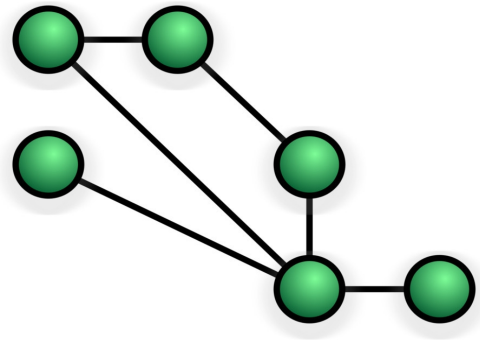
Relazione Progetto Gestione di Reti

Estensione NTOPNG per Batman-adv Dissector e Rilevazione Antenne Rete Mesh

Autori: ***Elif Beraat Izgordu, Francesco Staccini*** **A/A: 2014/2015**
Docente: ***Luca Deri***

Cos'è una Rete Mesh?

Una Wireless mesh network è una rete a maglie implementata tramite una WLAN (Wireless Area Network), cioè una rete locale senza fili.



Una rete a maglie è costituita da un numero di nodi, che fungono da ricevitori, trasmettitori e ripetitori. Questo tipo di infrastruttura è spesso decentralizzata (non ci sono server centrali).
È molto adattabile e resistente, dal momento che ogni nodo deve solamente trasmettere un segnale al massimo fino al nodo successivo o all'eventuale gateway di uscita.

Le reti mesh sono inoltre estremamente affidabili, poiché ogni nodo è connesso a molti altri nodi. Se un nodo viene meno alla rete, a causa di problemi hardware o qualunque altro motivo, i nodi vicini semplicemente cercano percorsi alternativi per trasmettere il segnale rivolgendosi ad altri nodi.

A Pisa è presente eigenNet, una Comunità di Rete Wireless Mesh, creata e mantenuta da eigenLab.

EigenNet

EigenNet è una rete wireless mesh libera e decentralizzata a Pisa. In altri termini si tratta di una rete wireless diffusa tramite antenne dislocate in vari punti della città. Questi dispositivi formano una vera e propria rete in cui le intersezioni o "nodi" sono in comunicazione wireless o via cavo tra di loro. La rete unisce tutti quelli che si connettono e li tratta alla pari, permettendo a tutti di accedere ai servizi offerti da tutti. Oltre ad essere paritaria, la rete è anche ridondante, cioè ogni dispositivo si occupa anche di rilanciare il segnale della rete.

Come già detto, i nodi partecipano alla rete tutti in modo paritario e sono connessi fra loro solitamente tramite wireless ma dove possibile anche tramite link ethernet. Sulle antenne viene installata una versione modificata di OpenWRT. Questa verisione usa il protocollo Batman-adv per gestire il routing tra le antenne ed ha la peculiarità di lavorare a Layer 2.

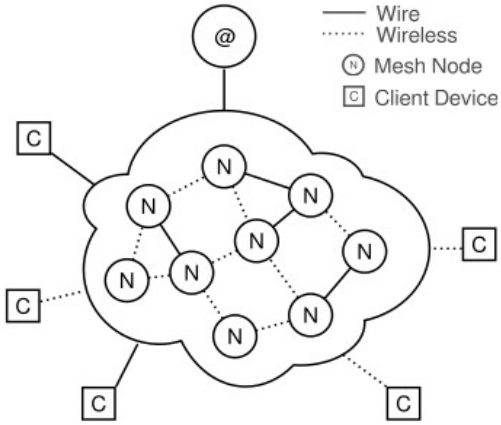
Per ulteriori informazioni: https://wiki.eigenlab.org/index.php/EigenNet

Batman-adv

Cos'è?

È un implementazione del protocollo di routing B.A.T.M.A.N. sotto forma di modulo del kernel linux (per questioni di performance), operante a livello 2 ISO/OSI.

A differenza di molti altri protocolli di routing che operano al livello 3 (tramite pacchetti UDP e routing table) Batman-adv opera interamente al livello 2, sia per quanto riguarda le informazioni di routing sia per il traffico dati stesso, il quale è sempre gestito da batman-adv. I pacchetti vengono incapsulati e spediti verso la destinazione utilizzando un sistema che emula uno switch virtuale alla quale sono attaccati tutti i nodi della rete.



Perché batman-adv?

Le reti mesh cambiano dinamicamente la loro topologia e sono basate su nodi non molto affidabili, per questo serve un approccio particolare.
B.A.T.M.A.N. suddivide le informazioni che riguardano il miglior percorso tra due nodi della mesh, tra i nodi che conpongono il percorso stesso.
Ogni nodo riceve, e mantiene, solo le informazioni riguardanti il "miglior nodo vicino" (next best hop) che va verso gli altri nodi. Così viene meno la necessità di sapere sempre l'intera topologia esatta per poter comunicare.

Batman-adv ha anche dei meccanismi per regolare, limitare e correggere il traffico per le info di routing, evitare loop all'interno della rete ed altro ancora.
Per avere tutte le informazioni riguardanti il protocollo visitare: http://www.open-mesh.org/projects/batman-adv/wiki

Struttura dei Pacchetti di Batman-adv (versione 2013/14)

Per motivi di comprensione del punto successivo presentiamo brevemente qui di seguito la struttura di alcuni pacchetti di batman-adv.

Batman-adv incapsula ogni pacchetto con il suo header prima di inviarli.
Come è illustarato sotto, l'header è composto da due sezioni:

```
 _____________________Batman-adv Header_____________________
/                                                            \
-------------------------------------------------------------------------
Dst.Mac|Src.Mac|Type|Packt Type|Version|TTL|......|Dst.Mac|Src.Mac|Type
-------------------------------------------------------------------------
\____________________/\_____________________________________/\____________________/
   First Ethernet          Batman Section           Second Ethernet
    II Section                                          II Section
```

La prima sezione Ethernet II:
Necessaria per il WiFi (o ethernet) per spedire pacchetti sul collegamento fisico tra i nodi vicini.
Tale sezione sarà diversa per ogni “hop” nella mesh

La sezione di batman-adv ha in comune tre campi, precisamente:
-Packet Type, che rappresenta la tipologia del pacchetto.
-Version, che rappresenta la versione di batman utilizzata.
-TTL(time to leave), contatore di time-stamp, serve per limitare il numero dei nodi sulla quale passerà il pacchetto.
Oltre a questi campi ce ne sono altri, che dipendono dal tipo di pacchetto batman-adv.

La seconda sezione Ethernet II:
Contiene il MAC dei dispositivi che stanno attualmemte parlando, come ad esempio i Client.

Per riuscire a capire quali sono le antenne a cui sono collegati i client, e chi sono quei stessi client, abbiamo preso in considerazione il pacchetto unicast (BATADV_UNICAST), e quello per la comunicazione broadcast (BATADV_BCAST).

I pacchetti unicast hanno in comune anche il campo destination, contenente il MAC dell'antenna alla quale è diretto il messaggio. Ci è utile in quanto quel MAC address è quello dell'antenna alla quale è collegato il client a cui è diretto un eventuale messaggio.
Il pacchetto broadcast invece, oltre ad altri campi, contiene il MAC address dell'antenna a cui è collegato il client che ha generato il messaggio.

NTOPNG

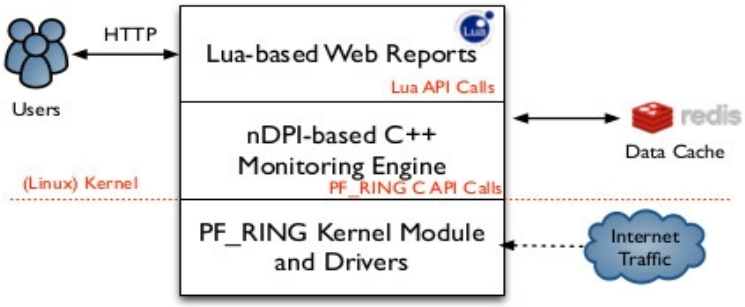
È un software open-source per monitorare ed analizzare il traffico di rete. Il nome deriva da “NTOP Next Generation” e riesce ad ottenere ottime performance con un basso uso di risorse rispetto al suo predecessore.

Rilasciato sotto licenza GPLv3 è disponibile per Unix, Linux, Mac OS X e Windows.

L'engine è scritto in C++ e l'interfaccia web in Lua. Inoltre si basa su un server Redis key-value (il più diffuso nella sua categoria) invece che un DB tradizionale, supporta la geolocalizzazione degli IP, sfrutta le librerie nDPI per un migliore trattamento dei protocolli ed è capace di eseguire il monitoraggio e analisi dei flussi in tempo reale sugli host connessi.

Link al sito: http://www.ntop.org/products/traffic-analysis/ntop/

Architettura di NTOPNG:



Dissector:

Le modifiche apportate ad NTOPNG intervengono nel momento in cui il software disseziona il pacchetto per trarne informazioni.

Appena arrivato alla decodifica dell'header Ethernet II, quindi layer 2, il programma controlla che il protocollo usato sia quello di batman-adv. Una volta stabilito ciò controlla quale tipo di pacchetto è tra quelli usati da batman-adv, se è uno dei tipi che ci interessa (broadcast o unicast) allora andrà ad estrarre il MAC address dell'antenna (sorgente o destinataria) per confrontarlo con il MAC address presente nella seconda sezione Ethernet II.
Se i MAC sono diversi vuol dire che il messaggio è diretto ad un client collegato ad un'antenna oppure proviene da un client collegato ad un'antenna

Gli header degli altri pacchetti vengono ignorati, ma resta la possibilità di utilizzare le informazioni che portano con loro in futuro per altri scopi.

Questo meccanismo è stato implementato per le versioni di batman-adv 2013 e 2014.

Grazie a questo trattamento delle informazioni NTOPNG riesce a generare dei report riguardanti le antenne della rete mesh, alle quali sono collegati i client che stanno generando traffico.