dest is not in Init's cache AND Dest is not a neighbour of Init
  - RREQ pck is broadcasted to neighbours
  - If a node receiving RREQ is not Dest, neither knows a route to Dest, it
    - updates its info about route to Init
    - updates RREQ with its ID
    - broadcasts the updated RREQ
  - Else it
    - unicasts a RREP pck back to Init

Routing Table

- Ogni nodo possiede una propria routing table in cui ogni riga è associata a un nodo Dest conosciuto
- Per ogni riga sono registrate:
  - Dest IP address
  - Dest Sequence Number: stabilisce quando sono state ricevute le info su Dest
  - HopCount: distanza di Dest
  - NextHop: nodo successivo per raggiungere Dest

Formato del Pacchetto RREQ

- Contiene:
  - BroadcastID: numero intero associato ad ogni nodo
  - HopCount: inizialmente vale 0 ed è incrementato a ogni hop
  - DestIPAddress / InitIPAddress
  - DestSequenceNumber / InitSequenceNumber: I seq nums più aggiornati

Sicurezza nelle MANET

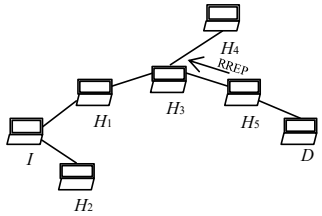
MANET = Open System

- Data la mobilità degli host e l'assenza di host di controllo è molto difficile implementare politiche di sicurezza
- MANET è molto sensibile a molti tipi di attacchi
- Ci concentreremo sull'esempio del BlackHole Attack

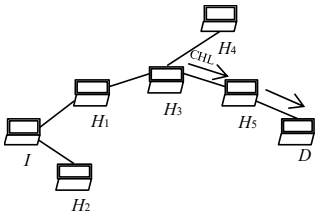
Concetto generale

- 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

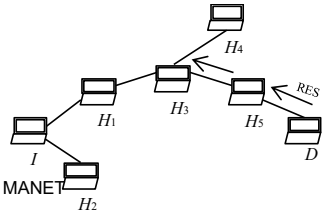
Blackhole-free AODV (B-AODV)



An RREP is unicasted towards H_3



A CHL is unicasted towards D



A RES is unicasted towards H_3

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

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. $n, f_n, \ldots, d$

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

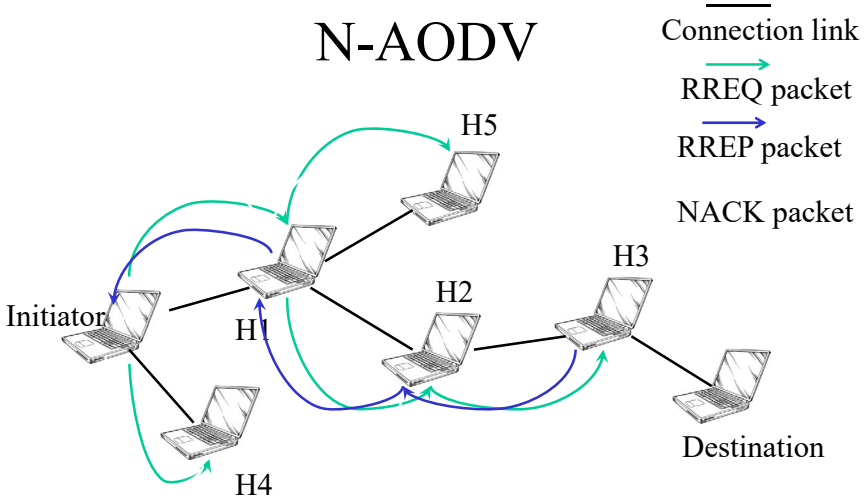
Blackhole-free N-AODV

Our research

- Formal specification of Blackhole-free N-AODV routing protocol – BN-AODV
  - The Ad-hoc On-demand Distance Vector (AODV) protocol is one of the most popular routing protocol for MANETs (*)
  - We proposed a variant of the protocol: the NACK-based AODV (N-AODV) (#); it improves the network topology awareness (°)
  - We now enrich N-AODV with the capability to intercept blackholes
- The protocol is formally specified by means of Abstract State Machines (ASMs)

(*) C.E. Perkins et al. http://tools.ietf.org/html/rfc3561 (2003)
(#) A. Bianchi et al. Preliminary Description of NACK-based AODV for MANETs. ICSOFT (2014)
(°) A. Bianchi et al. CoreASM-based Evaluation of the N-AODV Protocol for Mobile Ad-hoc NETworks. J. of Mobile Multimedia, (2016)

N-AODV

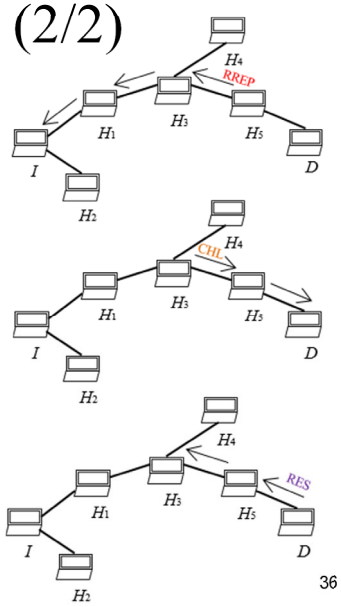


BN-AODV (1/2)

- The main idea is that each host receiving an RREP must verify the trustworthiness of the next host to dest
- It adds two control pcks:
  - Challenge (CHL) – is an encrypted nonce to be decremented by the destination
  - Response (RES) – is the encrypted decremented value

BN-AODV (2/2)

- H_3 receives an RREP from H_5
- H_3 unicasts CHL to D through H_5
  - If H_3 receives back the correct RES, then H_5 is considered trusted
  - Else H_5 is considered a blackhole



ASM model of BN-AODV

Generality

- The MANET is modeled by a DASM, including a set of Hosts=$\{h_1, h_2, \ldots, h_n\}$
  - each h_i models the behavior of the i-th host executing the protocol
- Each host behaves either as an honest node, or as a blackhole, or as a colluder
  - No greyhole
- Three ASMs describe the three different

Honest ASM - HostProgram

```
if ¬isEmpty(requests(self)) then {
let RREQ = top(requests(self)), previousHop = RREQ.sender in
  UpdateRoutingTable(self, RREQ)
  Router(RREQ, previousHop)
  dequeue RREQ from requests(self)
}
if wishToInitiate(self) = true then
 forall dest ∈ Hosts with dest ≠ self do
  if initiateTo(self, dest) = true then
   Initiator(dest)
if ¬isEmpty(replies(self)) then
 let RREP = top(replies(self)), nextHop = RREP.sender in
  if RREP.Init ≠ self then {
   choose entry ∈ routingTable(self) with entry.dest = RREP.init
   previousHop := entry.nextHop seq
   hasToVerify(previousHop, nextHop, RREP.dest):=true
  }
forall previousHop ∈ neighb(self) do
 forall nextHop ∈ neighb(self) do
  forall dest ∈ Hosts do
   if hasToVerify(previousHop, nextHop, dest) then
    Verify(top(replies(self)), previousHop, nextHop, dest)
if ¬isEmpty(responses(self)) then {
 let RES = top(responses(self)) in
  if RES.dest ≠ self then {
```

```
if ¬isEmpty(challenges(self)) then {
 let CHL = top(challenges(self)) in
  if CHL.dest = self then {
   let previousHop = CHL.sender in
    create_RES seq
    enqueue RES into responses(previousHop)
    dequeue CHL from challenges(self)
  }
  if CHL.dest ≠ self then {
   choose entry ∈ routingTable(self) with entry.dest = CHL.dest
   nextHop := entry.nextHop seq
   enqueue CHL into challenges(nextHop)
   dequeue CHL from challenges(self)
  }
}
if ¬isEmpty(nacks(self)) then {
 let NACK = top(nacks(self)) in
  if NACK.dest ≠ self then {
   choose entry ∈ routingTable(self) with entry.dest = NACK.dest
   previousHop := entry.nextHop seq
   UpdateRoutingTable(self, NACK)
   enqueue NACK into nacks(previousHop)
   dequeue NACK from nacks(self)
  }
}
```


Honest ASM - Initiator

```
if dest ∈ neighb(self) ∨ dest ∈ routingTable(self) then {
 CommunicationSession(dest)
 initiateTo(self, dest) := false
}
if dest ∉ neighb(self) ∧ dest ∉ routingTable(self) then {
 create_RREQ seq
 BroadcastRREQ(RREQ)
 initiator_waiting(self, dest) := true
 initiator_timeout(self, dest) := default_value
}
if initiator_waiting(self, dest) then
initiator_timeout(self,dest):=initiator_timeout(self,dest)
−1 seq
 if ¬isEmpty(replies(self)) then
  forall r∈replies(self) with r.init=self and r.dest =dest
   if trusted(self, dest) then {
    CommunicationSession(dest)
    initiateTo(self, dest) := false
    initiator_waiting(self, dest) := false
   }
   if initiator_waiting(self, dest) ∧
¬isEmpty(nacks(self)) then {
    forall n ∈ nacks(self) with n.dest = self do {
     UpdateRoutingTable(self, n)
     dequeue n from nacks(self)
    }
   }
   if initiator_waiting(self, dest) ∧
initiator_timeout(self, dest)=0
   then {
    initiateTo(self, dest) := false
    initiator_waiting(self, dest) := false
```

Honest ASM - Verify

```
if ¬verify_waiting(self, dest) then {
  create_CHL seq
  enqueue CHL into challenges(nextHop)
  verify_waiting(self, dest) := true
  verify_timeout(self, dest) := default_value
 }
 if verify_waiting(self, dest) then {
  if ¬isEmpty(responses(self)) then
   if ReliableRREP(self, top(responses(self))) then {
    trusted(self, dest) := true
    verify_waiting(self, dest) := false
    dequeue top(responses(self)) from responses(self)
   }
  verify_timeout(self, dest) := verify_timeout(self, dest) – 1
 }
 if verify_waiting(self, dest)∧verify_timeout(self, dest) =0 then {
  trusted(self, dest) := false
  verify_waiting(self, dest) := false
 }
 if trusted(self, dest) then {
  UpdateRoutingTable(self, RREP)
  UpdateTrustTable(self, nextHop)
  dequeue RREP from replies(self)
  if previousHop ≠ null then
   enqueue RREP into replies(previousHop)
 }
 if ¬trusted(self, dest) then {
  UpdateTrustTable(self, nextHop)
  dequeue RREP from replies(self)
 }
```


Malicious ASM - Blackhole Program

```
if ¬isEmpty(requests(self)) then
  let RREQ = top(requests(self)), previousHop = RREQ.sender in
   UpdateRoutingTable(self, RREQ)
   MaliciousRouter(RREQ, previousHop)
   dequeue RREQ from requests(self)
 }
```

where the MaliciousRouter submachine is simply:

```
MaliciousRouter(RREQ, previousHop) ≡
 create_RREP seq
 enqueue RREP into replies(previousHop)
```


Malicious ASM – Colluder Program

```
if ¬isEmpty(replies(self)) then{
  let RREP = top(replies(self)) in
   if RREP.init ≠ self then {
    choose entry ∈ routingTable(self) with entry.dest = RREP.init
    let previousHop := entry.nextHop seq
     enqueue RREP into replies(previousHop)
    dequeue RREP from replies(self)
   }
 }
```


Correctness (1/2)

- *The honest hosts intercept any single blackhole attack*
  - Let $n_0, n_1, \ldots, n_{k-1}, n_k, b$ be the route from the initiator to the blackhole
  - We prove that any fRREP is discarded by n_k, i.e. fRREP is not enqueued to n_{k-1}
  - The only rule allowing n_k to enqueue fRREP to n_{k-1} is guarded by a condition that verifies the trustworthiness of the route (red rule in Verify)

Correctness (2/2)

- *The honest hosts intercept any cooperative blackhole attack*
  - Let's consider the worst case: n_0, c_1, c_2, …, c_k, b
  - We prove that if fRREP is enqueued to n_0 the communication does not start
  - In fact the communication starts (blue rule in Initiator) only after n_0 has verified the trustworthiness of the route
  - If at least one honest node is between n_0 and b is a

Contribution

- From the application domain viewpoint:
  - Proposal of BN-AODV
  - Thanks to ASMs, the ability of the protocol to intercept all attacks is proved
- From the ASM viewpoint:
  - Application to a complex, real, up-to-date case study

Future work

- Investigation about greyhole
- Simulations for mitigating
  - the draconian approach of BN-AODV
  - the high overhead
- Implementation in ASMETA