Università degli Studi di Bergamo
Dipartimento di Ingegneria dell'Informazione e Metodi Matematici

Wireless Multihop Networks: Routing & Security

Wireless Networks: Ciclo di Seminari

Ing. Stefano Paris

Introduzione

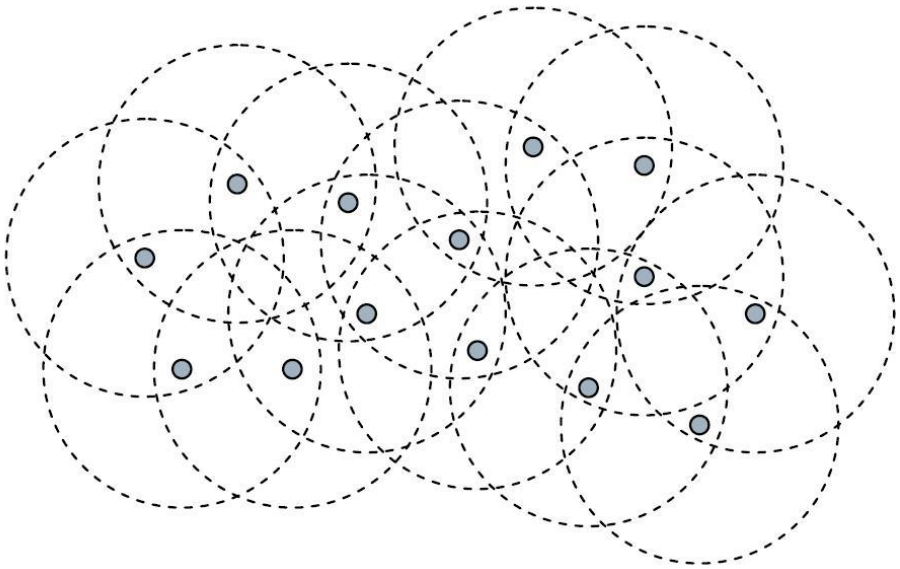
- La flesibilità fornita dalla tecnlogia wireless ha permesso lo sviluppo di nuovi paradigmi di rete
  - Mobile Ad-hoc NETworks (MANETs)
  - Wireless Mesh Networks (WMNs)
  - Vehicular Networks (VANETs)
- A differenza delle MANETs e delle VANETs, le WMNs sono caratterizzate da infrequenti cambiamenti di tolpologia di rete

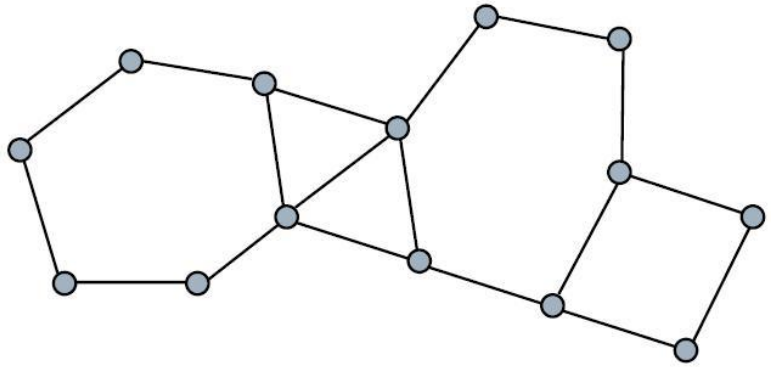
MANETs

- Caratteristiche delle MANETs
  - Reti composte da nodi mobili
  - I nodi sono equipaggiati con interfaccie di comunicazione wireless
  - Nessuna infrastruttura preesistente
  - La comunicazione tra due dispositivi della rete coinvolge più nodi intermedi
- Implicazioni
  - I dispositivi agiscono sia come hosts sia come routers
  - La topologia di rete cambia dinamicamente

MANETs

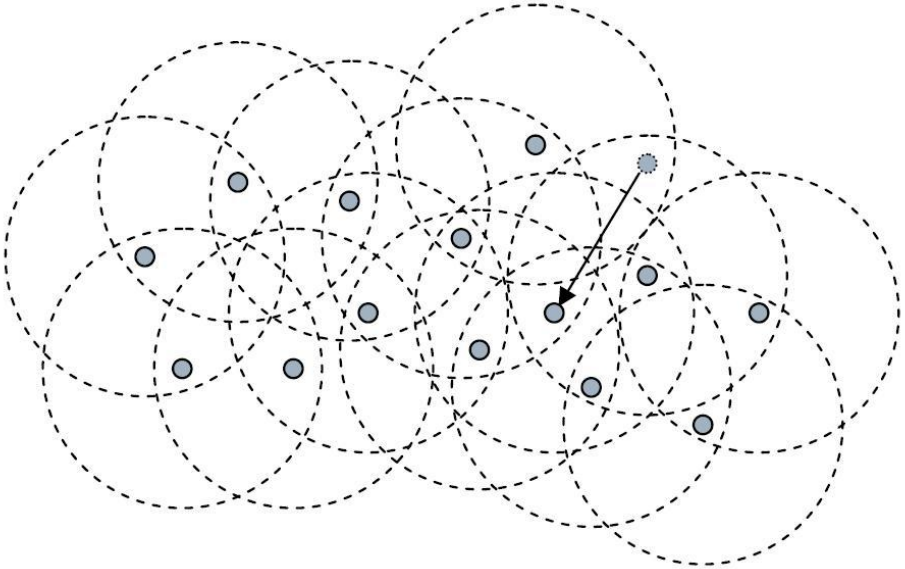
- Ogni nodo può comunicare direttamente con un sottoinsieme dei nodi di rete (i propri vicini)
  - Il range di comunicazione varia in base alle condizioni del mezzo fisico
  - Se si utilizzano antenne omnidirezionali il range di comunicazione può essere approssimato con un cerchio

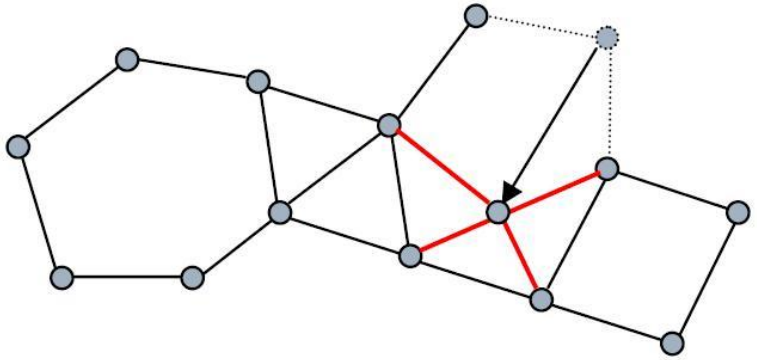




MANETs

- La mobilità provoca modifiche alla topologia di rete
  - Tali modifiche provocano un cambiamento delle decisioni di inoltro dei nodi intermedi
  - Adattamento real time al tali modifiche



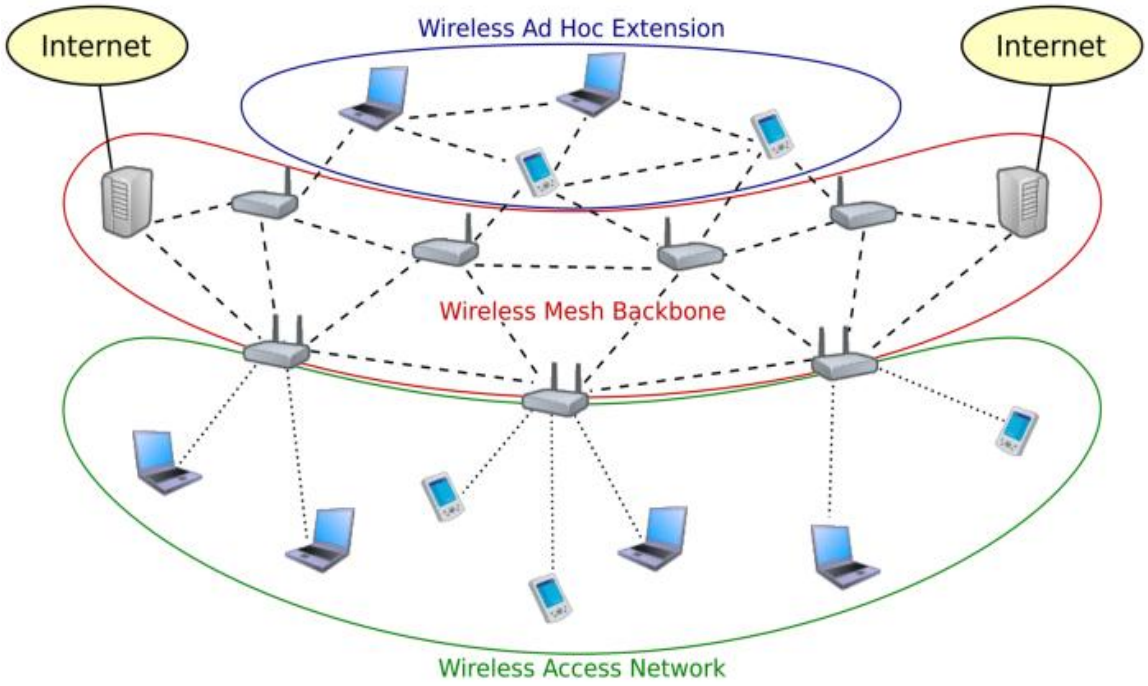


WMNs

- Caratteristiche delle WMNs
  - Reti composte essenzialmente da due tipologie di nodi
    - Mesh Clients (es. Laptop, PDA, etc.)
    - Mesh Routers: dispositivi che forniscono accesso e inoltrano il traffico dei clients
  - Infrastruttura di rete preesistente tra mesh routers
  - I nodi sono equipaggiati con interfaccie di comunicazione wireless
  - La topologia di rete è meno soggetta a cambiamenti

WMNs

- Esistono tre principali architetture di rete:
  - Infrastrutturate
  - Ad Hoc
  - Ibride



Università degli Studi di Bergamo
Dipartimento di Ingegneria dell'Informazione e Metodi Matematici

Mesh Networking

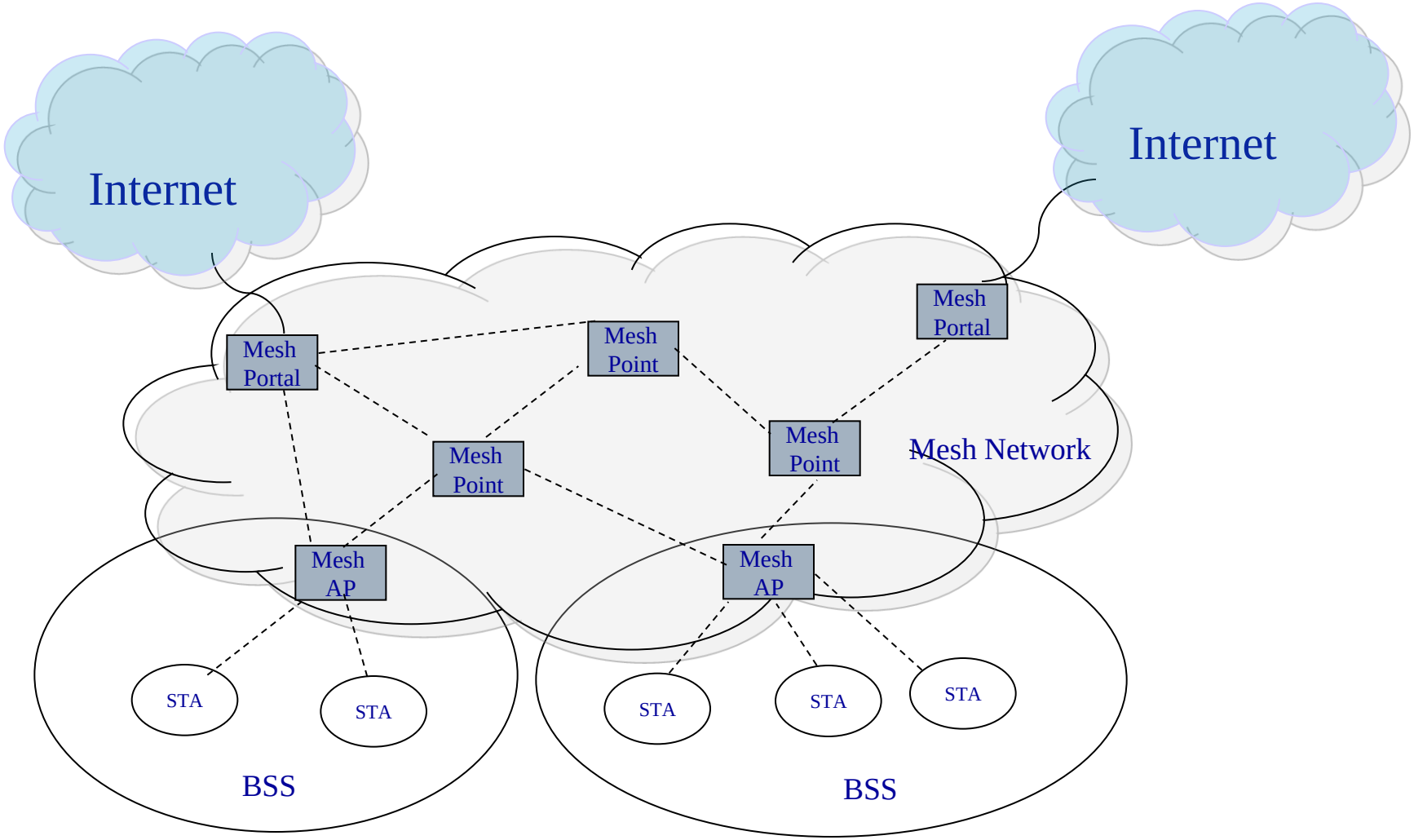
IEEE 802.11s

Soluzioni commerciali

Mesh Networking e 802.11

- Obiettivi
  - Estendere le dimensioni degli *hot spot* 802.11 tramite un'infrastruttura di tipo *mesh*
  - Ampliare gli scenari applicativi della tecnologia WLAN
- Soluzioni
  - Infrastrutture decentralizzate
  - Reti magliate di *Infrastructure BSS* con gli AP connessi tramite un sistema di distribuzione *wireless*

Esempio di rete *Mesh*



Scenari Applicativi

- Accesso residenziale (concorrenza con WiMax)
- Uffici
- Reti pubbliche di accesso ad internet
- Reti pubbliche di sicurezza
- Reti militari

Il mercato delle reti *Mesh*

- Applicazioni residenziali
  - Indoor
  - Dimensioni ridotte
  - Coesistenza con altre reti
- Applicazioni Business
  - Indoor
  - Dimensioni ridotte
  - Complessità (e quindi costo) maggiore
- Campus/Reti cittadine/Accesso pubblico
  - Connettività su ampie aree geografiche
  - Scalabilità
  - Riconfigurabilità
- Applicazioni Militari

Standardizzazione

- Il TG 802.11s ha lo scopo di definire un *Extended Service Set (ESS)* per supportare servizi b*roadcast/multicast* ed *unicast* in reti *multihop.*
- Draft 1.0 Novembre 2006
- Draft 2.0 Marzo 2008
- Draft 3.0 Marzo 2009

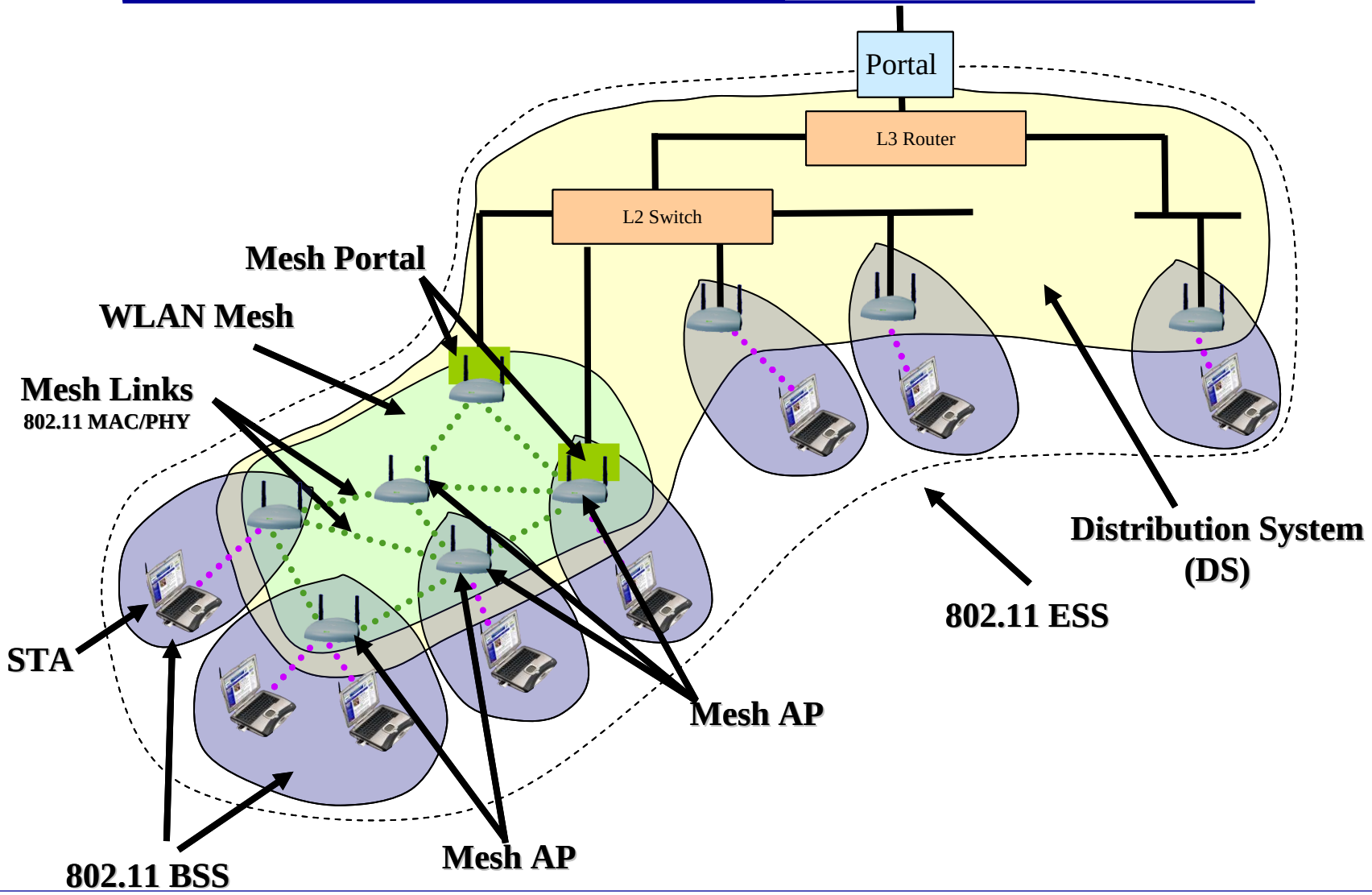
802.11s

- Routing robusto ed efficiente:
  - *Mesh Topology Learning,*
  - *Routing and Forwarding*
- Sicurezza:
  - *Compatibilità con 802.11x*
- Flessibilità del livello MAC
  - *Mesh Measurement*
  - *Mesh Discovery and Association*
  - *Mesh Medium Access Coordination*
  - *Supporto alla QoS*
- Trasparente ai livelli superiori
- Compatibile con dispositivi *legacy*

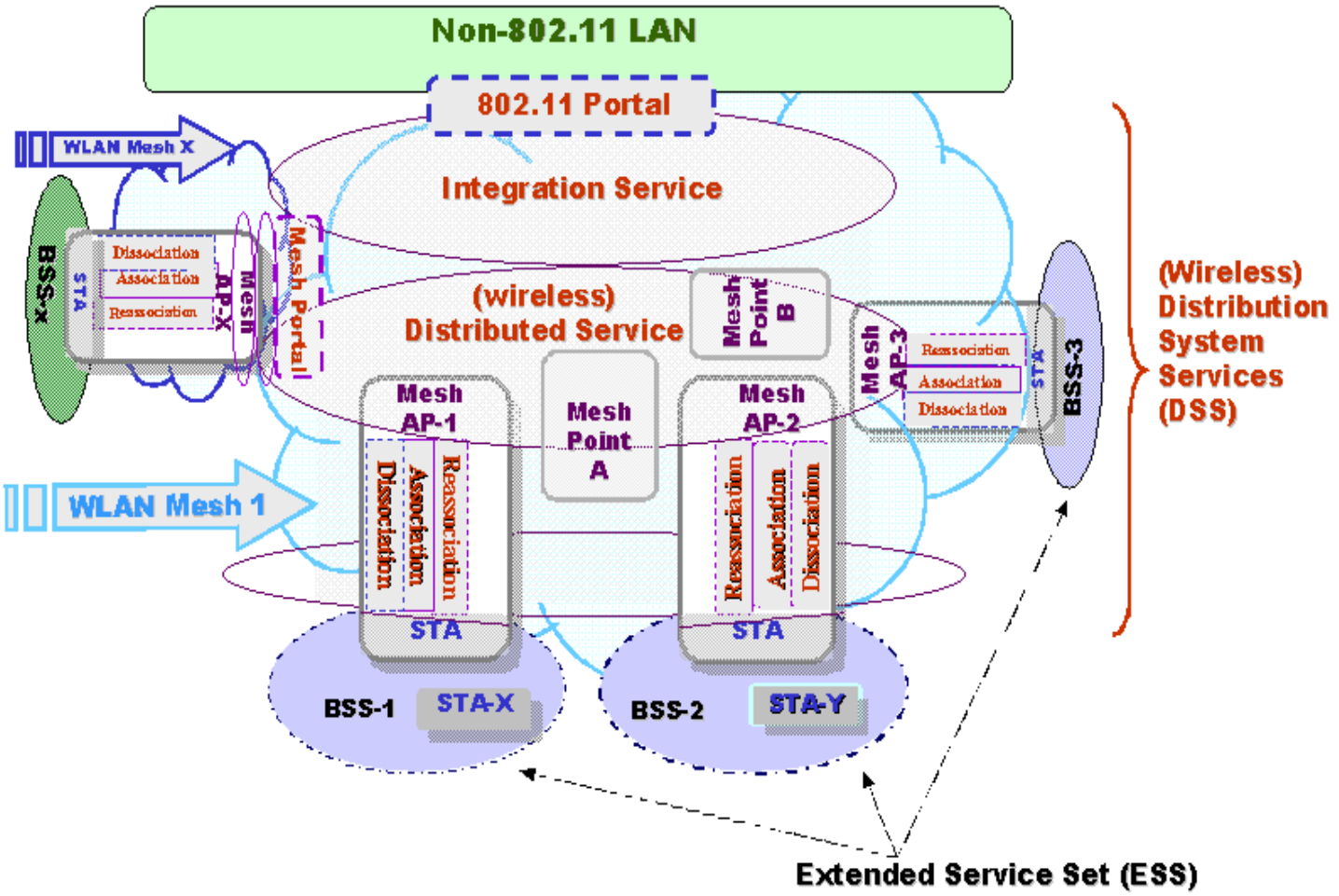
802.11s

- Routing: Hybrid Wireless Mesh Protocol (HWMP) – combinazionbe di AODV e protocollo tree-based
- Applicazioni:
  - OLPC (One Laptop Per Child)
  - Open802.11s

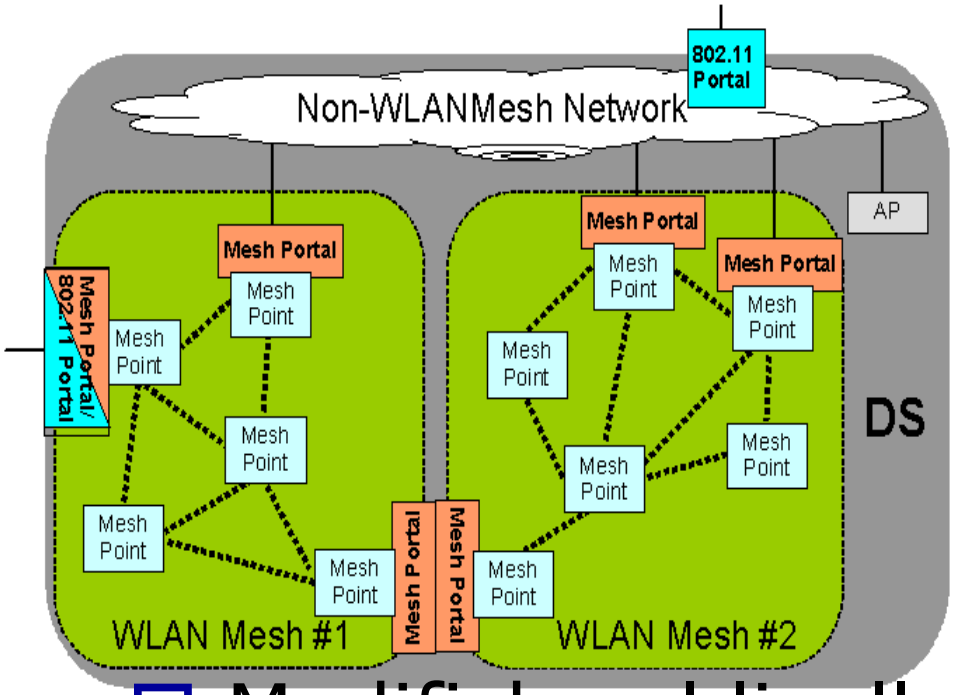
Una rete *Mesh*

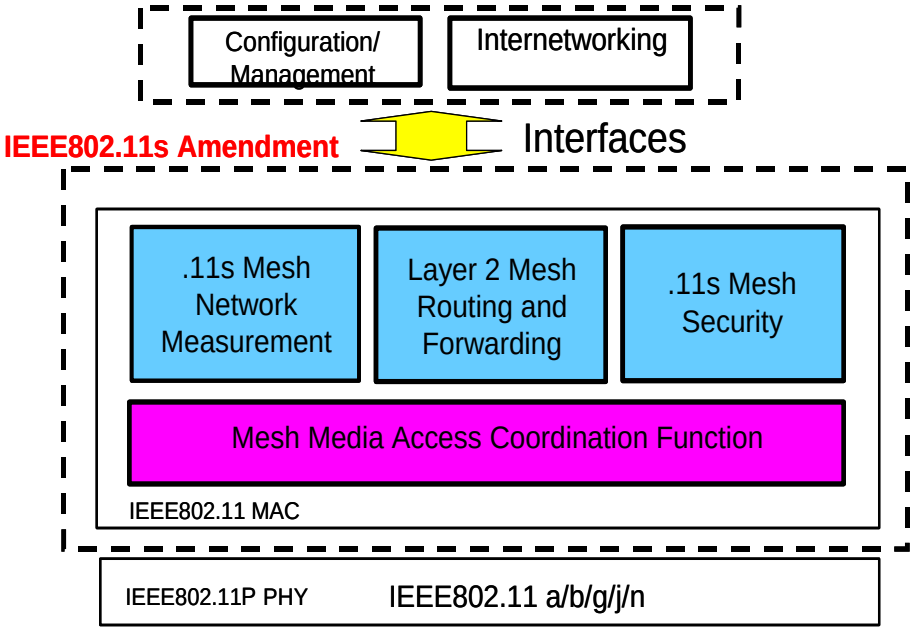


Architettura di una rete Mesh



Architettura e Protocolli





- Modifiche al livello MAC e al livello di routing
- Nessuna modifica al livello fisico

Soluzioni "Off the Shelves"

- Molte aziende producono già dispositivi per l'implementazione di reti *mesh*:
  - *Motorola* (*MeshNetworks*^{TM}): *MeshNetworks Enabled Appliances* (MEA)
  - *Tropos Networks (*802.11-*compliant)*
  - *Nortel (*802.11-*compliant)*
- Tutte le soluzioni commerciali forniscono l'hardware e il software (proprietario) per l'implementazione delle reti *Mesh*

Università degli Studi di Bergamo
Dipartimento di Ingegneria dell'Informazione e Metodi Matematici

Routing in Wireless Multihop Networks

Introduzione

- I protocolli di routing convenzionali sviluppati per reti statiche (o che variano lentamente) non sono adatti alle reti wireless multihop
  - La dinamica dei cambiamenti di rete è più rapida del tempo necessario a tali protocolli per convergere
  - Spreco di risorse limitate
  - Non sono ottimizzati per specifici obiettivi (risparmio energetico)
- Sono stati sviluppati numerosi protocolli di routing per reti Ad-Hoc riutilizzati anche da WMNs e VANETs
- Non esiste un protocollo ottimo per tutti i tipi di rete e/o condizioni

Classificazione dei Protocolli di Routing

- Reactive Protocols
  - Determinano il percorso on-demand
- Proactive Protocols
  - Mantengono un percorso per ogni possibile destinazione indipendentemente dalle condizioni di traffico
- Hybrid Protocols
  - Percorsi locali in modo proattivo
  - Percorsi lunghi in modo reattivo
- Geographical Protocols
  - Sfruttano le informazioni geografiche dei nodi di rete

Classificazione dei Protocolli di Routing

- Reactive Protocols
  - Causano alti ritardi di trasmissione tra la richiesta di trasmisisone del primo pacchetto e la sua consegna
  - Basso overhead in scenari di rete con poco traffico
- Proactive Protocols
  - I pacchetti sono consegnati immediatamente poiché i percorsi sono aggiornati di continuo
  - Alto overhead per l'aggiornamento dei percorsi di rete
- Hybrid Protocols
  - Operano in modo tale da garantire un buo trade-off tra ritardo e overhead di segnalazione

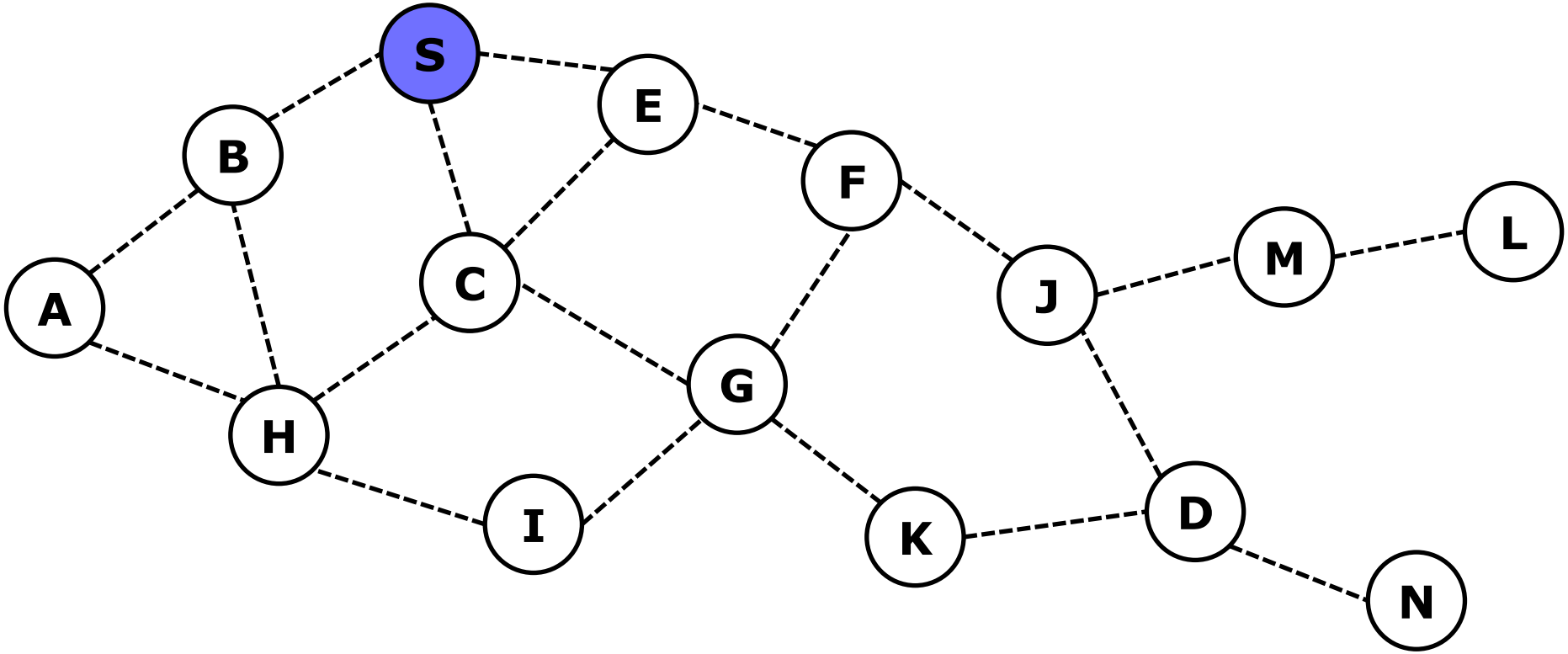
Trade-Off

- Ritardo della procedura di route discovery:
  - I protocolli proattivi possono fornire una latenza minore poiché i percorsi sono aggiornati di continuo
  - I protocolli reattivi possono causare ritardi di consegna più elevati poiché il percorso viene selezionato solo quando è necessario
- Overhead di segnalazione e aggiornamento dei percorsi
  - I protocolli reattivi possono generare meno traffico di segnalazione poiché la fase di discovery del percorso ottimo è effettuata solo se necessario
  - I protocolli proattivi possono causare un più alto overhead di segnalazione poiché i percorsi sono aggiornati di continuo

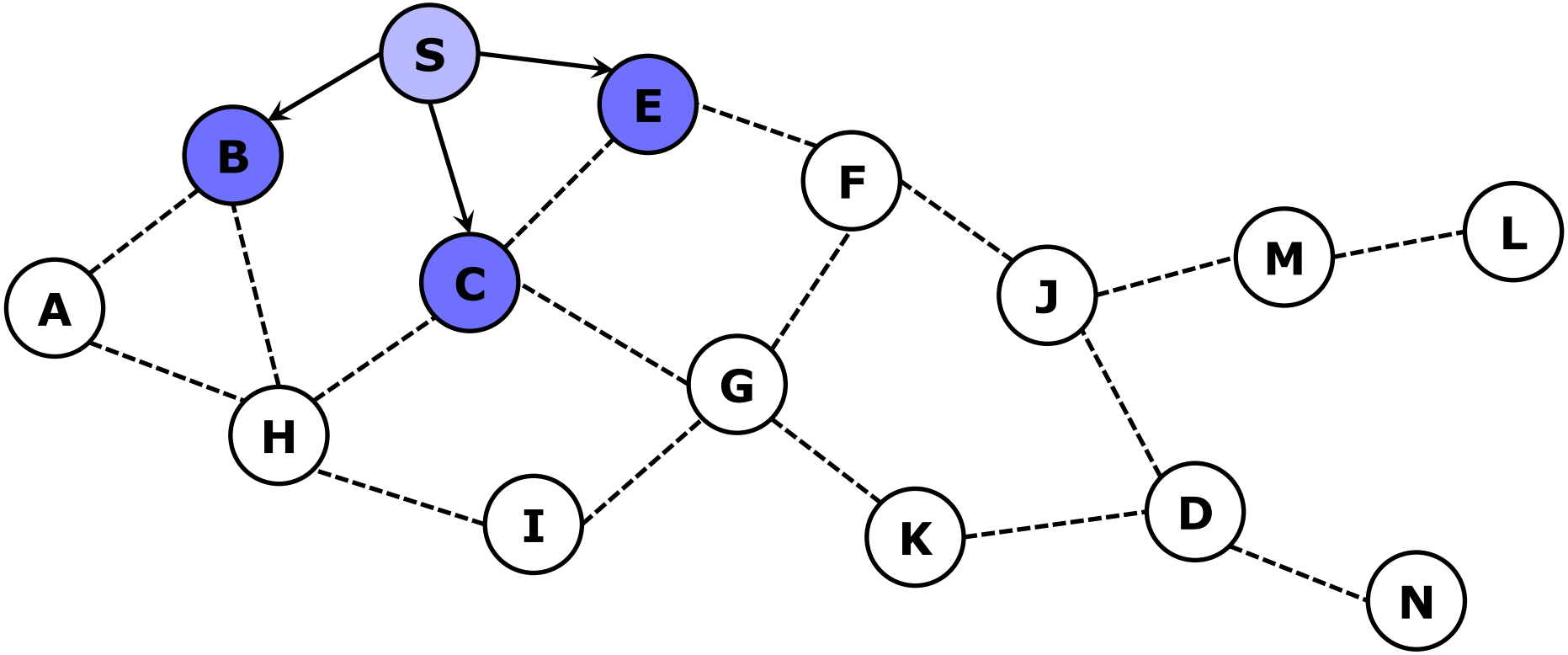
Flooding

- Il nodo sorgente S trasmette il pacchetto P a tutti i suoi nodi vicini
- Ogni nodo che riceve il pacchetto P lo ritrasmette ai suoi vicini
- Numeri di sequenza sono utilizzati per evitare di ritrasmettere pacchetti già precedentemente inoltrati
- Il pacchetto P raggiunge la destinazione D, se D è raggiungibile
- Il nodo D non ritrasmette P

Flooding



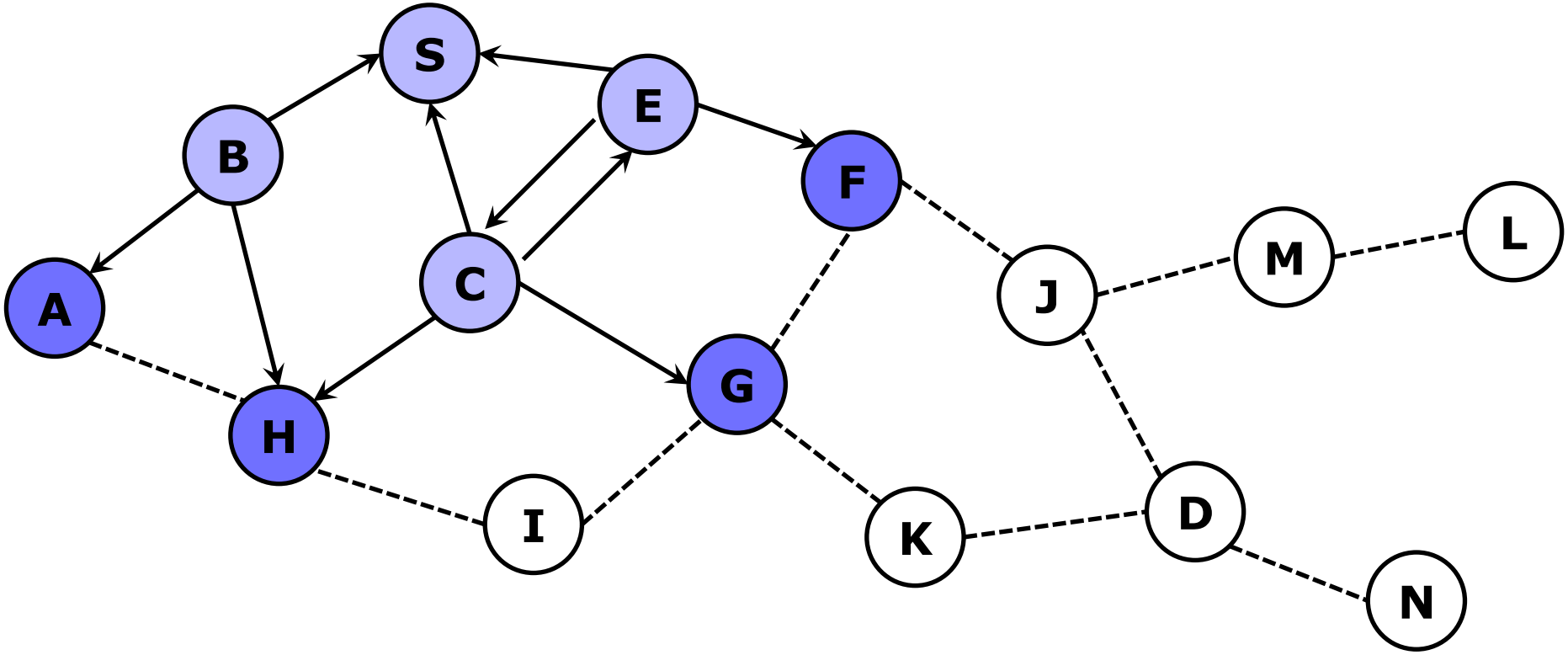
Flooding



Rappresenta un nodo che ha ricevuto il pacchetto P

Rappresenta la trasmissione del pacchetto P

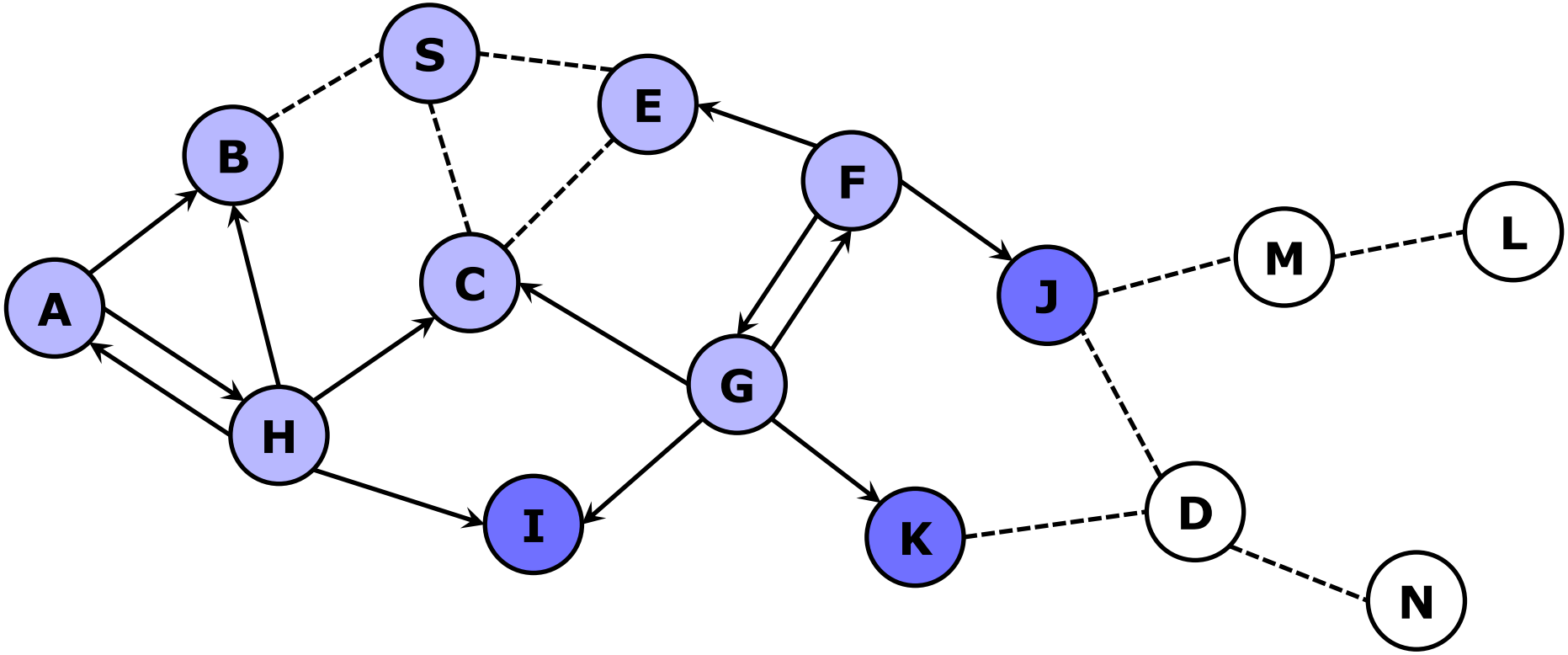
Flooding



Il nodo H riceve il pacchetto da due nodi (C e B)

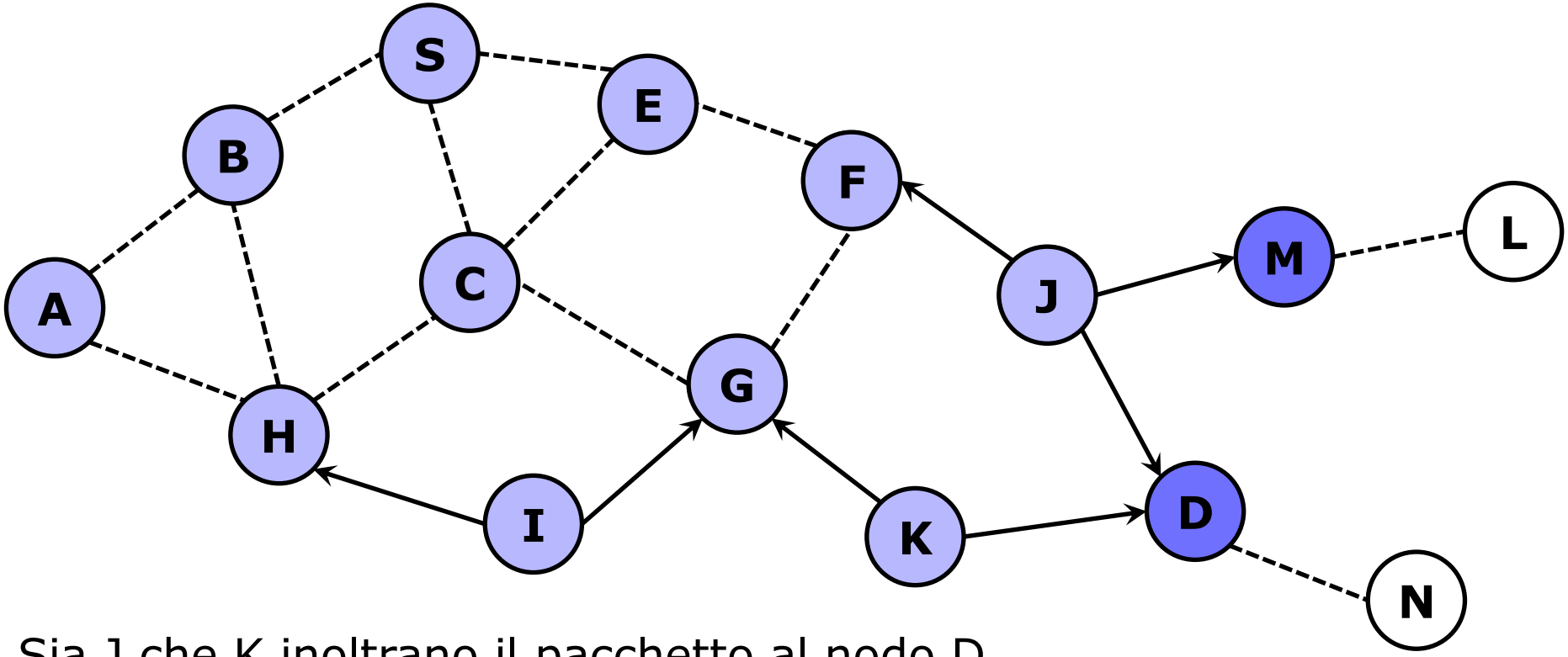
- Possibile ***collisione***

Flooding



Il nodo C riceve il pacchetto da G e H, ma non lo ritrasmette poiché ha già inoltrato il pacchetto P

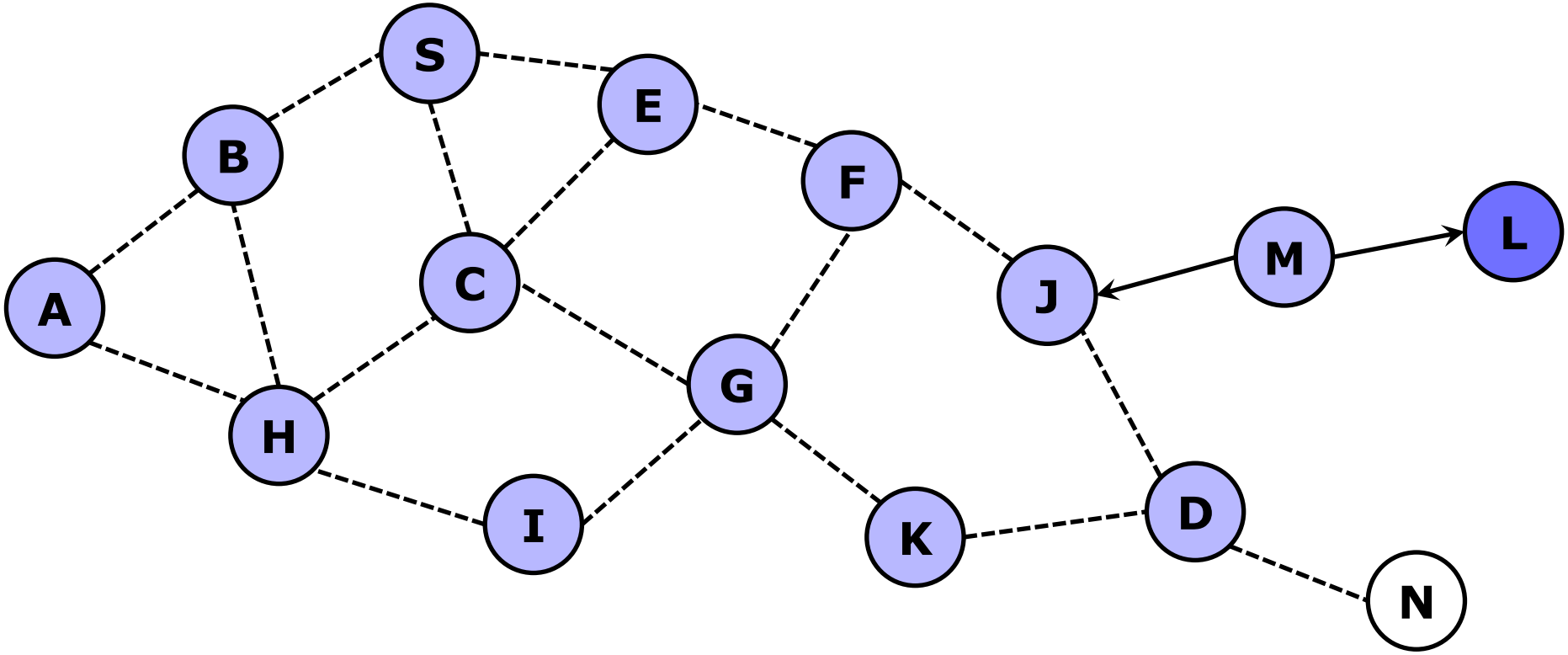
Flooding



Sia J che K inoltrano il pacchetto al nodo D
Poiché i nodi J e K sono nascosti, la loro trasmissione può collidere

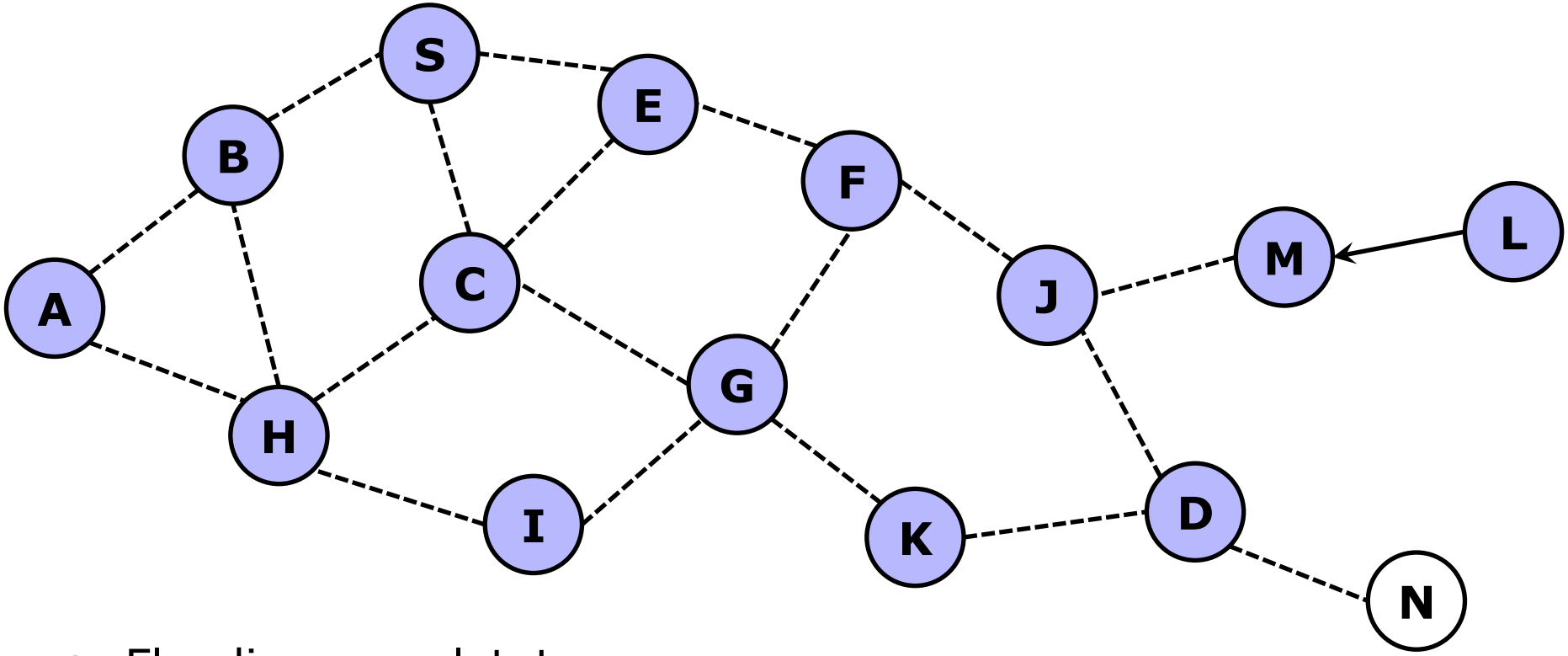
- Il pacchetto P potrebbe **non essere consegnato** a D, nonostante il flooding

Flooding



Il nodo D non ritrasmette il pacchetto P, poiché è la destinazione

Flooding



- Flooding completato
- Attraverso il Flooding i pacchetti vengono consegnati a molti nodi (nel caso peggiore tutti i nodi della rete sono raggiunti dal pacchetto)

Flooding: Svantaggi

- Può causare un alto overhead di segnalazione
  - I pacchetti vengono trasmessi a molti nodi a cui tali pacchetti non servono
- Il meccanismo di consegna può risultare poco affidabile
  - Il Flooding utilizza la trasmissione broadcast
    - E' molto complesso realizzare un meccanismo affidabile di consegna boradcast
    - In 802.11 il broadcast è inaffidabile
- Nell'esempio i nodi J e K potrebbero trasmettere contemporaneamente causando una collisione al ricevitore D
  - In questo caso il pacchetto non viene consegnato

Flooding di pacchetti di Controllo

- Molti protocolli eseguono il flooding dei pacchetti di controllo
- I pacchetti di controllo sono utilizzati per scoprire i individuare di rete
- I percorsi individuati utilizzando i pacchetti di controllo vengono utilizzati per la consegna dei pacchetti di dati
- L'overhead dei pacchetti di controllo è ammortizzato

Università degli Studi di Bergamo
Dipartimento di Ingegneria dell'Informazione e Metodi Matematici

Reactive Protocols

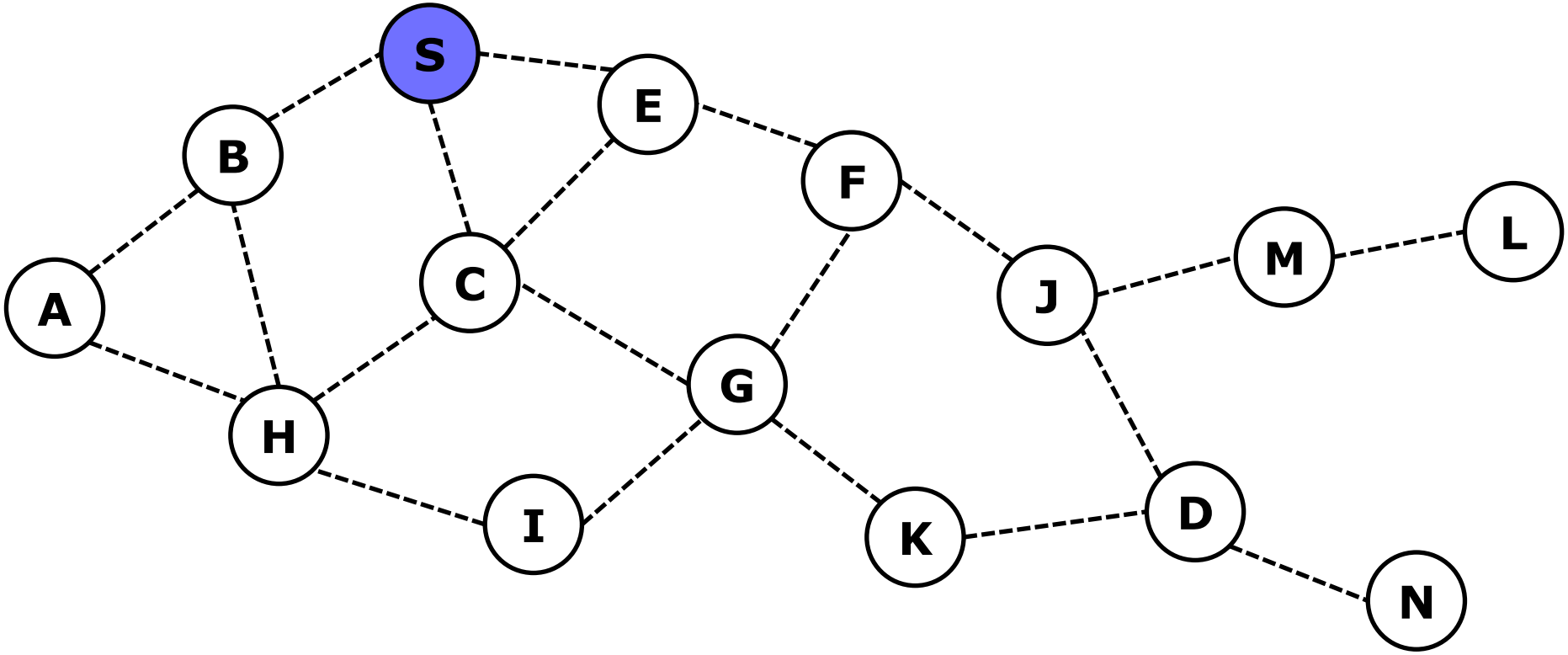
Dynamic Source Routing

Ad-Hoc On-demand Distance Vector

Dynamic Source Routing (DSR)

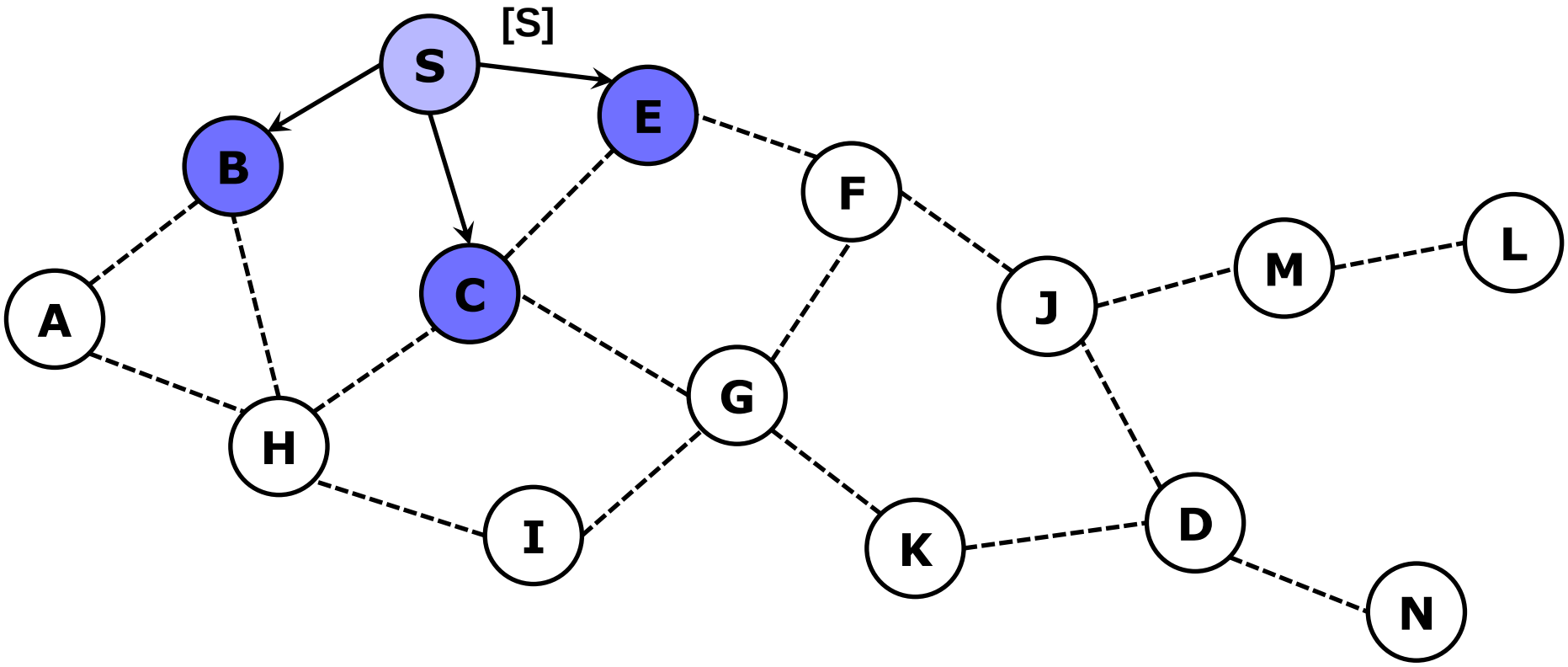
- Quando un nodo S vuole inviare uno o più pacchetti a un nodo D, senza conoscere il percorso verso la destinazione, inizia la fase di **Route Discovery**
- Il nodo sorgente trasmette attraverso il flooding una **Route Request** (**RREQ**)
- Ogni nodo **appende** il **proprio identificativo** alla RREQ prima di reinoltrarla

Route Discovery in DSR



Rappresenta un nodo che ha ricevuto la RREQ per D da S

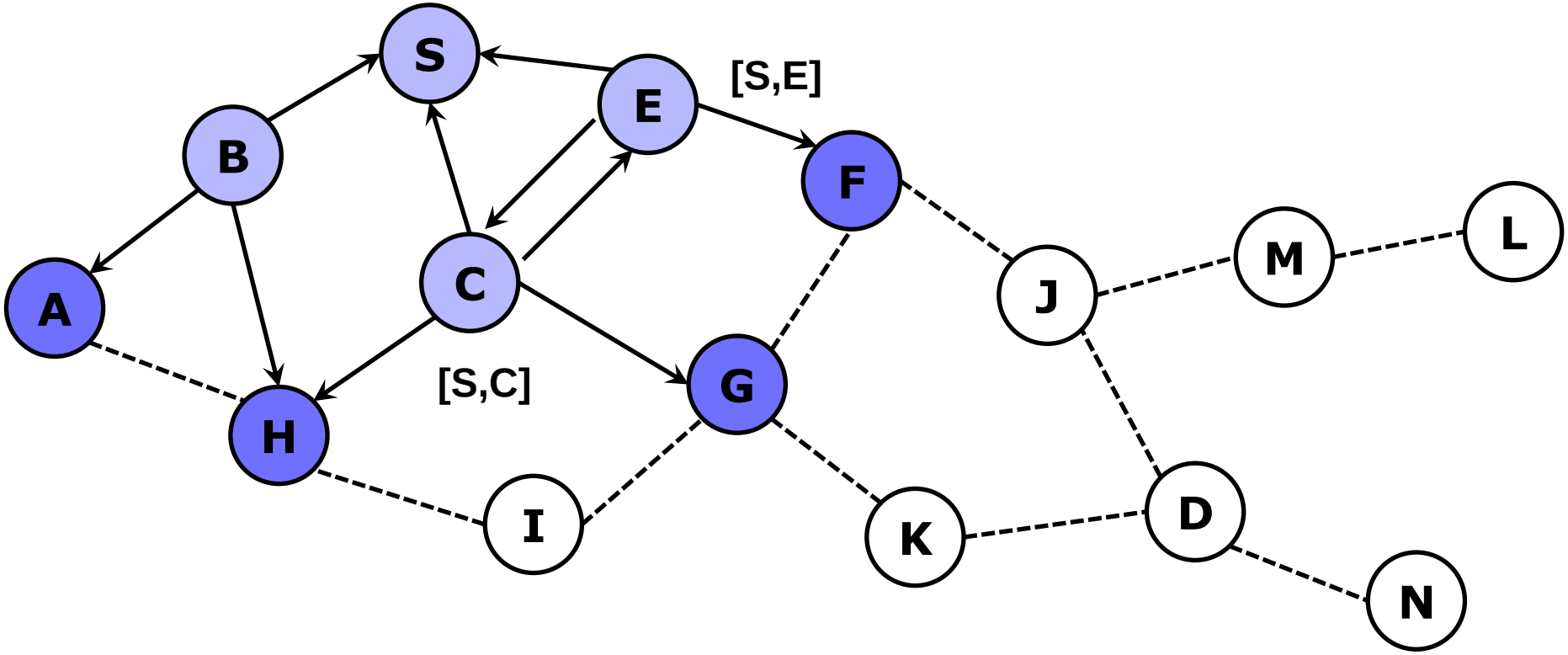
Route Discovery in DSR



[X,Y] Rappresenta la lista di identificativi nella RREQ

⟶ Rappresenta la trasmissione della RREQ

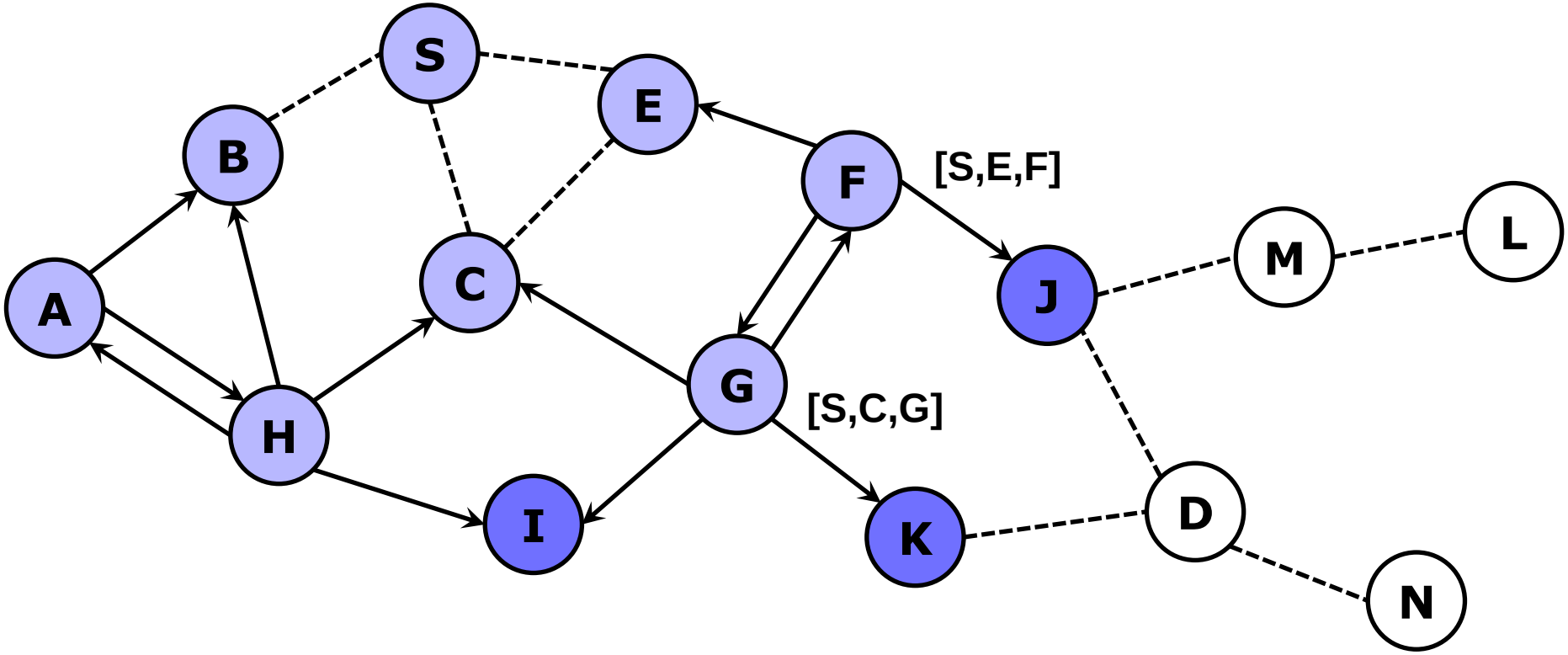
Route Discovery in DSR



Il nodo H riceve la RREQ da due nodi (C e B)

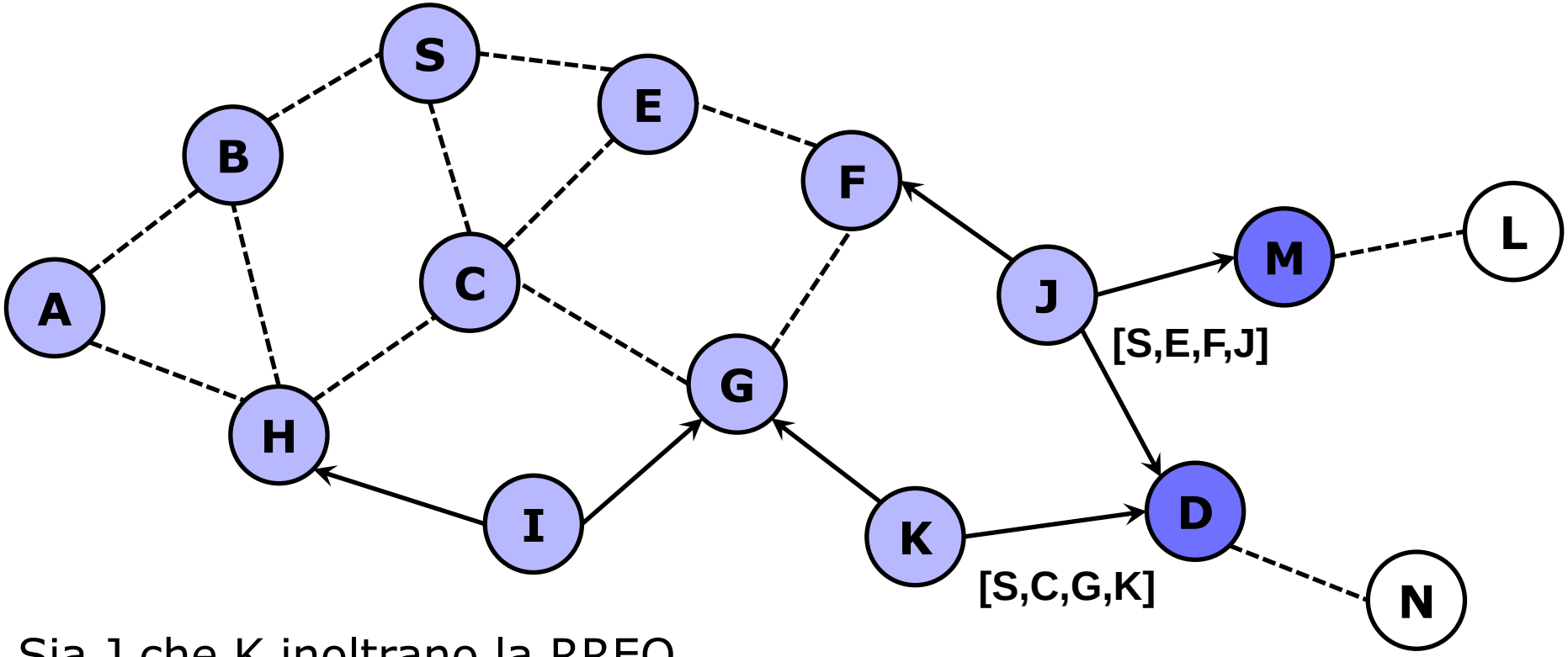
- Possibile ***collisione***

Route Discovery in DSR



Il nodo C riceve la RREQ dai due nodi G e H, ma non la ritrasmette poiché l'ha già inoltrato

Route Discovery in DSR

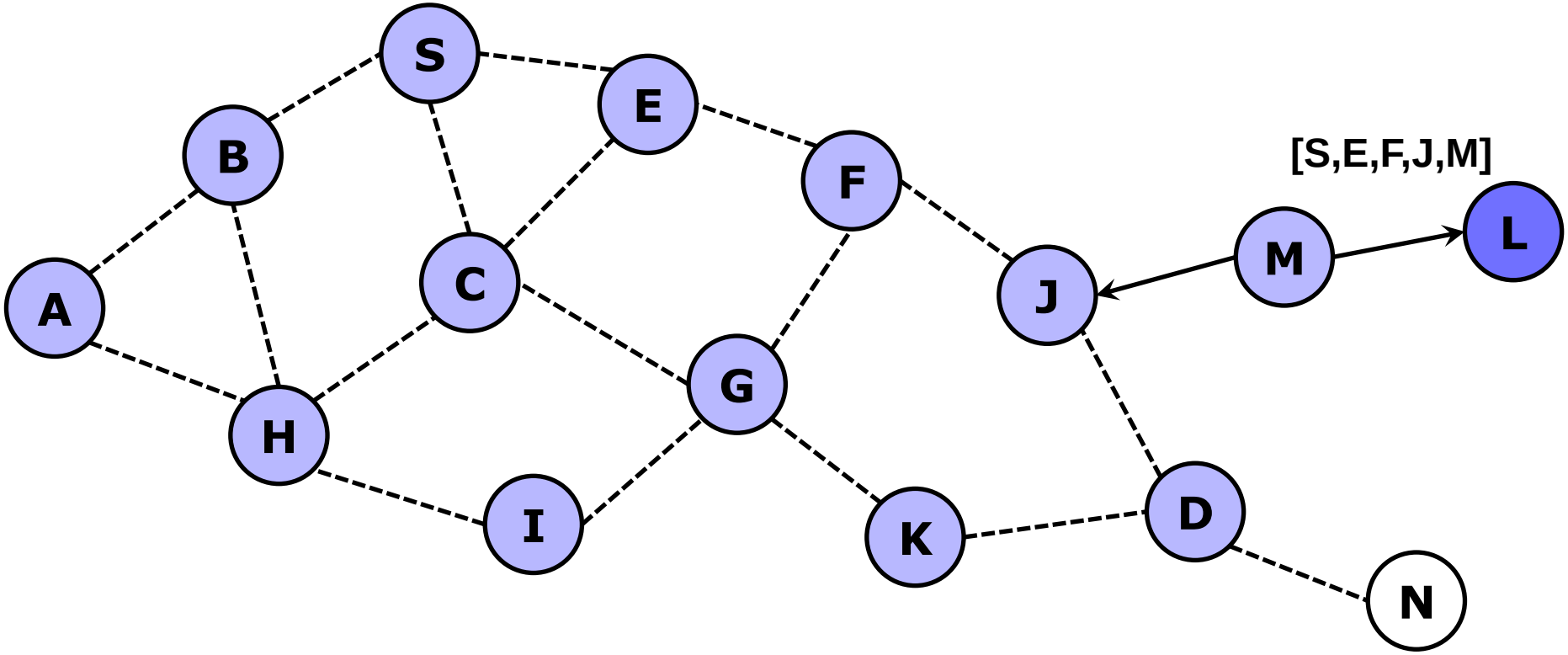


Sia J che K inoltrano la RREQ

Poiché i nodi J e K sono nascosti, la loro trasmissione può collidere

- La RREQ potrebbe **non essere consegnata** a D, nonostante il flooding

Route Discovery in DSR

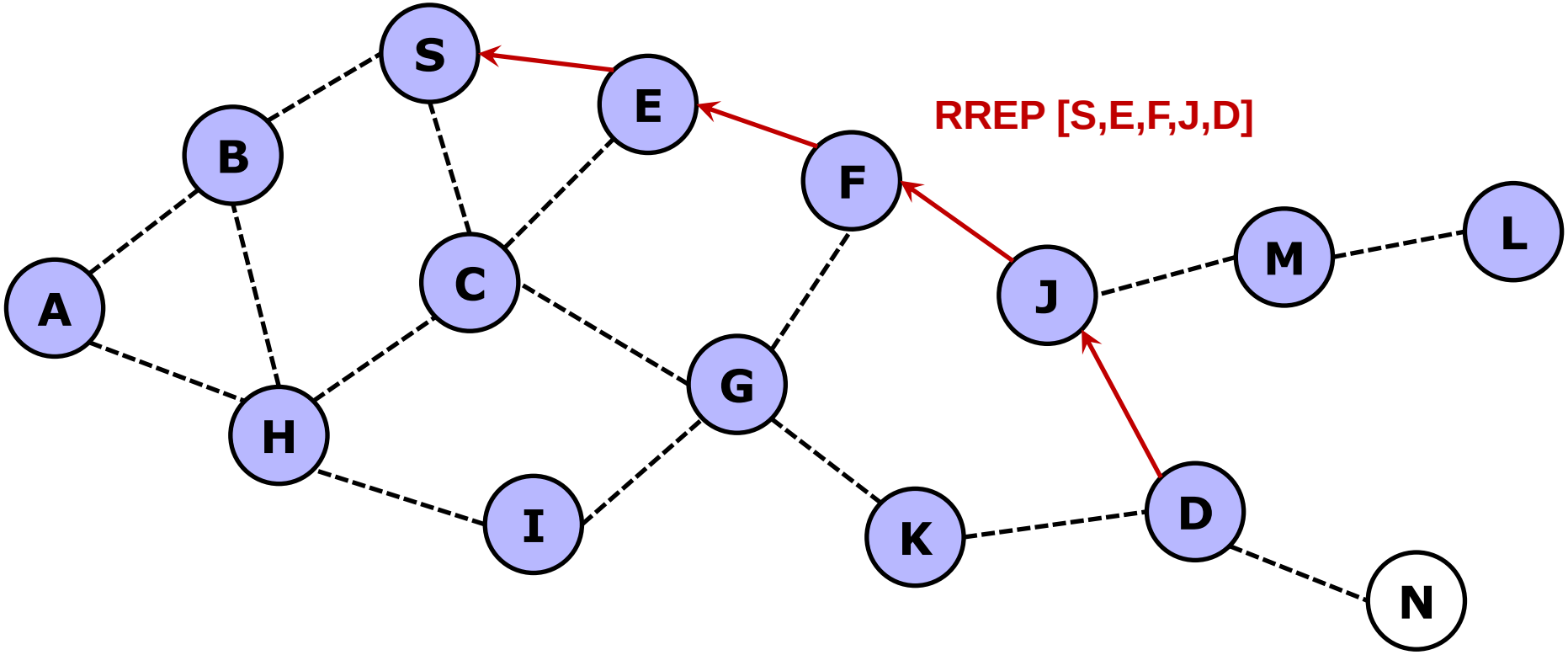


Il nodo D non ritrasmette la RREQ, poiché è la destinazione

Route Discovery in DSR

- Quando la destinazione D riceve la prima RREQ, invia una **Route Reply** (**RREP**) al nodo S
- La RREP è inviata al nodo S utilizzando il percorso ottenuto invertendo quello contenuto nella RREQ ricevuta
- La RREP contiene il percorso da S a D dal quale è stata ricevuta la prima RREQ

Route Reply in DSR



La RREP è trasmessa sul percorso inverso ottenuto dalla prima RREQ ricevuta

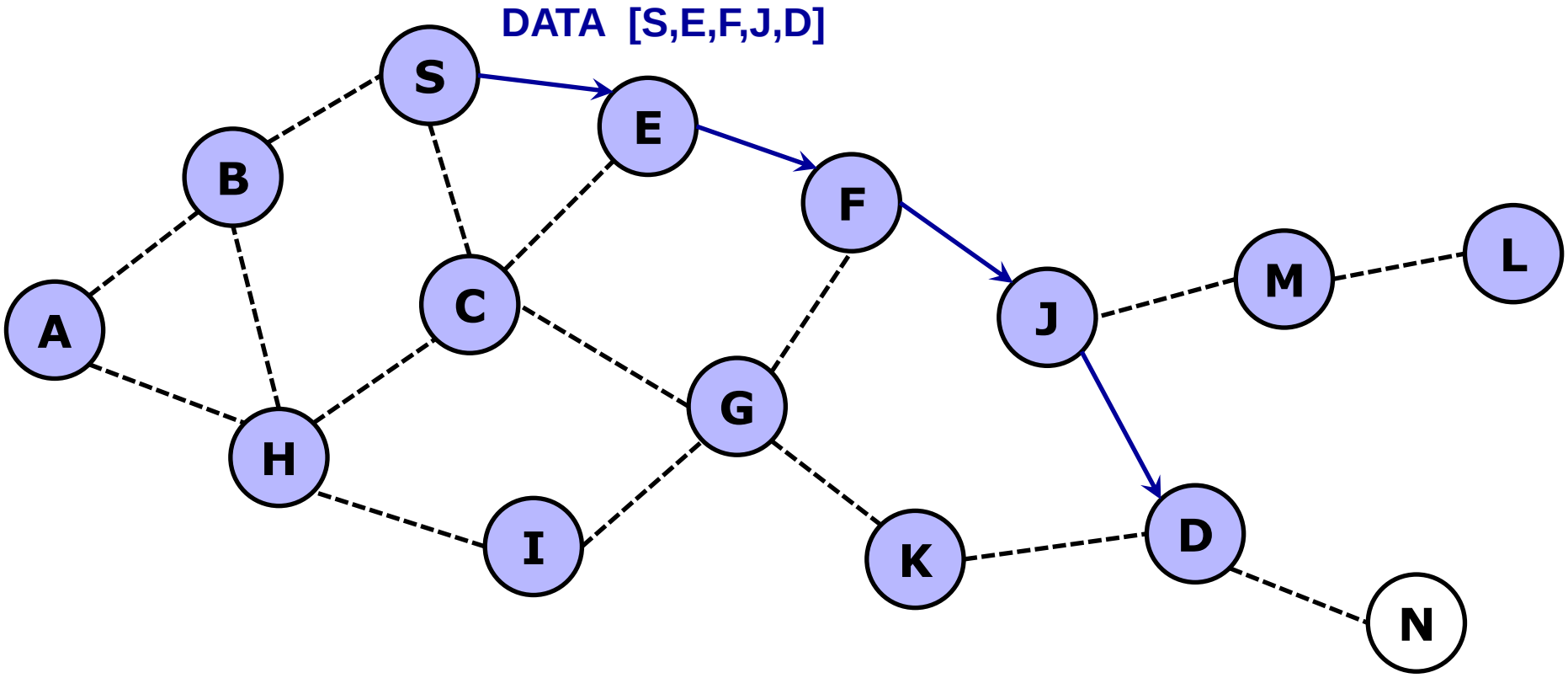
Route Reply in DSR

- Le Route Reply può essere inviata utilizzando il percorso inverso contenuto nella RREQ solo se i collegamenti wireless sono bi-direzionali
  - Per garantire tale proprietà le RREQ dovrebbero essere inoltrate solo se ricevute da un nodo con cui si è stabilito un collegamento bi-direzionale
- Se sono ammessi link unidirezionali, la RREP richiede una route discovery da D a S
  - Solo se il percorso da D a S non è noto
  - Se la route discovery è eseguita da D verso S, la RREP è inviata nella RREQ (piggybacked)
- Se si utilizza il MAC 802.11 i link devono essere bidiriezionali in quanto ogni trama unicast richiede un ACK

Dynamic Source Routing (DSR)

- Alla ricezione della RREP, il nodo S memorizza il percorso contenuto nella RREP
- Quando S invia un pacchetto dati verso D include il percorso nel header del pacchetto dati
  - Da qui il nome *Source Routing*
- I nodi intermedi utilizzano il percorso contenuto nel pacchetto dati per decidere a quale nodo inoltrarlo

Data Delivery in DSR



L'header del pacchetto dati contiene il percorso seguito dal pacchetto dati

- La dimensione del pacchetto cresce con la lunghezza del percorso

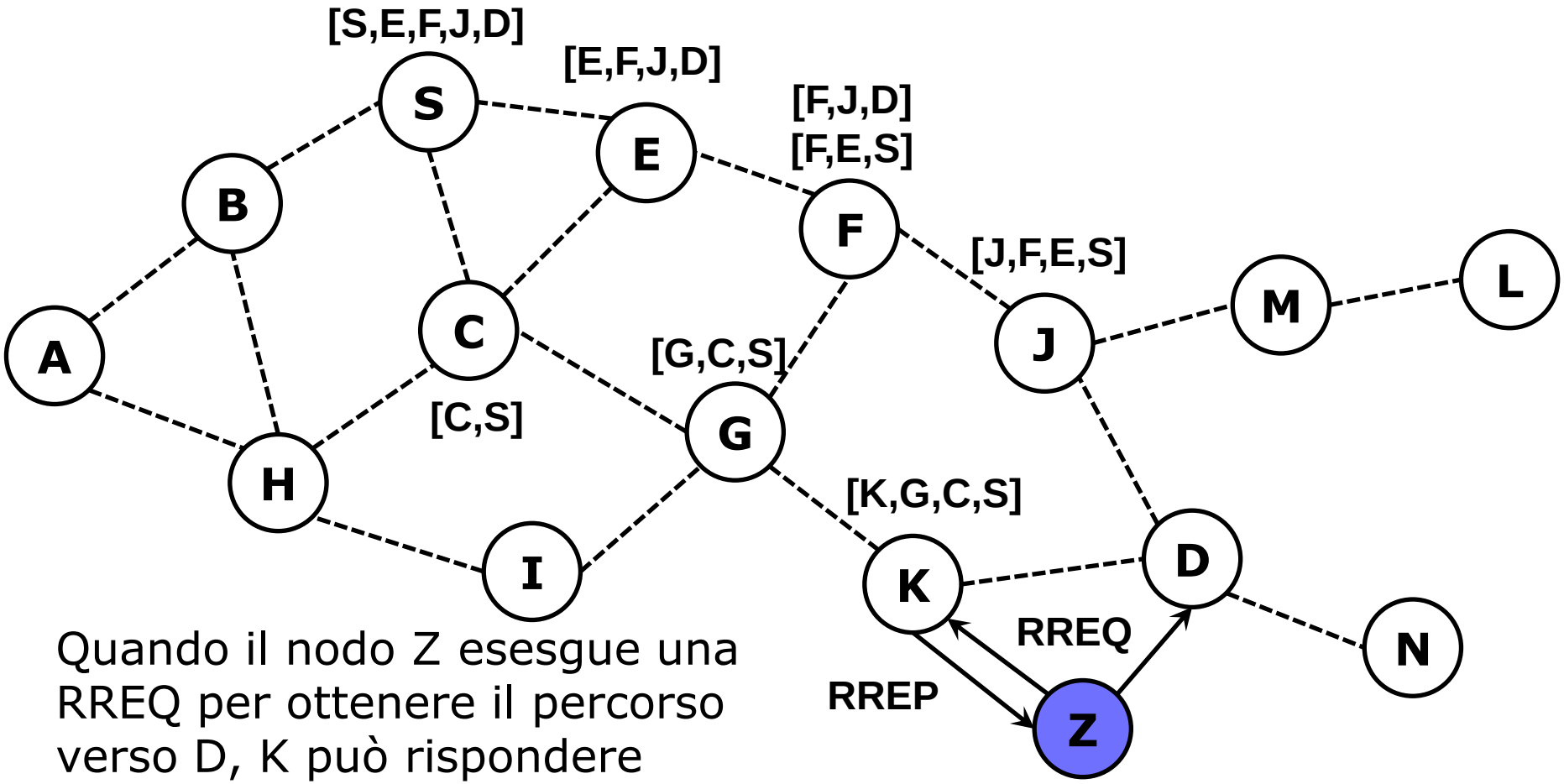
Ottimizzazioni DSR: Route Caching

- Ogni nodo memorizza un nuovo percorso appreso in un modo qualsiasi
- Quando il nodo S apprende il percorso per D [S,E,F,J,D], apprende anche il percorso per F [S,E,F]
- Quando il nodo K riceve la RREQ a lui destinata [S,C,G], K apprende il percorso alla destinazione S [K,G,C,S]
- Quando il nodo F inoltra la RREP [S,E,F,J,D], il nodo F apprende il percorso [F,J,D] verso D
- Quando il nodo E inoltra i dati contenenti il percorso [S,E,F,J,D], apprende il percorso [E,F,J,D] verso D
- Un nodo può apprendere un nuovo percorso anche dall'ascolto delle trasmissioni dati non dirette a lui

Utilizzo della Cache di Routing

- Quando il nodo S apprende che il percorso al nodo D e rotto, utilizza un altro precorso contenuto nella sua cache locale (se un tale percorso esiste). Se non esiste un'alternativa viene eseguita una nuova Route Discovery
- Un nodo X alla ricezione della RREQ per il nodo D può inviare subito la RREP se conosce un percorso verso D
- L'utilizzo della cache di routing
  - Può accelerare la Route Discovery
  - Può ridurre l'overhead di segnalazione

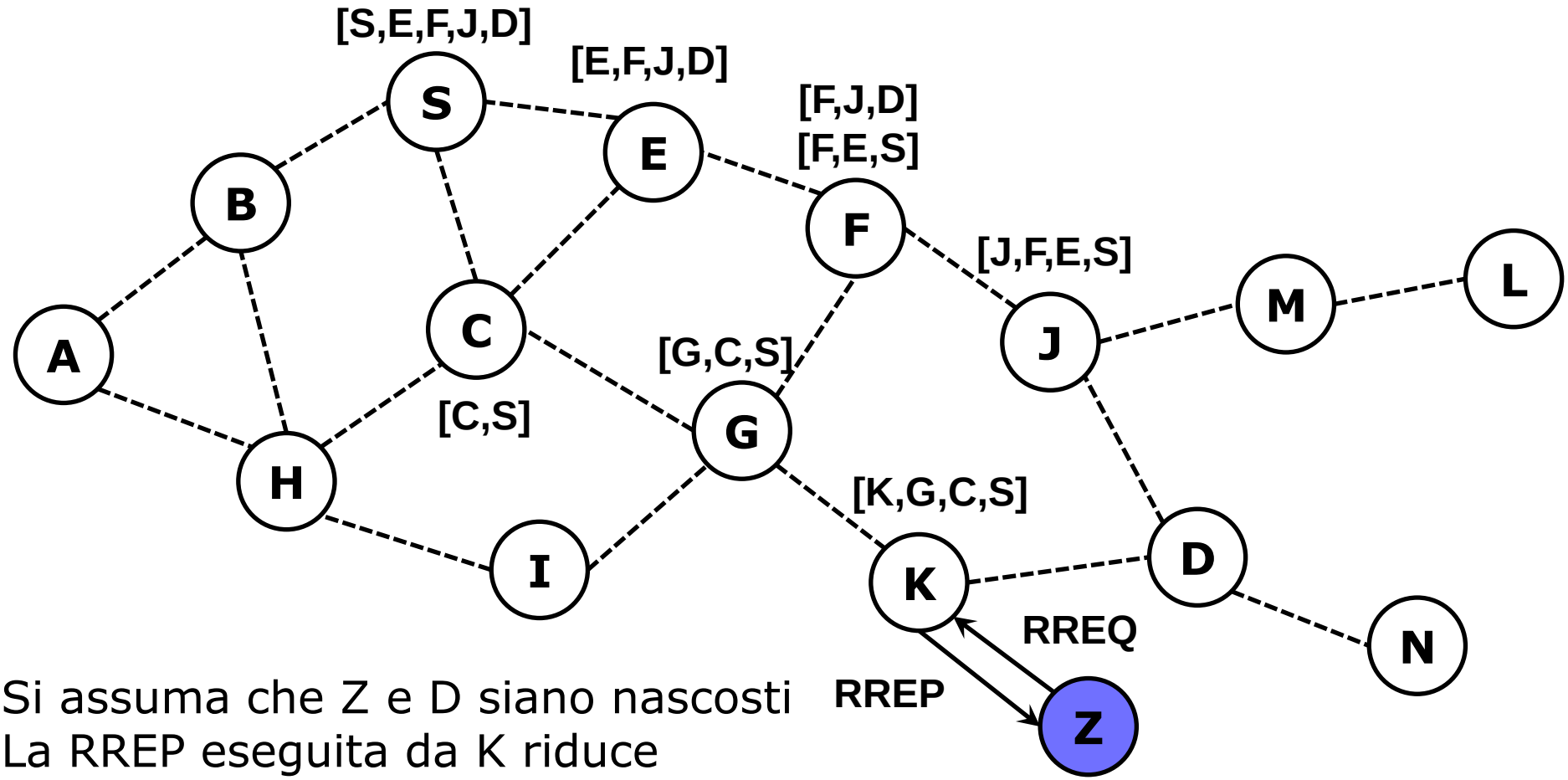
Route Caching in DSR



Quando il nodo Z esesgue una RREQ per ottenere il percorso verso D, K può rispondere immediatamente con una RREP [Z,K,G,C,S]

➡ Route Discovery **più veloce**!

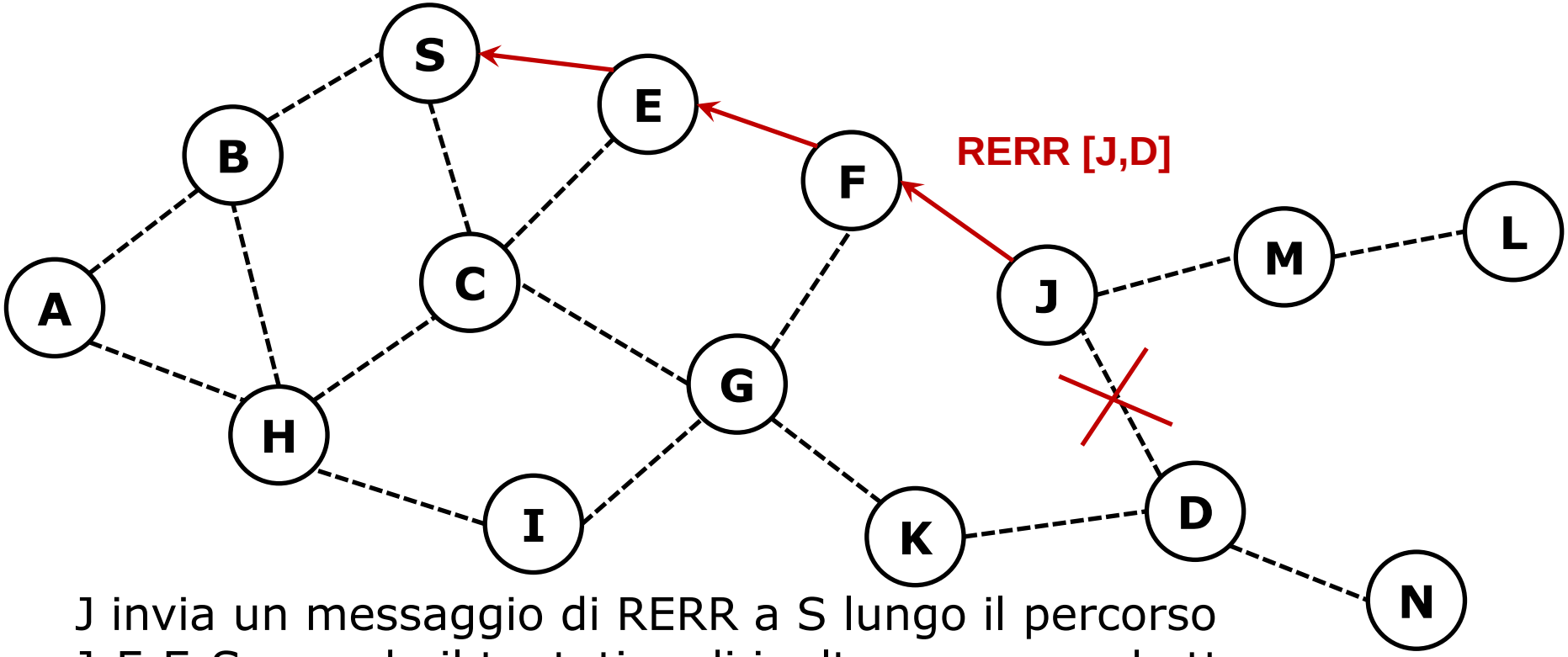
Route Caching in DSR



Si assuma che Z e D siano nascosti
La RREP eseguita da K riduce l'overhead di segnalazione (la RREQ non viene propagata)

➡ Route Discovery **più eficiente**!

Route Error in DSR



J invia un messaggio di RERR a S lungo il percorso J-F-E-S quando il tentativo di inoltrare un pacchetto dati di S sul link (J;D) fallisce

I nodi che ricevono la RERR aggiornano la propria route cache per rimuovere il link (J;D)

Route Caching: Attenzione!

- Se la cache contiene informazioni non aggiornate (vecchie) può influire negativamente sulle prestazioni
- Con il passare del tempo le informazioni contenute nella cache possono diventare non più valide
- A nodo sorgente può dover provare diversi percorsi non validi prima di trovare una rotta valida verso la destinazione

DSR: Vantaggi

- I percorsi sono memorizzati e aggiornati solo tra nodi che desiderano comunicare
  - Basso overhead di segnalazione
- Route caching può migliorare ulteriormente le prestazioni
- Una singola Route Discovery permette di scoprire percorsi verso destinazioni multiple grazie ai nodi intermedi che utilizzano il meccanismo della Route Caching

DSR: Svantaggi

- La dimensione dell'header cresce con la lunghezza del percorso
- Il flooding delle RREQs può potenzialmente raggiungere tutti i nodi della rete
- E' necessario porre attenzione al meccanismo di propagazione delle RREQs, poiché la trasmissione contemporanea di più RREQs può causare una collisione
  - Ritardi casuali prima della ritrasmissione delle RREQs

DSR: Svantaggi

- La contesa del canale aumenta se troppi nodi intermedi rispondono con le loro informazioni locali
  - Route Reply Storm
  - Soluzione: se un nodo sente la trasmissione di una RREPs per la stessa coppia S-D non trasmette la propria RREP
- Un nodo intermedio potrebbe trasmettere una RREP con informazioni non aggiornate, causando la propagazione dell'errore nelle route cache degli altri nodi
  - Soluzione: introduzione di un meccanismo per l'eliminazione di percorsi (potenzialmente) non validi

Ad-Hoc On-demand Distance Vector (AODV)

- DSR include il percorso nell'header del pacchetto
  - Degradazione delle prestazioni possono essere causate dalla dimensione dell'header
- AODV migliora le prestazioni di DSR utilizzando tabelle di routing ad ogni nodo rimuovendo la necessita di inserire il percorso nell'header del pacchetto
- AODV mantiene le caratteristiche di DSR
  - I percorsi sono mantenuti solo dai nodi che desiderano comunicare

AODV

- Route Requests (RREQs) sono trasmesse in modo simile a DSR
- Quando un nodo ritrasmette una RREQ, crea un percorso inverso che punta verso la sorgente S della RREQ
  - AODV assume link simmetrici (comunicazione bi-direzionale)
- Quando la destinazione D riceve la RREQ risponde inviando una Route Reply (RREP)
- La RREP raggiunge la sorgente S utilizzando il percorso inverso creato durante l'inoltro della precedente RREQ

Route Reply in AODV

- Un nodo intermedio (diverso dalla destinazione) può inviare la RREP se conosce un percorso più recente di quello precedentemente conosciuto da S
- Per determinare se il percorso conosciuto dal nodo intermedio è **più recente** di quello di S si utilizzano **destination sequence numbers**
- La probabilità che un nodo intermedio invii una RREEP con AODV è più bassa di DSR
  - A una nuova RREQ inviata da S per una destinazione D è assegnato un nuovo dest. seqno. Un nodo intermedio che conosce un percorso verso D ma con un dest seqno più basso non può inviare la RREP

AODV Timeouts

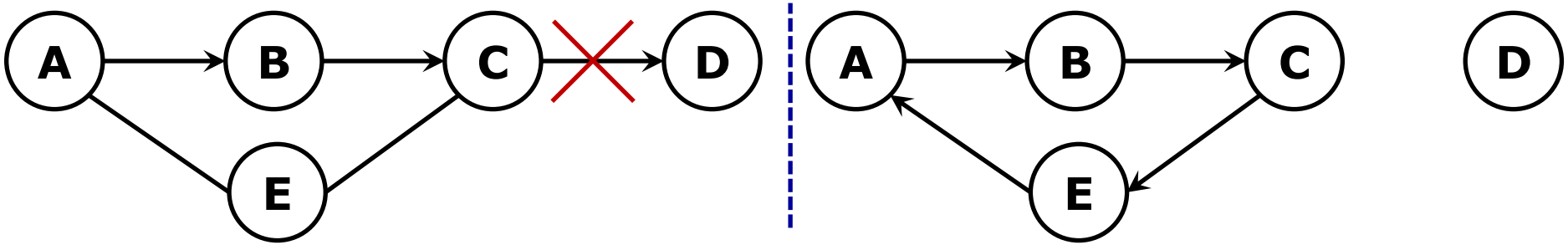
- Una entry della tabella di routing contenente un **percorso inverso** è rimossa quando un timeout scade
  - Il timeout dovrebbe essere sufficientemente lungo da permettere alla RREP di tornare indietro
- Una entry della tabella di routing contenente un **percorso diretto** è rimossa se **non utilizzata** per un periodo pari a **active_route_interval**
  - Se una entry della tabella di routing non è utilizzata per inviare dati verrà rimossa (anche se è ancora valida)

Link Failure Reporting

- Un vicino di un nodo X è considerato attivo per una entry della tabella di routing, se il vicino (che è stato utilizzato per inoltrare il pacchetto) invia un pacchetto entro **active_route_timeout**
- Quando la entry corrispondente a un nodo utilizzato come next hop è rimossa, tutti i vicini vengono informati
- Link failures sono propagate nella rete per mezzo di messaggi di Route Error (RERR) che aggiornano i destination sequence numbers

Perché Sequence Numbers?

- Per evitare l'utilizzo di vecchi link non più valdi (o esistenti)
  - Per determinare quale messaggio contiene il percorso più recente
- Per evitare la formazione di cicli



  - Si assumi che A non conosca che il link (C;D) è rotto poiché la RERR inviata da C è stata persa
  - C esegue una Route Discovery per D. Il nodo A riceve la RREQ (ad es. attraverso C-E-A)
  - A risponde poiché conosce un percorso per D attraverso B
  - Si ottiene il ciclo C-E-A-B-C

Route Error in AODV

- Quando un nodo X non può inoltrare il pacchetto P (da S a D) sul link (X;Y) genera un messaggio di Route Error
- Il nodo X incrementa il dest seqno per D memorizzato nella sua cache
- Il dest seqno N aggiornato è incluso nella RERR
- Quando il nodo S riceve la RERR inizia una nuova Route Discovery per D utilizzando come dest seqno un valore almeno pari a quello contenuto nella RERR (pari a N)

Destination Sequence Numbers

… Continua dalla precedente slide

- Quando il nodo D riceve la RREQ con dest seqno pari a N, il nodo memorizza N come proprio dest seqno se non ha già ricevuto un valore più elevato

Link Failure Detection

- Messaggi di Hello: i nodi periodicamente trasmettono un messaggio di Hello (keep alive)
- L'assenza del messaggio di Hello è utilizzata come indicazione di Link Failure
- Alternativamente, l'impossibilità di ricevere diversi riscontri (ACK) a livello MAC può essere utilizzato come indicazione di Link Failure

Ottimizzaioni AODV: Expanding Ring Search

- Le RREQ sono inizialmente trasmesse con un basso valore del Time-To-Live (TTL), per limitare la loro propagazione
  - DSR implementa un'ottimizzazione simile
- Se non è ricevuta alcuna RREP si trasmette una nuova RREQ con un valore del TTL maggiore

AODV: Conclusioni

- I percorsi non devono essere inseriti nei pacchetti dati
- I nodi mantengono una tabella di routing contenente le sole entries che rappresentano percorsi attivi
- Per ogni destinazioni viene mantenuto un solo nodo next hop
  - Estensioni per routing multi-paths possono essere previste
  - DSR può mantenere più rotte per ogni singola destinazione
- I percorsi non utilizzati vengono rimossi anche se la topologia non è cambiata

Università degli Studi di Bergamo
Dipartimento di Ingegneria dell'Informazione e Metodi Matematici

Proactive Protocols

Optimized Link State Routing (OLSR)

Link State Routing

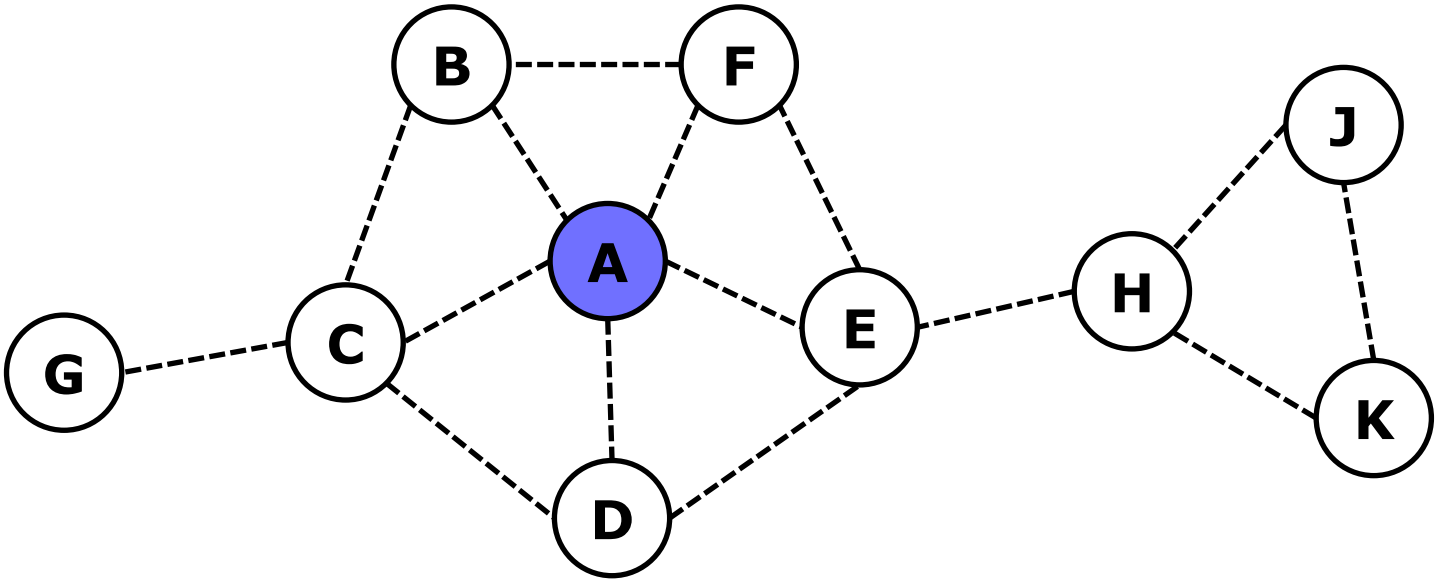
- Ogni nodo periodicamente trasmette informazioni contenenti lo stato dei link ad esso adiacenti
- Ogni nodo ritrasmette le informazioni ricevute dai nodi vicini attraverso il meccanismo del flooding
- Ogni nodo memorizza le informazioni sullo stato dei link ricevute dagli altri nodi
- Ogni nodo utilizza le informazioni ricevute per calcolare il next hop per ogni destinazione
  - Si calcola la topologia di rete (grafo di rete)
  - Si calcolano i percorsi ottimi utilizzando l'algoritmo di Dijkstra

Optimized Link State Routing (OLSR)

- L'overhead di rete causato dal flooding delle informazioni sullo stato dei link è ridotto utilizzando un sottoinsieme dei nodi vicini per la ritrasmissione
- Una trasmissione broadcast del nodo X è ritrasmessa solo dai suoi Multi-Point Relays (MPRs)
- I MPRs di un nodo X sono quei vicini di X (nodi a 1 hop) tali per cui ogni nodo a 2 hop da X è un vicino di uno degli MPRs (a 1 hop)
  - Ogni nodo trasmette periodicamente la lista dei suoi vicini, cosicché ogni nodo X può individuare i nodi che sono a 2 hop e scegliere i MPRs

Optimized Link State Routing (OLSR)

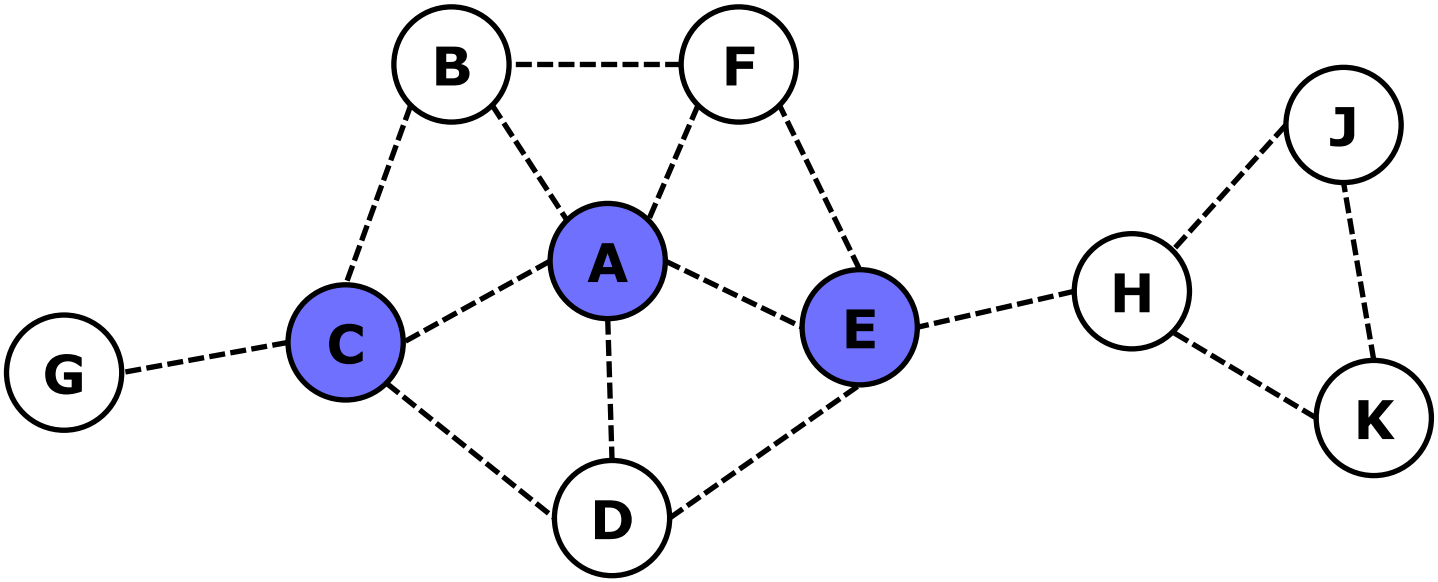
- I nodi C e E sono Multipoint Relays del nodo A



Rappresenta il nodo che trasmette le informazioni di stato dei link adiacenti di A

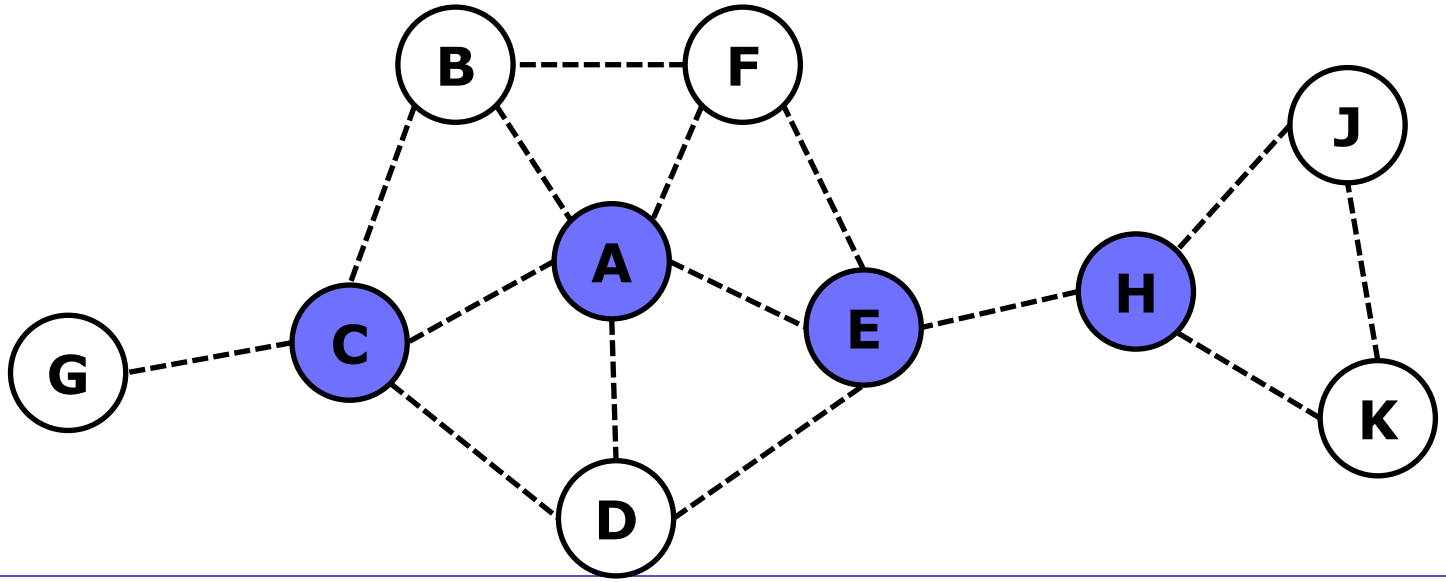
Optimized Link State Routing (OLSR)

- I nodi C e E ritrasmettono le informazioni di stato dei link ricevute da A



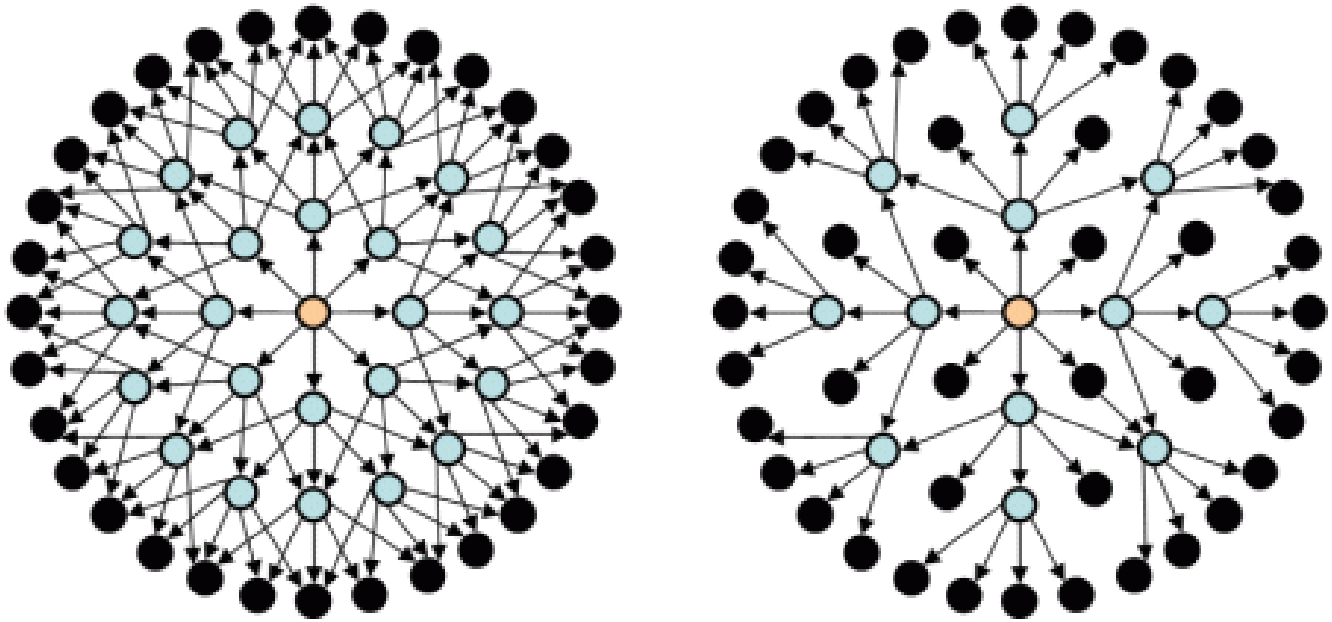
Optimized Link State Routing (OLSR)

- I nodi E e K sono Multipoint Relays del nodo H
- Il nodo K inoltra le informazioni ricevute da H
  - E ha già inoltrato le informazioni ricevute



OLSR

- OLSR esegue il flooding delle informazioni di routing attraverso il meccanismo degli MPRs
  - Riduzione del numero di trasmissioni



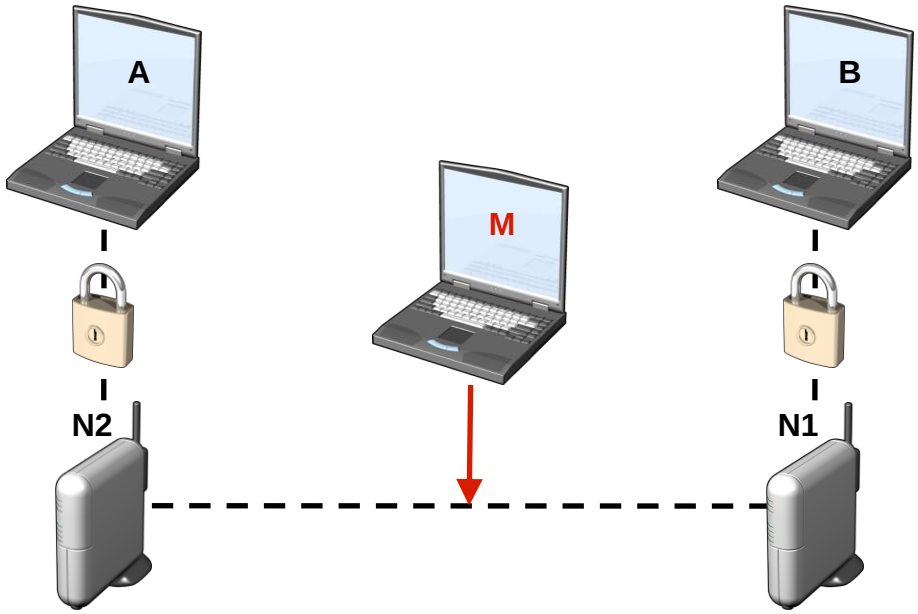
Security in WMNs

Motivations

- Security in WMNs is still in its infancy
- The design of a security scheme represents a challenging research topic
  - Multi hop communication model
  - Low cost of the devices used to deploy a WMN
- Several attacks have been analyzed and applied against WMNs
- The current schemes deal with security weaknesses related to a specific layer or protocol

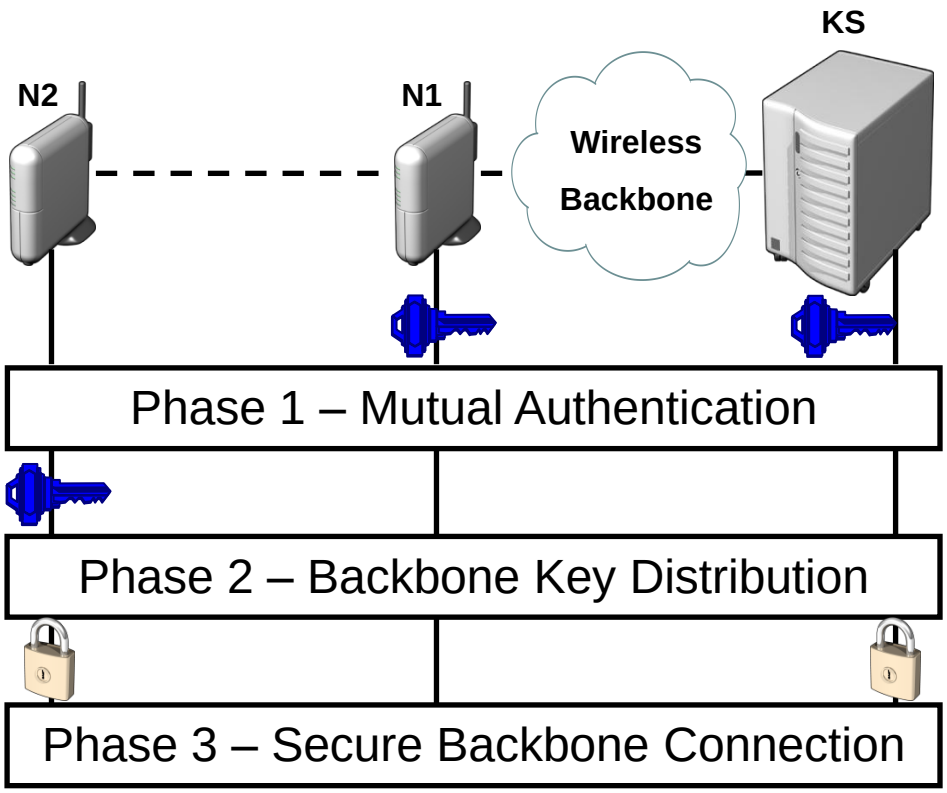
Access Network Security

- Authentication and encryption compliant with the IEEE 802.11i standard
  - High level of security
  - Compatibility with the current standard
- Unsuitable to protect the information exchanged between Mesh Clients through the wireless backbone



Backbone Network Security

- Three phases are carried out by a new node (N2):
  - N2 establishes a secure channel with a mesh router (node N1)
  - N2 obtains the information that will be used to generate the current backbone key
  - The device can connect to the wireless backbone
- Two protocols minimize the risk of using the same key



Security Architectures

- Two security architectures representing the two main families
  - MobiSEC ➔ Centralized Architecture
  - DSA-Mesh ➔ Distributed Architecture
- Security Assumptions:
  - There exists a CA recognized by all nodes
  - All network entities have a pair of public/private keys
  - All network entities have a certificate signed by the CA
  - Loose synchronization among devices

MobiSEC: A Centralized Security Architecture

- Key currently used and its validity time:

$$r = \left\lfloor \frac{t_n - t_S}{T} \right\rfloor + 1$$

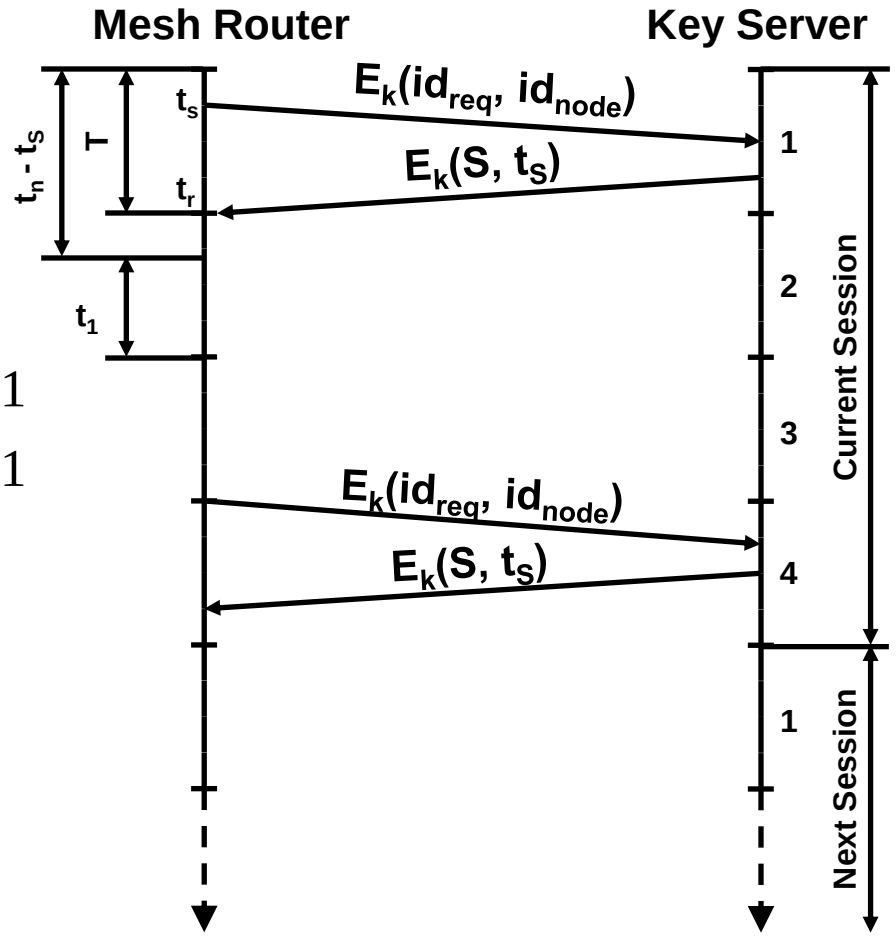
$$t_1 = rT - (t_n - t_S)$$

$$\begin{cases} key(r,S) = h(t_S \mid S) & r = 1 \\ key(r,S) = h(t_S \mid K \mid key(r-1,S)) & r > 1 \end{cases}$$

- Correction factor:

$$\begin{cases} c = \left\lceil \frac{\Delta t - T}{T} \right\rceil & \text{if } \Delta t \geq T \\ c = 0 & \text{if } \Delta t < T \end{cases}$$

$$\Delta t = t_r - t_S$$

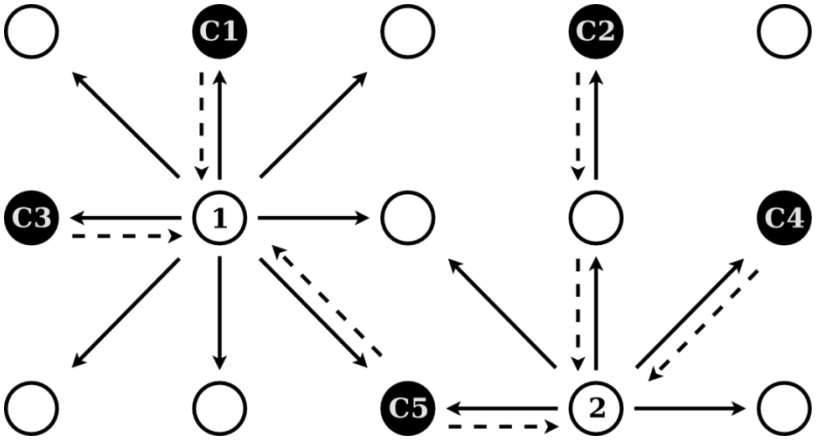


DSA-Mesh: A Distributed Security Architecture

- DSA-Mesh: distributed system based on threshold cryptography to overcome the single point of failure
  - Signatures can be verified using **t** out of **n** partial signatures
  - If **n = 2t – 1** and **t** nodes are tamper resistant the *key service* is secure
- The distribution of the *key service* requires:
  - A protocol to generate the session secret (Session Secret Agreement Protocol)
  - A protocol to deliver the session secret (Distribute Proactive Request)

DSA-Mesh: A Distributed Security Architecture

- Two sets of mesh routers:
  - **Core Nodes**: collaboratively generate the new session secret and provide it to the generic nodes
  - **Generic Nodes**: require the successive session secret to a subset of core nodes
- Increased reliability of the overall security architecture



(5,3)-threshold scheme

DSA-Mesh Availability

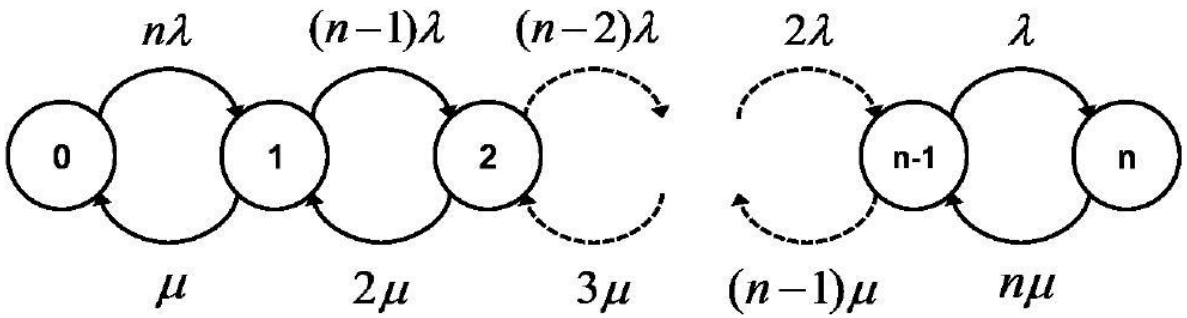
- A core node has only two states
  - it is either available or unavailable
- Different network nodes fail independently
- The inter-failure time and the repair time are independent with average
  - Mean Time To Failure (MTTF)
  - Mean Time To Repair (MTTR)
- Identical failure and recovery rates, equal to λ and μ, respectively

DSA-Mesh Availability

- Steady-state availability A of a single core node

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1/\lambda}{1/\lambda + 1/\rho} = \frac{1}{1+\rho}$$

- The system can be represented using the following Markov chain
  - Each node represents the number of unavailable core nodes



DSA-Mesh Availability

- Probability that at most (t−1) core nodes are down

$$P(U \le t-1) = \sum_{i=0}^{t-1} \pi(i)$$

$$\pi(i) = \binom{n}{i} (1-A)^i A^{n-i} = \binom{n}{i} \frac{\rho^i}{(1+\rho)^n}$$

Average down-time [h/y] (MTTR = 24 h)

MTTF (d)	1	(3,2)	(5,3)	(7,4)
30	283	27	3	0.3
60	143	7	0.34	0.02
90	96	3.2	0.11	0.004

DSA-Mesh: A Distributed Security Architecture

- Core Nodes

$$M = \left(S^e, t_s^e, t_g^e, n_g^e\right)$$

$$\forall c \in C \rightarrow \left[h(M)\right]^{K_c} \bmod N$$

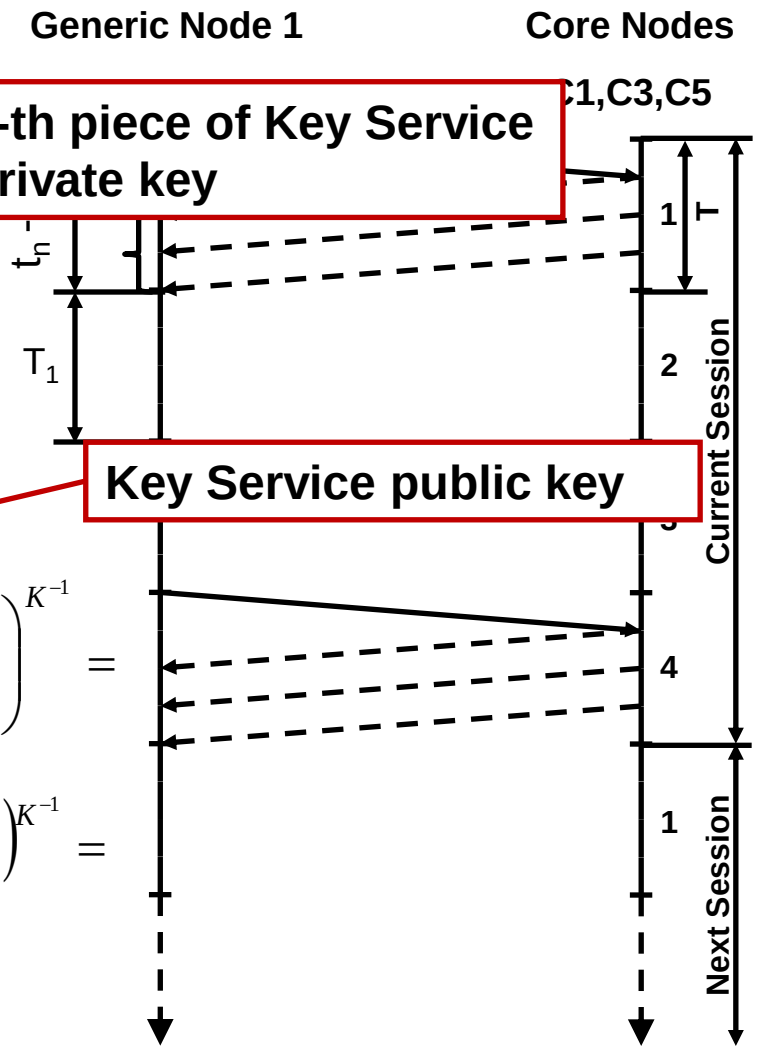
- Generic Nodes

$$\forall g \in G \rightarrow \prod_{c=1}^{t} \left[h(M)\right]^{K_c} \bmod N$$

$$\left(\prod_{c=1}^{t} \left[h(M)\right]^{K_c} \bmod N\right)^{K^{-1}} = \left(\left[h(M)\right]^{\sum_{c=1}^{t} K_c} \bmod N\right)^{K^{-1}} =$$

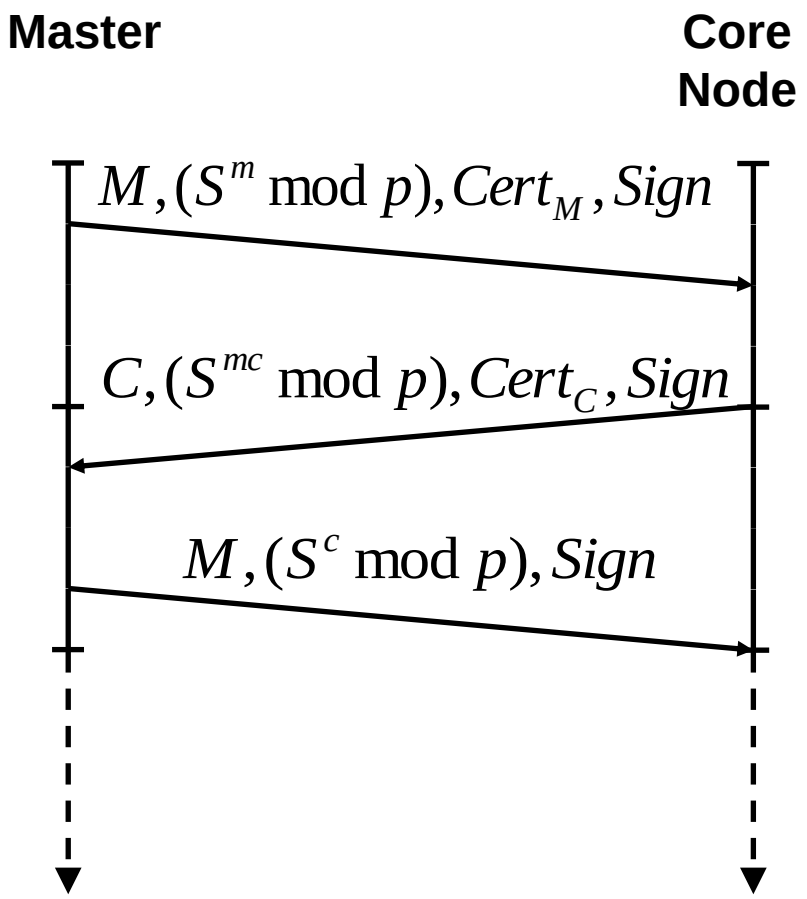
$$= \left(\left[h(M)\right]^{\sum_{c=1}^{t} l_c(0) s_c} \bmod N\right)^{K^{-1}} = \left(\left[h(M)\right]^{K} \bmod N\right)^{K^{-1}} =$$

$$= \left[h(M)\right]^{K \cdot K^{-1}} \bmod N = h(M)$$

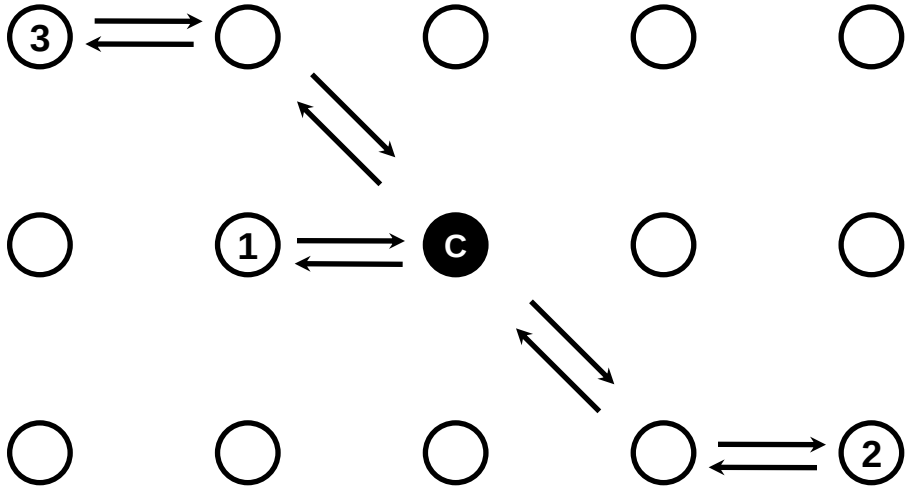


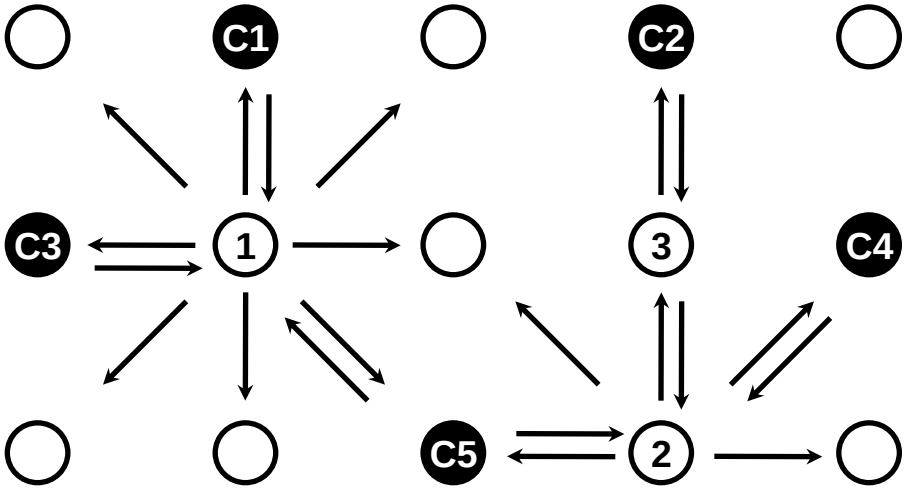
DSA-Mesh: A Distributed Security Architecture

- Three pass protocol
  - A master node trigger the protocol
  - The last message discloses the secret S
  - The last message is used to select the successive master
- Each message
  - is signed with the private key
  - contains the node certificate



Centralized vs. Distributed Security Architectures

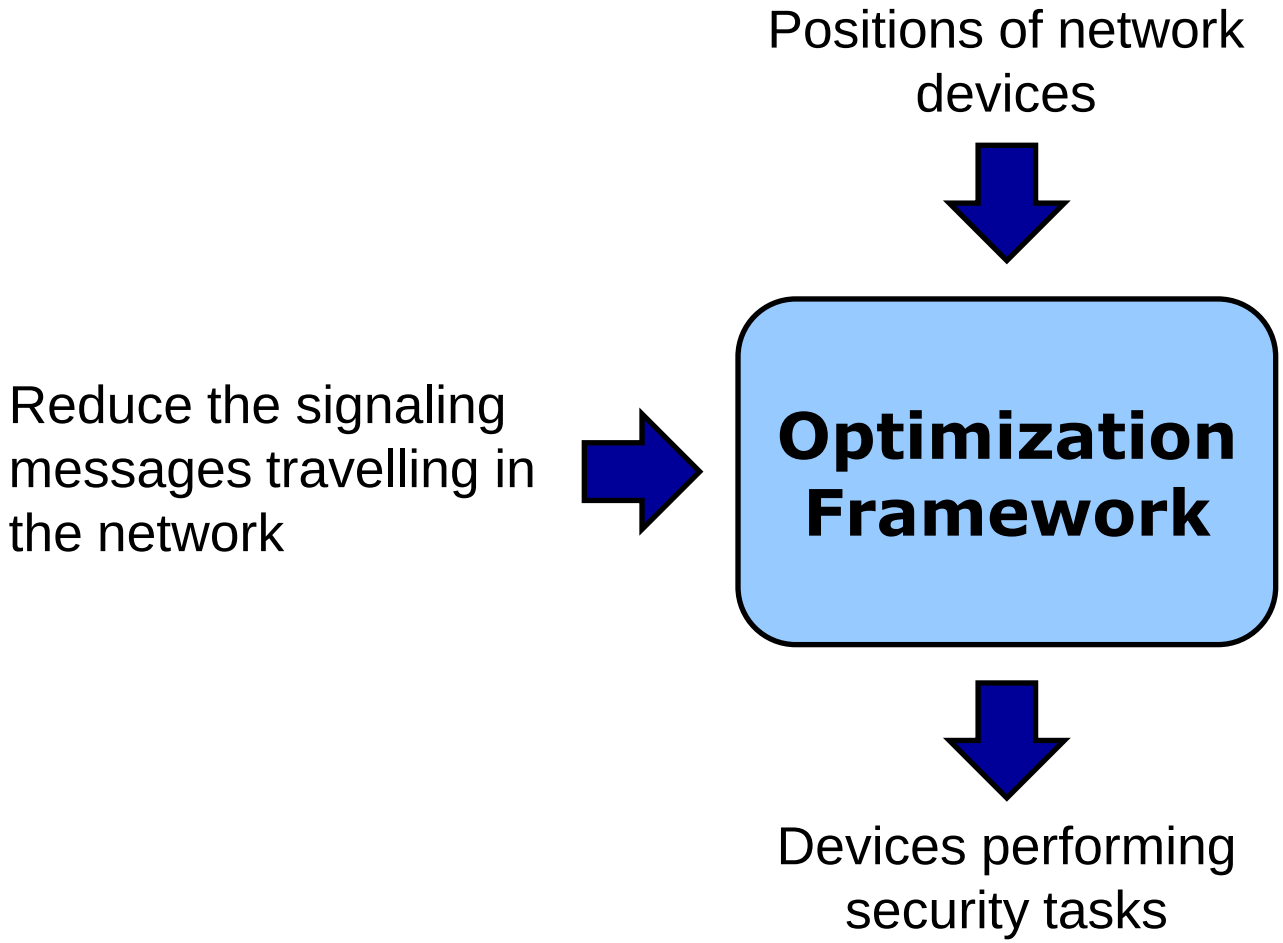




- Centralized security architectures
  - High responsiveness
  - Low reliability/robustness
  - Congestion

- Distributed security architectures
  - High communication overhead
  - High reliability/robustness
  - Load balancing

Centralized vs. Distributed Security Architectures



Optimal Planning of Distributed Security Architectures

- Object: Select the nodes that minimize the proactive request delay
- Placement formulated as an optimization problem
- Distance function (cost)
  - Proportional to the transmission delay
  - Hop count
- The problem is NP-hard (t-neighbor k-center problem)

Optimal Planning of Distributed Security Architectures

- Decision Variables:

$$y_i = \begin{cases} 1 & \text{if generic mesh router } i \text{ is selected as core node} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if generic mesh router } j \text{ is assigned to core node } i \\ 0 & \text{otherwise} \end{cases}$$

- ILP Model:

$$\min(u)$$

$$s.t. \sum_{i \in N} x_{ij} = t \qquad \forall j \in N$$

Assignment generic-core

$$\sum_{i \in N} y_i = n$$

Max number of core nodes

$$x_{ij} \leq y_i \qquad \forall i, j \in N$$

$$u \geq x_{ij} d_{ij} \qquad \forall i, j \in N$$

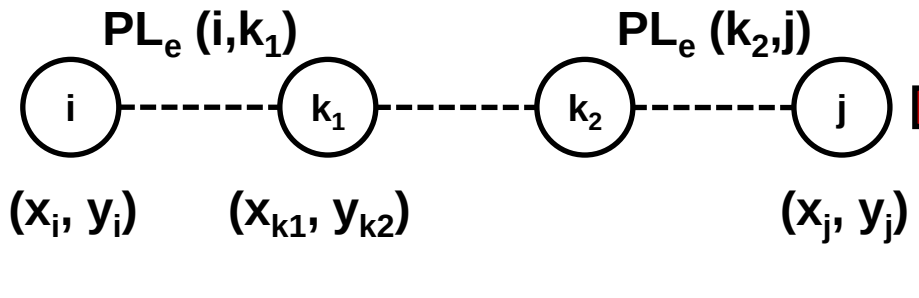
Min the max distance to the t closest core nodes

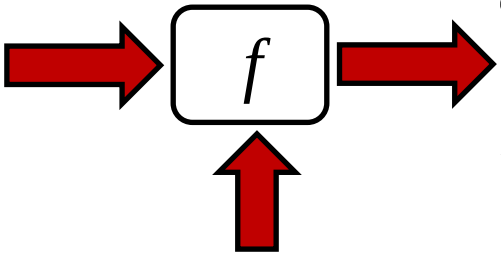
$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N$$

$$y_i \in \{0, 1\} \qquad \forall i \in N$$

Transmission Delay

- Link cost proportional to the transmission delay





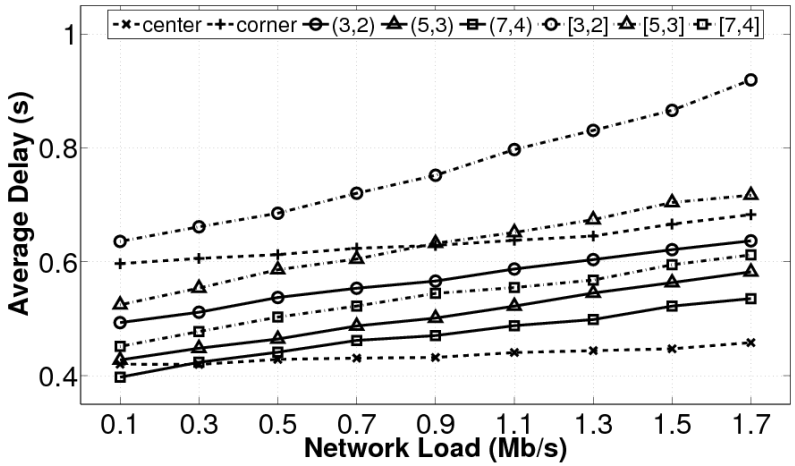
$$d_{ij} = \sum_{e \in P_{ij}} t_e = L \cdot \sum_{e \in P_{ij}} \frac{1}{r_e}$$

$$r_e = f\left(PL_{e,dB}\right)$$

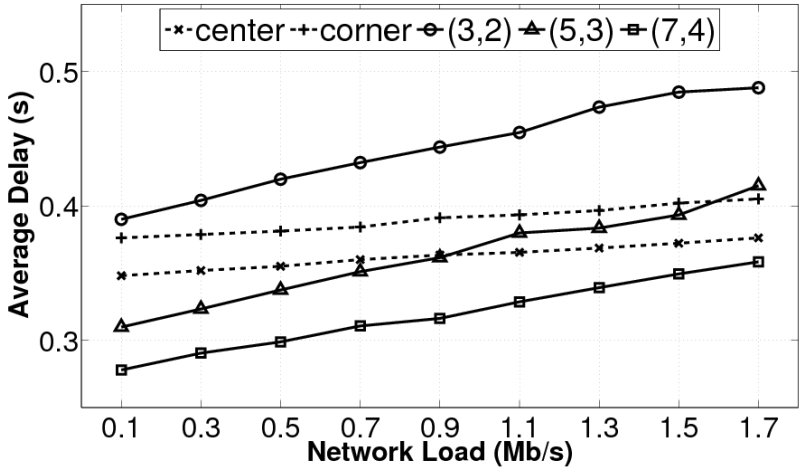
Path Loss (dB)	Data Rate (Mb/s)
88	6
87	9
85	12
83	18
80	24
75	36
73	48
71	54

Average Delay

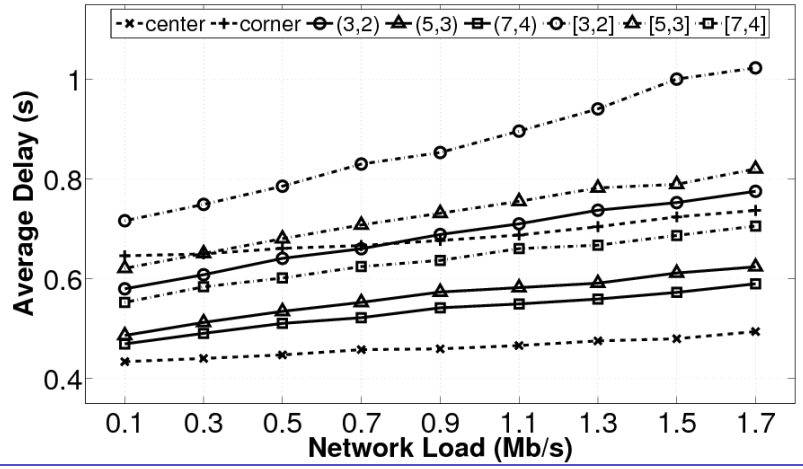
Grid topology (35 nodes)



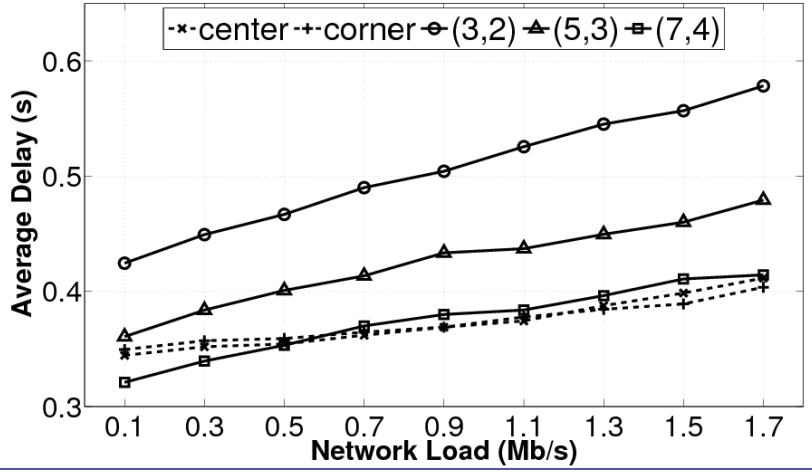
Random topology (35 nodes)



Grid topology (40 nodes)

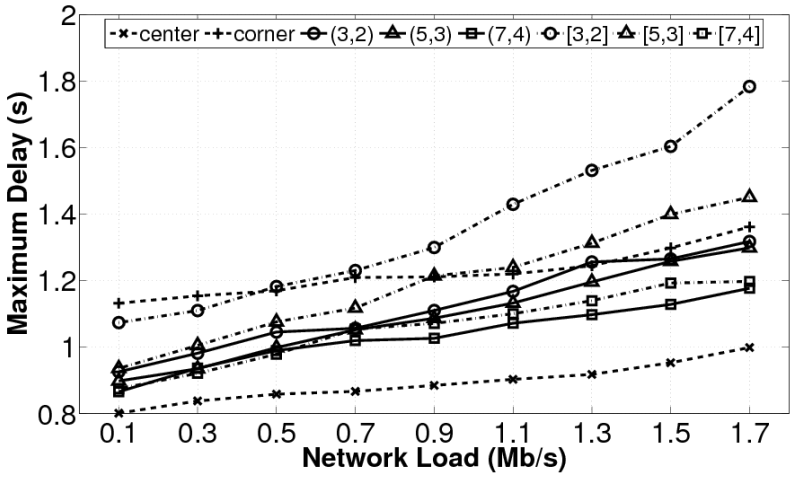


Random topology (40 nodes)

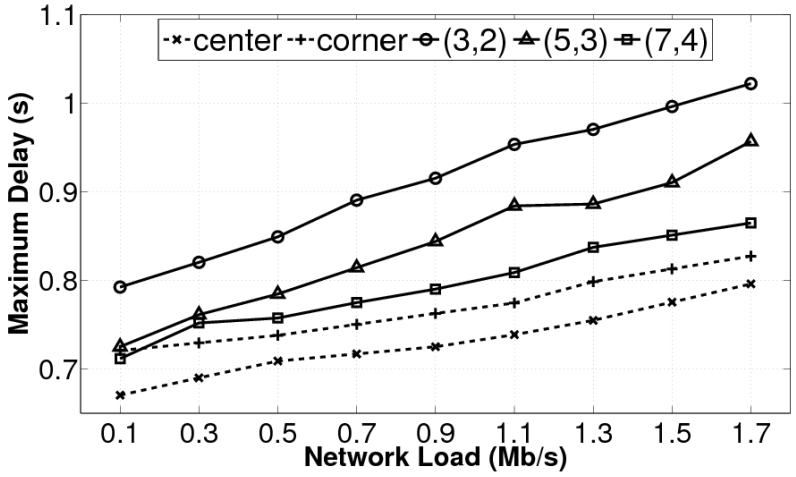


Maximum Delay

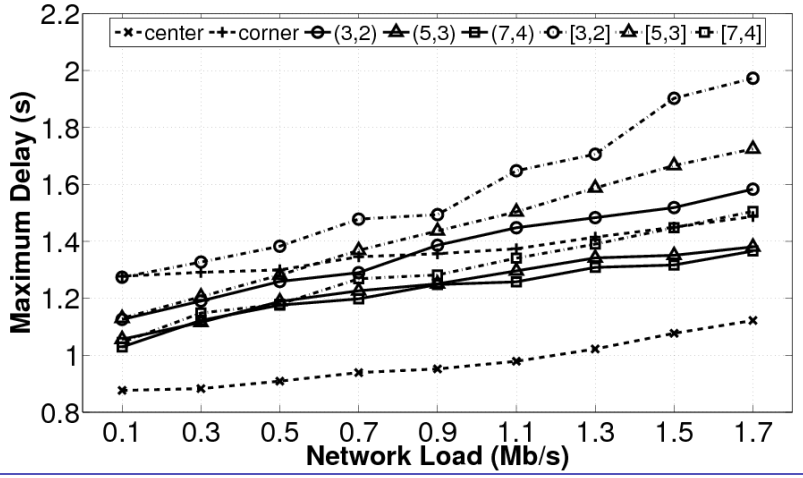
Grid topology (35 nodes)



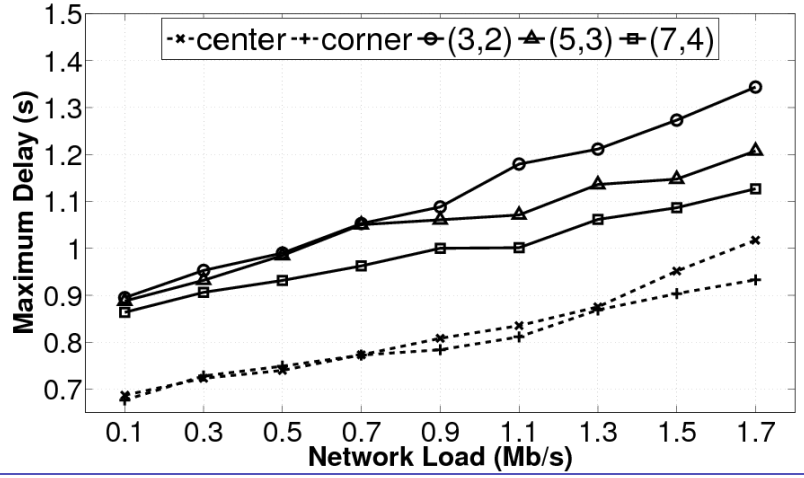
Random topology (35 nodes)



Grid topology (40 nodes)



Random topology (40 nodes)



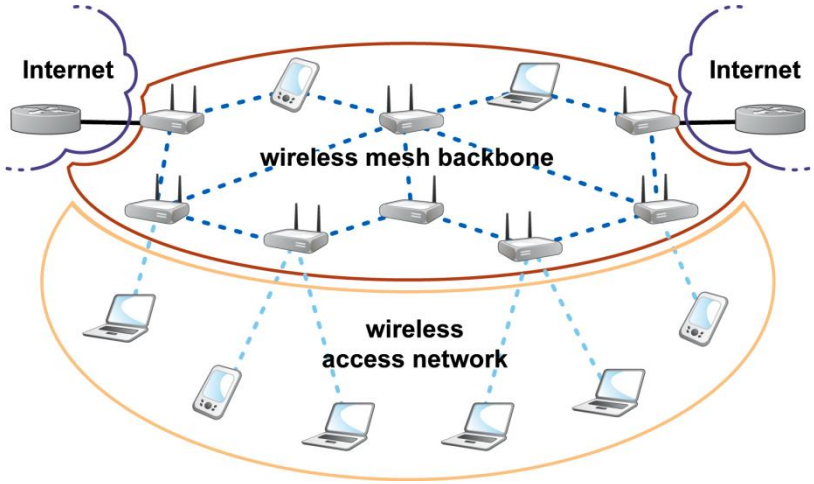
Thesis Proposals

Security

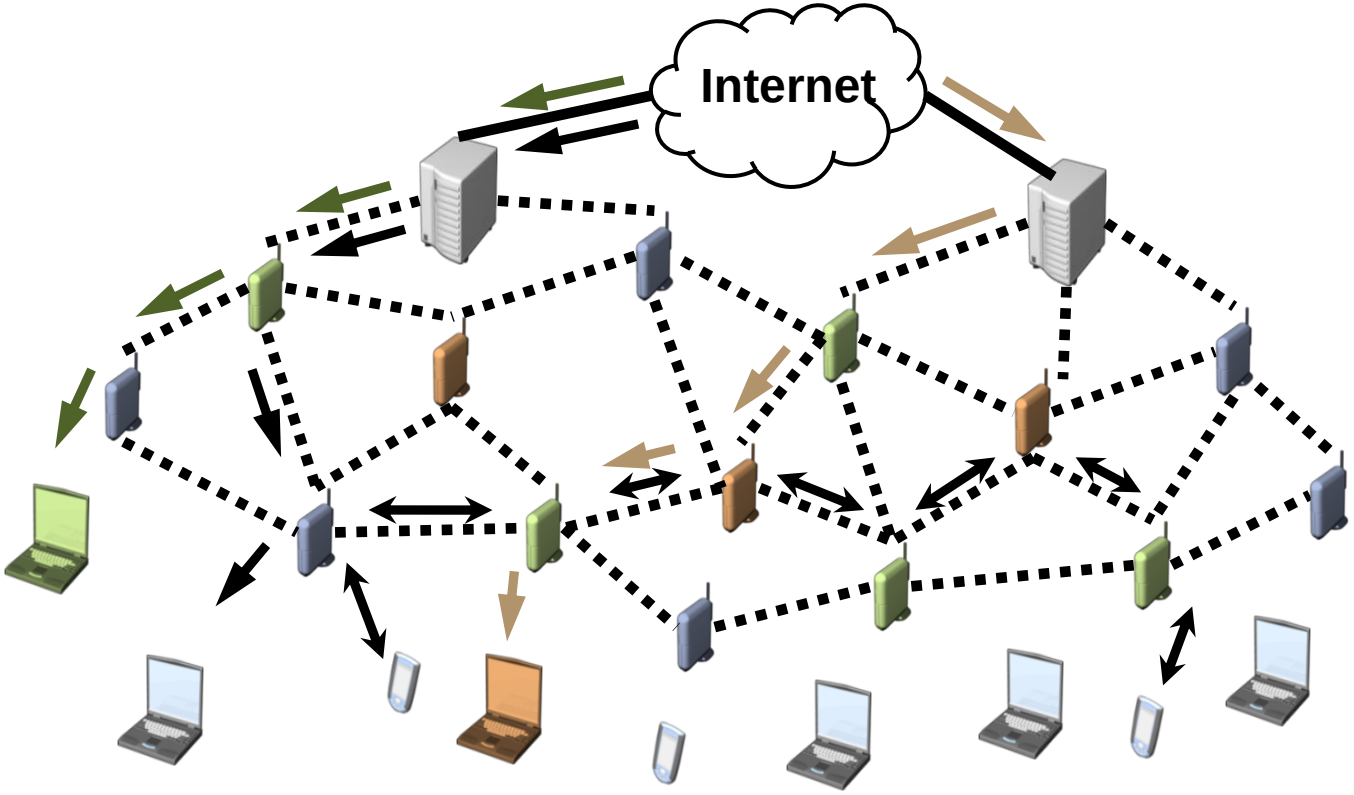
- MobiSEC has been implemented on embedded devices
  - Different architectures (x86, IXP4xx, MIPSEL)
  - Different WiFi chipsets (Broadcom, Atheros, Ralink)
- MobiSEC has been experimentally evaluated in different network scenarios/topologies:
  - Better performance than IPSec
  - Negligible packet loss

Security

- Wireless Router
  - OpenWRT
  - Single Radio
  - Broadcom Architecture
- Wireless Mesh Router
  - Multiple radio interfaces
  - **VIA** processor (x86)
  - Radio modules based on Atheros chipset
  - Omnidirectional and directional antenna
  - **GNU/Linux** OS
  - **OLSR** routing protocol
  - **NTP** protocol
  - Modified **Madwifi** Driver



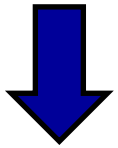
Survivability and Reliability



- Two different types of users:
  - Community Users
  - Customers

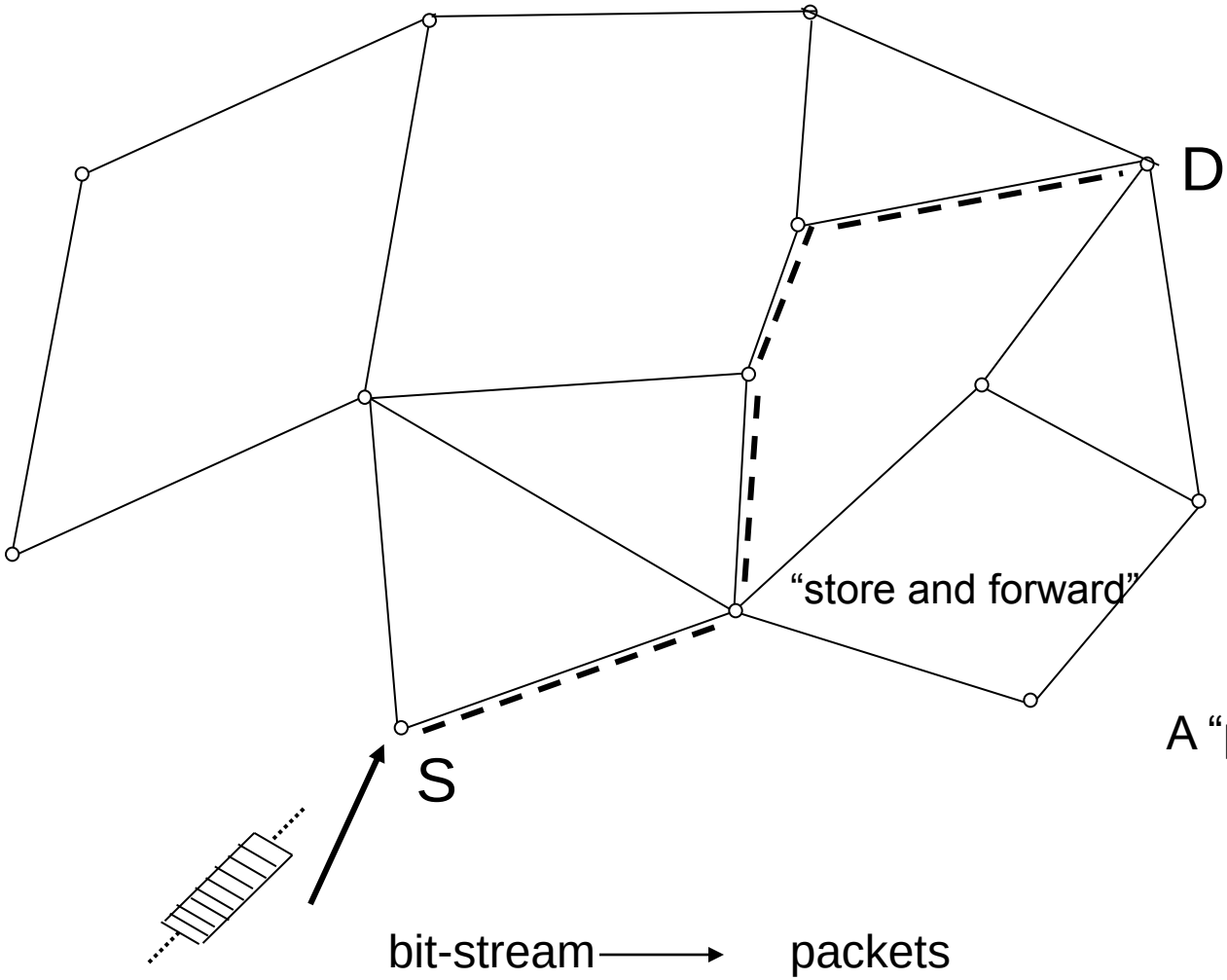
Survivability and Reliability

- Community users have two opposed interests
  - They compete against the customer devices which they serve for the available bandwidth
  - They are rewarded considering the customers satisfaction



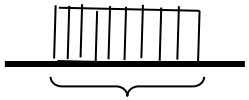
- Some mesh routers may misbehave:
  - Data Dropping Attacks
  - Lying Attacks

Store & Forward



packet "length" = N or T

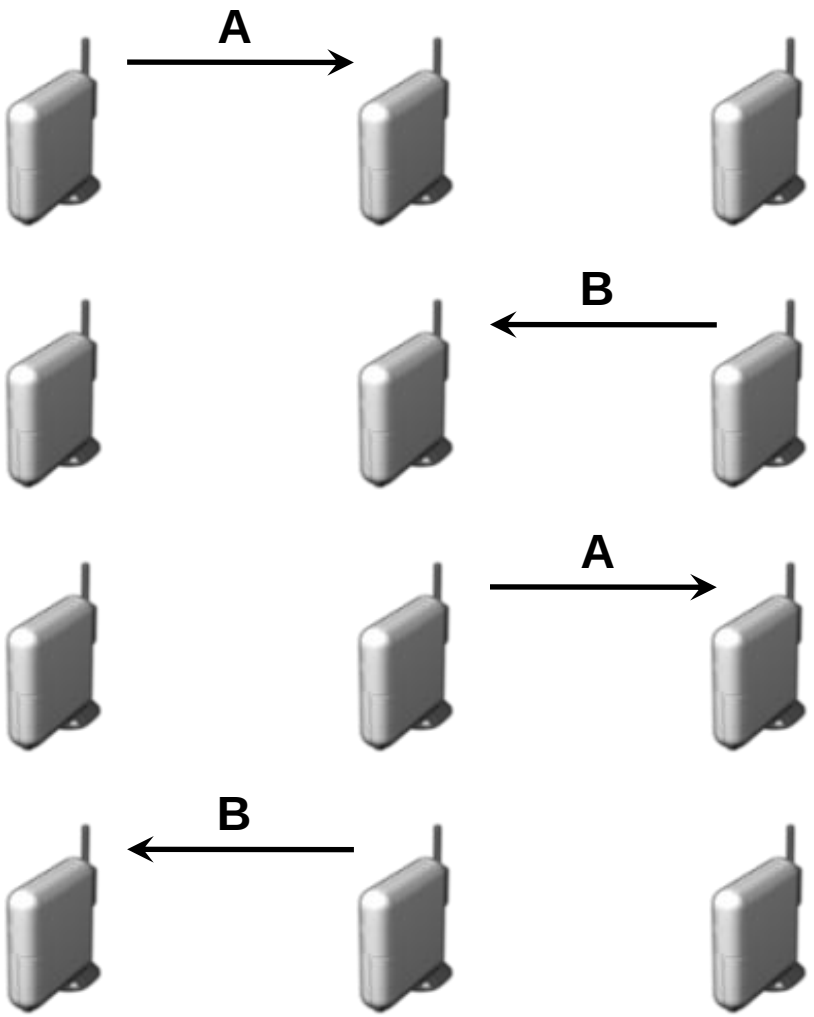
N = # of bits

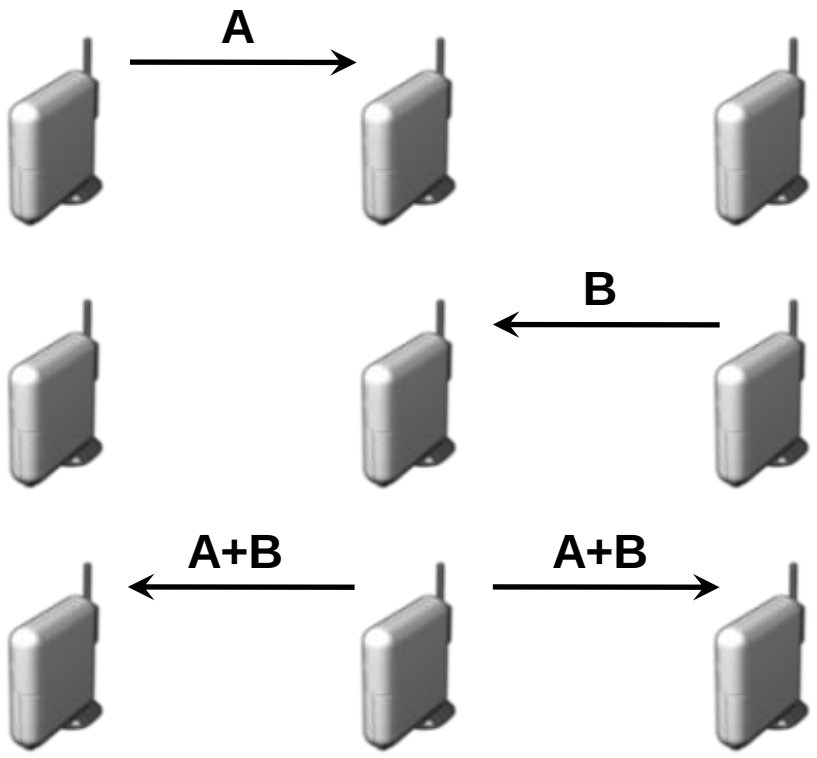


T = duration

A "packet" is a "monolith."

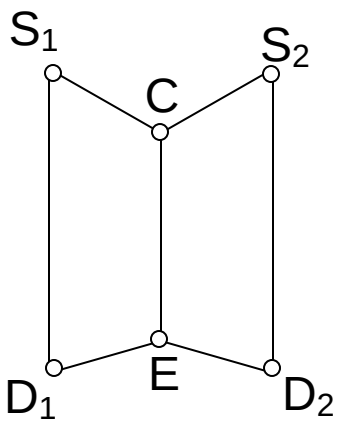
Network Coding





“+” = XOR
A+B+A=B
A+B+B=B

Network Coding



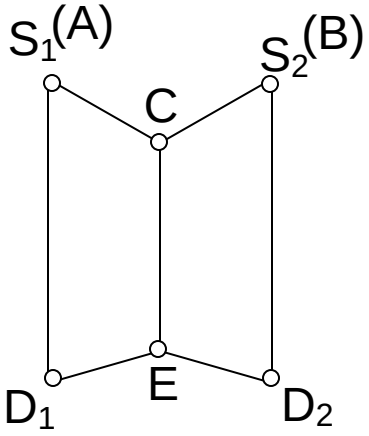
- S_1 wants to deliver packet A to both D_1 and D_2
- S_2 wants to deliver packet B to both D_1 and D_2
- Each link can carry one packet in each slot
  (capacity of each link = 1)

Q: How many slots are needed to complete this delivery under "store-and-forward"?

1: - S_1 sends A to C and D_1 (over S_1C and S_1D_1 respectively)
   - S_2 sends B to C and D_2 (over S_2C and S_2D_2 respectively)
2: - C sends A to E
3: - C sends B to E
   - E sends A to D_2
4: - E sends B to D_1

Answer: FOUR (4) SLOTS

Network Coding



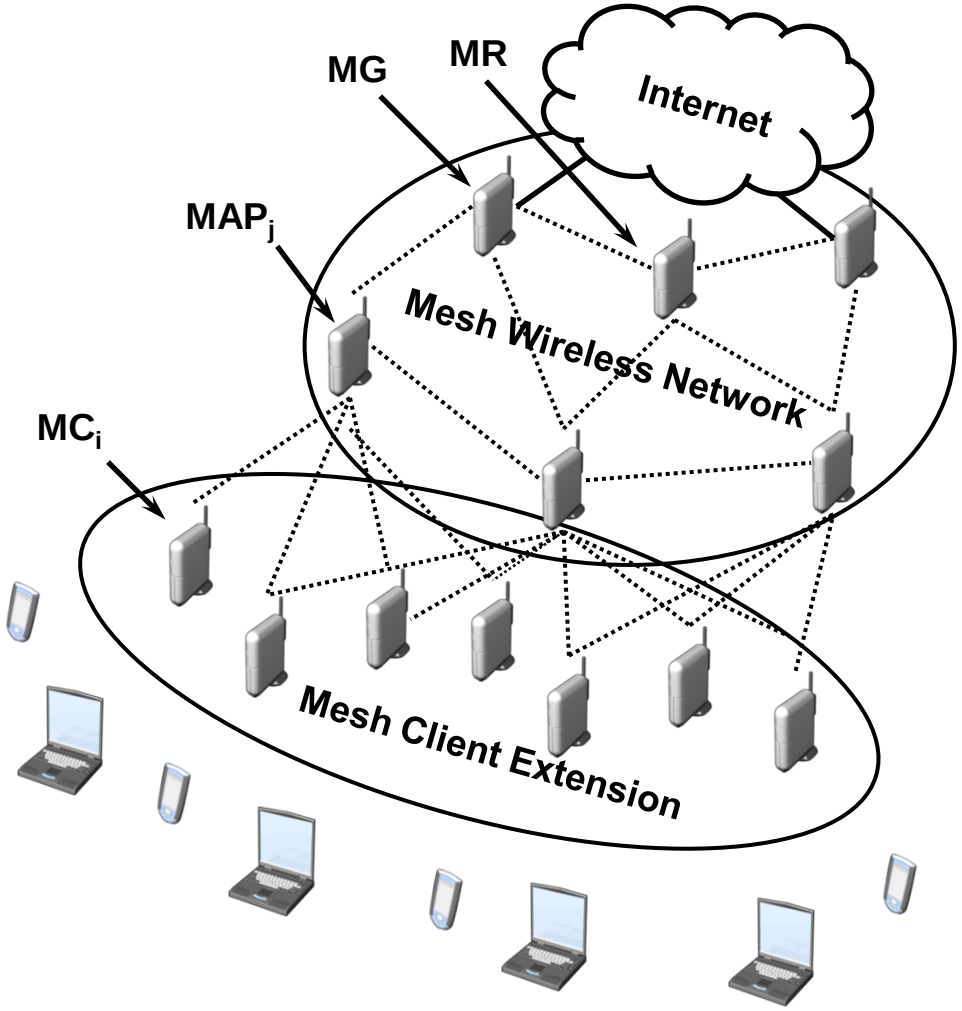
Q: How many slots are needed to complete the same delivery under, so called, "NETWORK CODING"?

- S_1 sends A to C and D_1 (over S_1C and S_1D_1, respectively)
- S_2 sends B to C and D_2 (over S_2C and S_2D_2, respectively)
- C sends A + B to E
- E sends A + B to D_1 and D_2 (over ED_1, and ED_2, respectively)

(D_1 and D_2 recover the missing packet by "X-OR"-ing A + B with the one they already have)

Answer: FOUR (3) SLOTS ➔ Savings: 25%

Network Design and Auction Theory



Network Design and Auction Theory

- Marketplace for the allocation of the WMN's available bandwidth
  - Truthful auction for revenue maximization of the WMN operator
- Marketplace for the available bandwidth of the users' BSs
  - Reverse auction for costs minimization
- Analysis and design of efficient algorithms to compute the best solutions
  - Guarantee that the selfish participants do not misbehave