

Caching Alternatives for a MANET-Oriented OCSP Scheme

Giannis F. Marias,
Konstantinos Papapanagiotou,
Panagiotis Georgiadis

e-Sec Group / e-Gov Lab
Dept. of Informatics and Telecommunications
University of Athens
Greece
TK15784





Outline

- Ad Hoc Wireless Networks (MANETs)
- Certificate Status Information in MANETs:
The ADOPT scheme
- Cache Management Issues in ADOPT
 - Cache Placement
 - Cache Update and Deletion
- Time Thresholds for ADOPT
- Remarks



Ad Hoc Wireless Networks

- Self-configured, peer-to-peer, multi-hop networks
- dynamic and open:
 - nodes constantly join / leave the network
 - the transfer medium (electromagnetic spectrum) is unlicensed and publicly available
- varying topology (mobile hosts)
- 2 types:
 - AANs (Autonomous Ad hoc Networks)
e.g., 802.11 Independent Basic Service Set (IBSS)
 - CANs (Connected Ad hoc Networks)
e.g., 802.11 Extended Service Set (ESS)
- selfish nodes
- malicious nodes



Certificate Schemes in MANETs

- Centralized CA: single point of failure and more...
- Distributed key management and CA functionality based on threshold cryptography
- CA access via GSM/GPRS in CANs
- Offline CA
- Applications:
 - ARAN
 - SAODV
 - TRM
- Certificate Status Information
 - CRLs (problems: size, periodicity, etc.)
 - Online Certificate Status Protocol (OCSP)



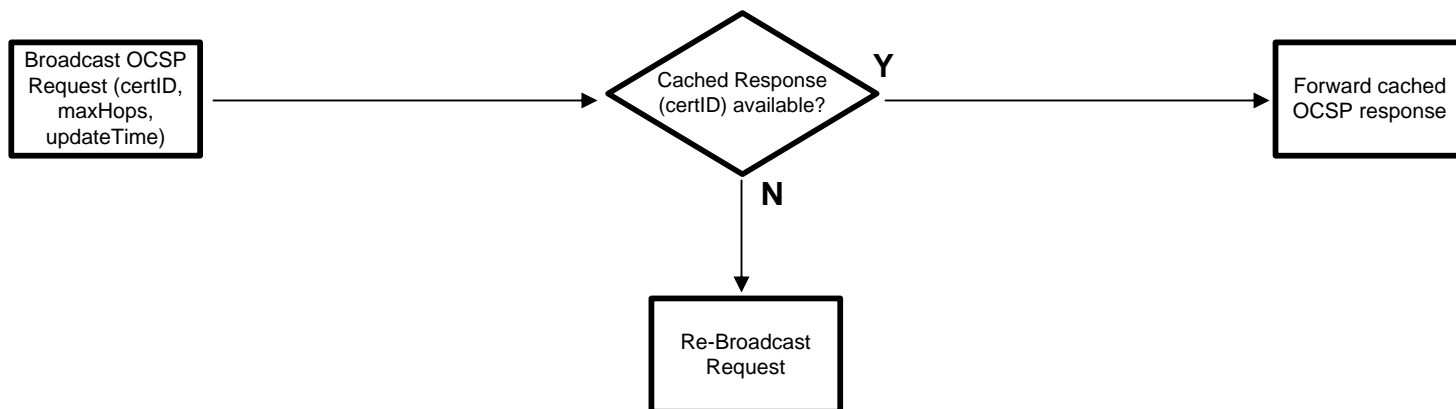
ADOPT

(Ad hoc Distributed OCSP for Trust)

- distributed, on-demand, always-available, OCSP based scheme
- cached OCSP responses
- 3 kinds of nodes:
 - Server-nodes: announce revocation status (ex. OCSP Responders)
 - Caching-nodes: cache and forward OCSP responses
 - Client-nodes: request the status of a certificate
- DSR-like mechanism for caching-node location



ADOPT Flowchart



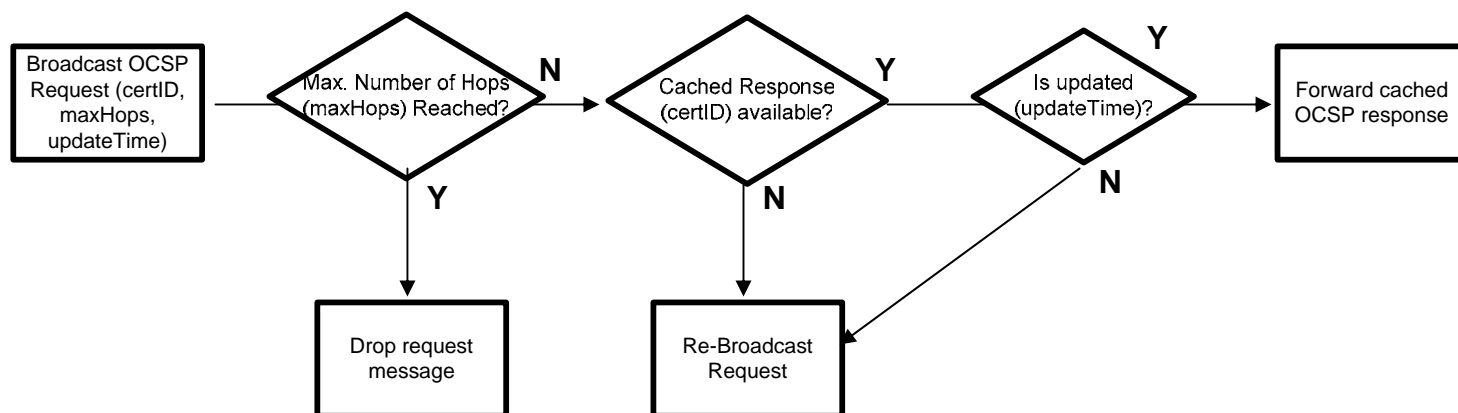


ADOPT Issues

- Advantages
 - Cached Responses are pre-signed
 - nodes don't have to compute signatures
 - signing keys are safe even if a caching-node is compromised
 - Messages' size – low comm. cost
 - saves bandwidth and resources
- Issues
 - “Request Loop”
 - Response Freshness
 - Cache Update



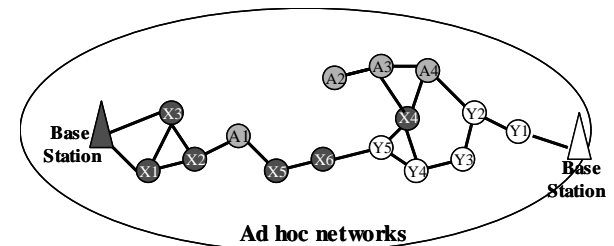
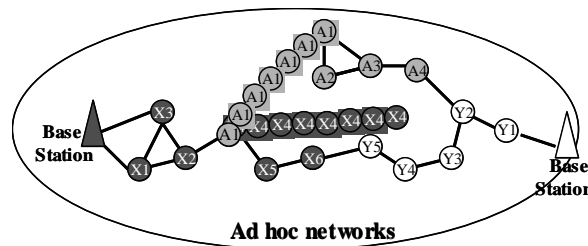
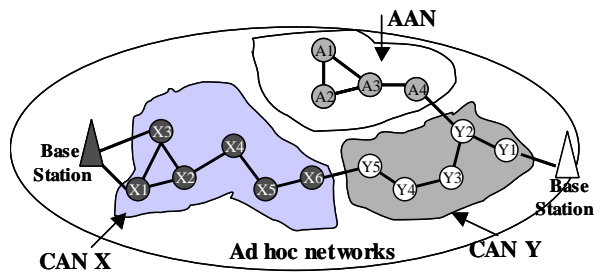
ADOPT Flowchart





Cache Management Issues

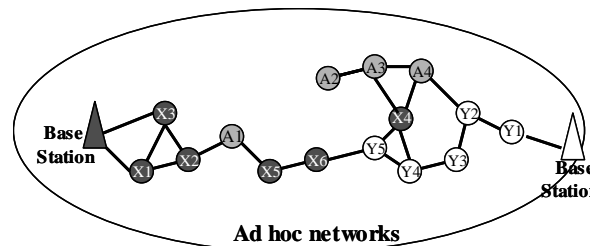
- Node selection – Cache Placement Policy
 - criteria that force peers to act as Caching or Server nodes
 - minimize:
 - access latency
 - energy costs
- Cache Update Policy:
 - writing new responses into the existing cache
- Cache Deletion Discipline





Cache Placement Policy

- Candidate caching nodes: those with adequate resources (CPU, storage, memory and energy)
- Caching at the Edges of Grids (e.g. A2, X1, Y1)
 - these nodes forward packets between grids (also nodes that keep a direct link with base stations)
 - node A_i caches responses according to the number of grids that it is associated with: $NCx(A_i, t) \geq 1$
- Caching at Hub Nodes (e.g. X4)
 - A_i calculates number of vicinal nodes $NR(A_i, t)$
 - central (hub) nodes cache status information with higher rate
- Caching at High Mobility Nodes

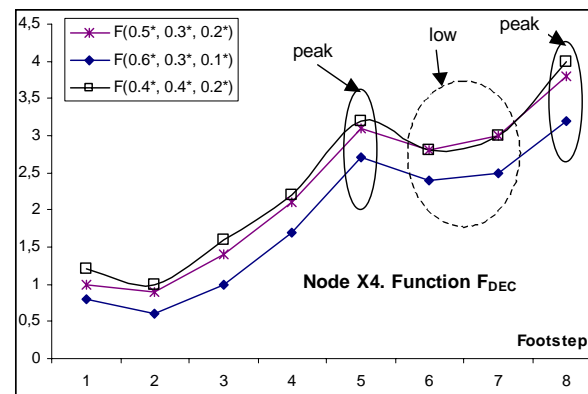
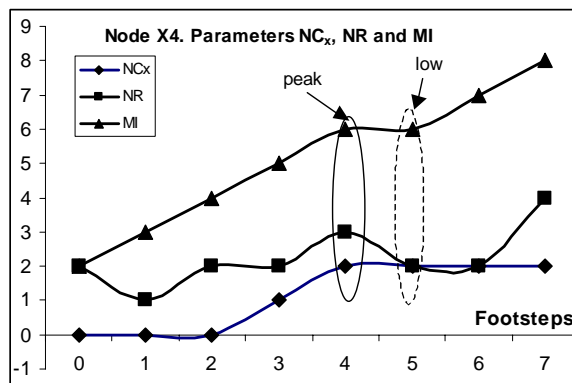




Cache Placement Policy (2)

- Caching at High Mobility Nodes
 - a node with high mobility is considered as a candidate to cache OCSP responses
 - node A_i senses $MI(A_i, t)$ neighbours per time unit (MI: Mobility Index)
 - a node performs “greedy” or “selective” caching according to this function:

$$f_{DEC}(A_i, t) = a \times NC_x(A_i, t) + \beta \times NR(A_i, t) + \gamma \times MI(A_i, t)$$





Cache Update and Deletion Policy

- OCSP date fields:
 - *thisUpdate*
 - *nextUpdate*
 - *producedAt*
- Caching-nodes sensing a response with a greater *thisUpdate* value, refresh the corresponding cache entry
- Entries illustrating an exceeded *nextUpdate* value are only deleted when there is no storage space to cache new responses



Caching Strategies

- Caching strategies:
 - **Greedy Caching State**
 - a node caches all sensed OCSP responses
 - **Selective Caching State**
 - a node caches k out of K sensed responses
 - a node caches a response after m appearances
 - **No Caching State**
 - a node does not perform any caching
 - it may delete expired responses
- The f_{DEC} function controls the transition between caching states
 - sudden increment forces transition from Selective to Greedy
 - when increased smoothly the node stays in its current state



Time Thresholds: Time To Live

- Two Critical Parameters:
 - Time To Live (TTL)
 - Waiting Window
- Time To Live
 - maximum number of nodes the request can pass through before finding a response
 - how it is computed
 - a node wishes to be informed about the status of another node's certificate before the latter gets out of range
 - A_j receives certificate C_i from A_i at time TIN_i . C_i is valid until TVA_i and A_i 's speed is S_i .

$$TTL_i \geq \left\lceil \frac{(TVA_i - TIN_i) \cdot S_i}{R} \right\rceil$$



Time Thresholds: Waiting Window

- time that a node has to wait to receive a response
- how it is computed
 - node A_j collects statistics of the one-hop delay when issuing OCSP requests.
 - A_j calculates the time interval required for a typical OCSP request packet to be successfully retransmitted by each of the neighbors
 - $D_{OCSP_k}(j,m)$ denotes the time interval between the arrival of the k -th OCSP request packet to the transmit queue of A_j and the successful transmission of this packet from the neighbor node A_m .



Waiting Window (2)

- For M one-hop neighbors, the maximum one-hop delay that A_j measures is: $D_{onehopMAX} = \max_{m=1\dots M} [D_{OCSP_k}(j, m)]$
- a node can record the roundtrip delays of previous OCSP requests with identical TTL parameter values. For $TTL=r$: $D_{RTD_MAX}(r) = \max_r [RTD_{OCSP_r}]$
- A_j evaluates WW as follows:

$$WW(r) = \max \{ r * D_{onehopMAX}, D_{RTD_MAX}(r) \}$$



Summary

- **ADOPT**
 - revocation scheme for MANETs based on OCSP caching
 - Fast, light, robust, reliable, distributed and always-available
- **Cache Management and Time Thresholds**
 - aim: provide up-to-date responses in an efficient, delay-sensitive way
 - cache placement update and deletion policies
 - two critical time parameters: TTL and WW
- **Future Work**
 - simulation scenarios
 - goal: enhance performance metrics (delay, overheads)
 - propose specific values for optimal performance



Q & As

conpap@di.uoa.gr