Proactive Key Distribution to support fast and secure roaming

Arunesh Mishra, Minho Shin, William Arbaugh

University of Maryland

College Park

Insun Lee, Kyunghun Jang

Samsung Electronics

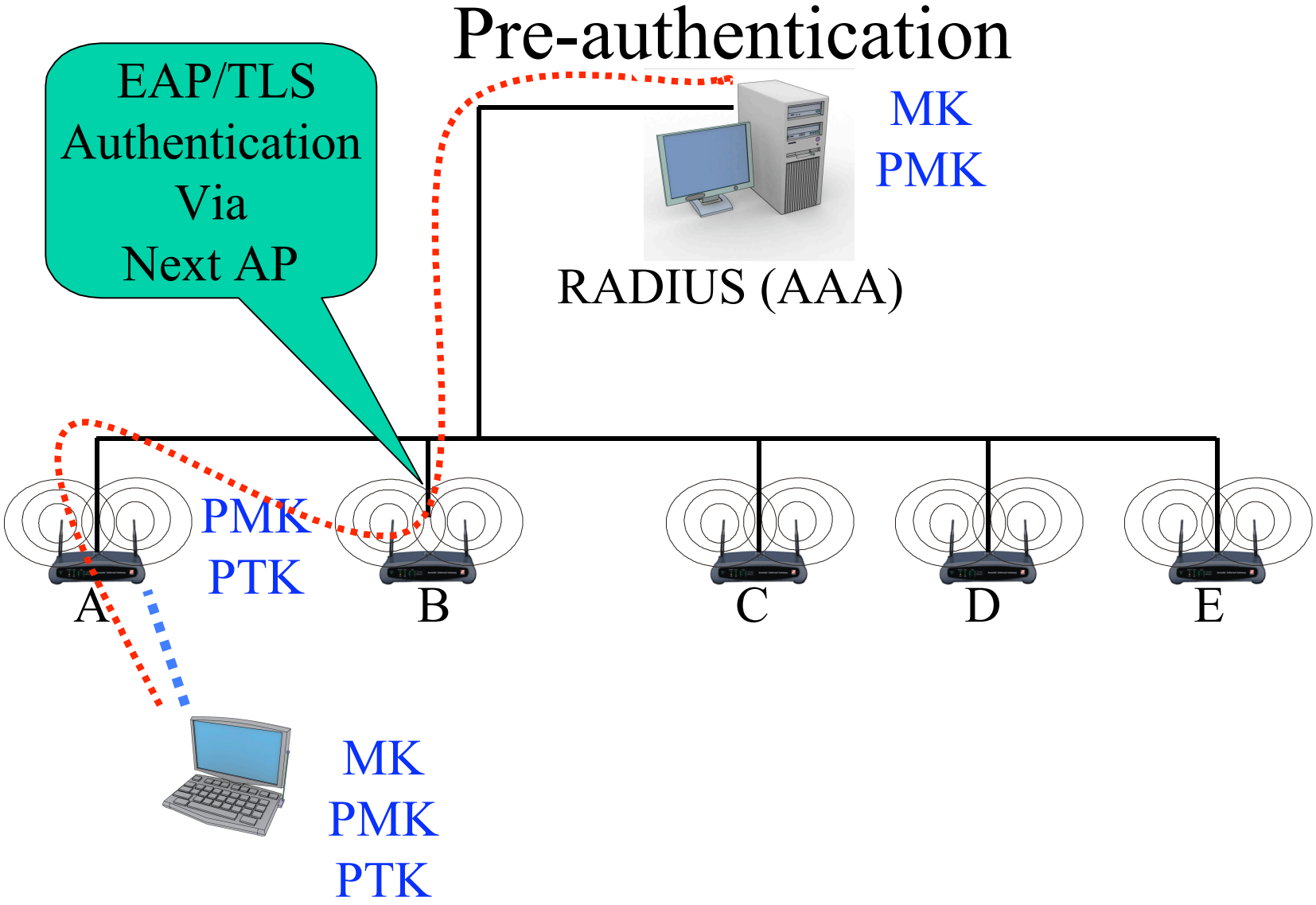
TGi MUST Support Fast Roaming

- Otherwise non-standard and non-vetted solutions will evolve….creating potential “brand” problems.
- Transparent roaming was one cause of exponential growth in the cellular market.
- Interworking is around the corner.

Backend Requirement

- Protocol MUST be standardized within IETF
  - This requires that key material NOT leave the AS.
  - This means that the protocol should fit within current and future AAA practice.

Pre-authentication



Problems with Pre-Auth

- Expensive in terms of computational power for client, and time (Full EAP-TLS can take seconds depending on load at RADIUS Server). TLS-Resume will make things faster, but other problems persist.
- Requires well designed and overlapping coverage areas
- Can not extend beyond LAN
- No opportunity for Interworking

Goals

- Permit fast roaming without reducing overall security
- Fast roaming occurs when the total cost of Layer 1-3 hand-off times is less than 50ms (Ideally 35ms).

TGi Fast Roaming Goals

- Handoff to next AP SHOULD NOT require a complete EAP/TLS re-authentication.
- Compromise of one AP MUST NOT compromise past or future key material, i.e. *perfect forward secrecy, and with stand known key attacks*.

TGi Trust Assumptions

- AAA Server is trusted
- AP to which STA is associated is trusted. All other AP's are untrusted.

Only Three Ways to meet TGi Goals

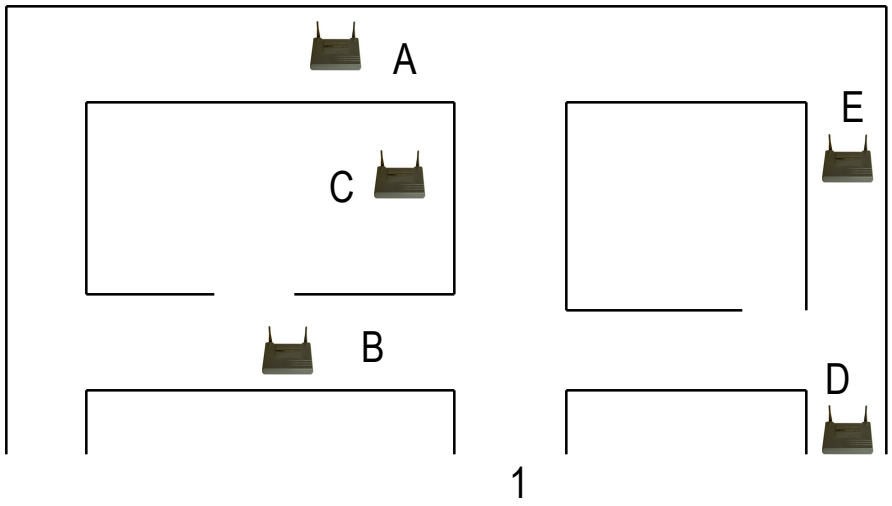
- Exponentiation support for asymmetric cryptographic operations at AP, or
- Trusted Third Party, i.e. Authentication/Roaming Server

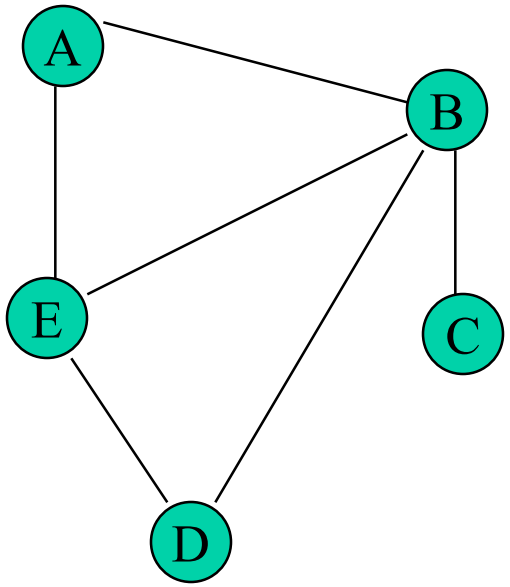
Proactive Key Distribution (TGi)

- Extend Neighbor Graphs and Proactive Caching (IEEE 11-02-758r1.ppt) to support key distribution by the AS
- Eliminates problems with sharing key material amongst multiple APs
  - Easily extended to support WAN roaming
  - Extendable to support Interworking

Neighbor Definition and Graph

- Two APs *i* and *j* are neighbors iff
  - There exists a path of motion between *i* and *j* such that it is possible for a mobile STA to perform a *reassociation*
  - Captures the '*potential next AP*' relationship
  - Distributed data-structure i.e. each AP or AS/RS can maintain a dynamic list of neighbors





AP Neighborhood Graph – Automated Learning

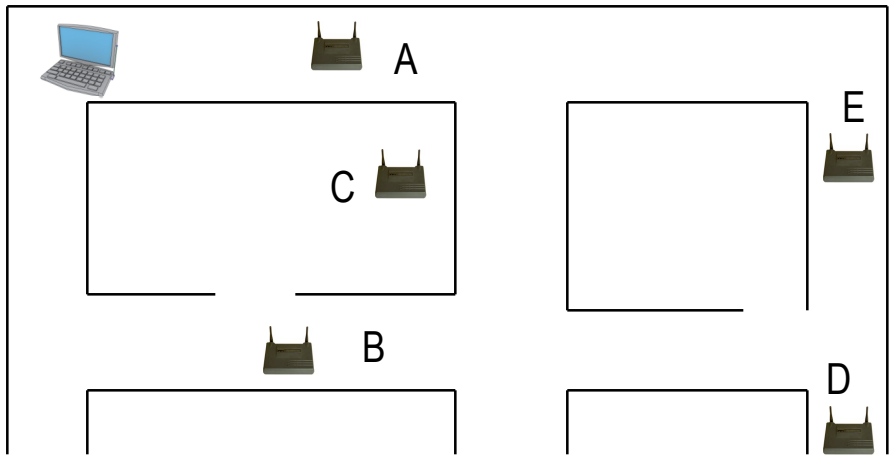
- Construction
  - Manual configuration for each AP/RS or,
  - AP/RS can learn:
    - If STA *c* sends *Reassociate Request* to AP *i*, with old-ap = AP *j* :
    - Create new neighbors *(i,j)* (i.e. an entry in AP *i*, for *j* and vice versa)
    - Learning costs only one '*high latency handoff*' per edge in the graph.
    - Enables mobility of APs, can be extended to wireless networks with an ad-hoc backbone infrastructure.
    - Dynamic implementation using LRU replacement permits invalid and stale entries to time out.

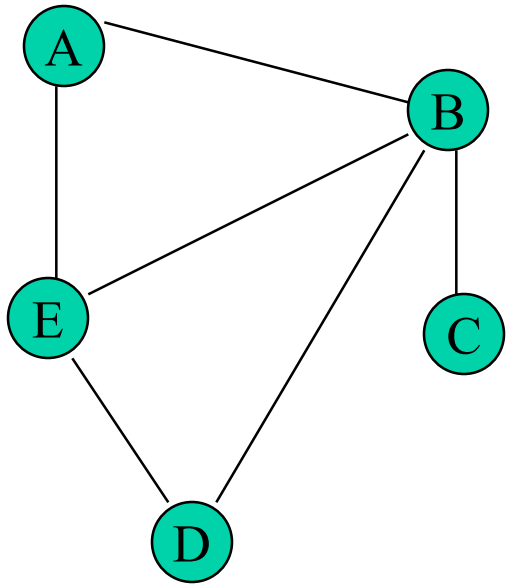
Graph Synchronization

- The graph's state at the accounting server is updated by:
  - Accounting-Request messages from the current AP (draft-congdon-radius-8021x)

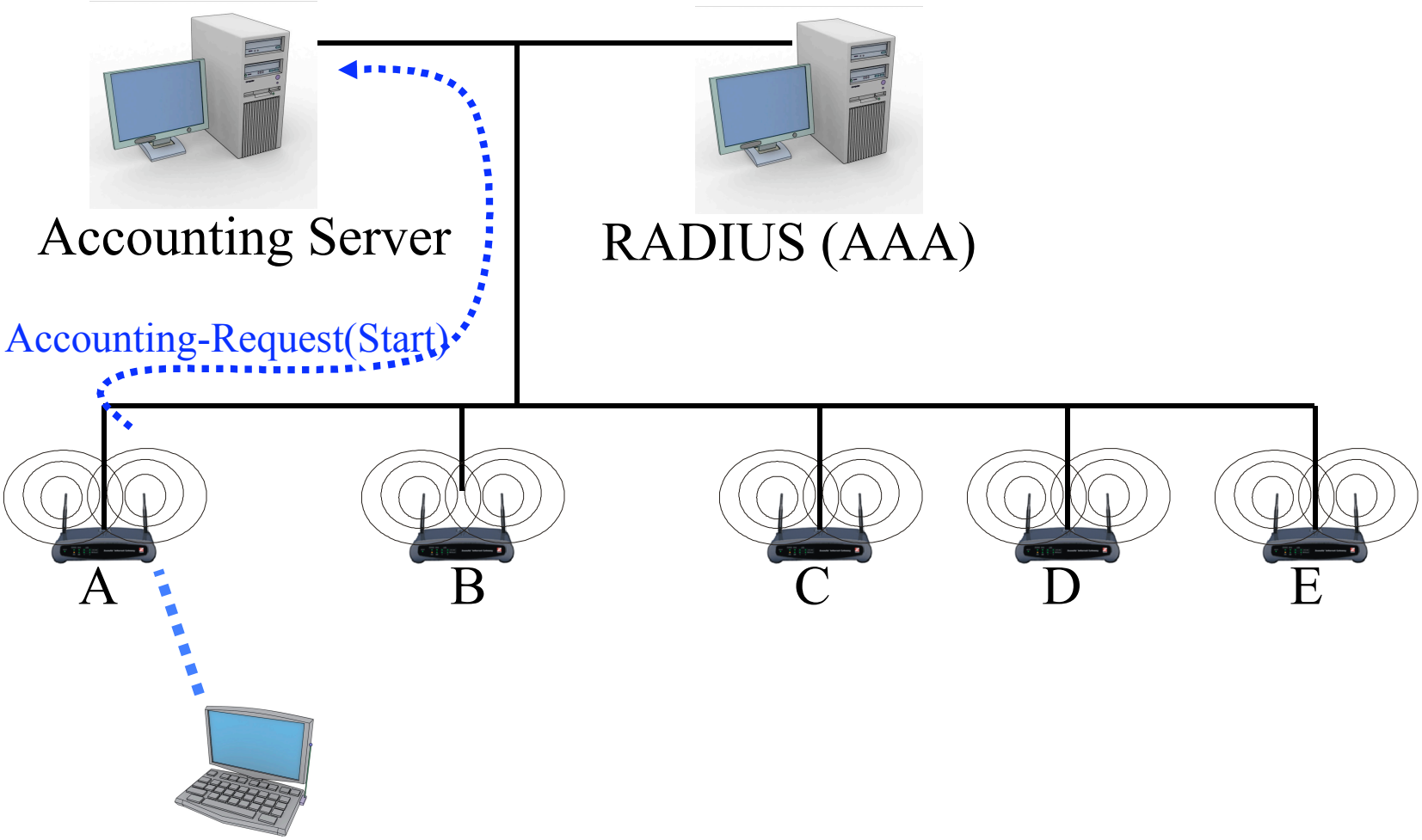
Roaming Example

- Given the following infrastructure with associated neighbor graph with STA about to associate to AP A.

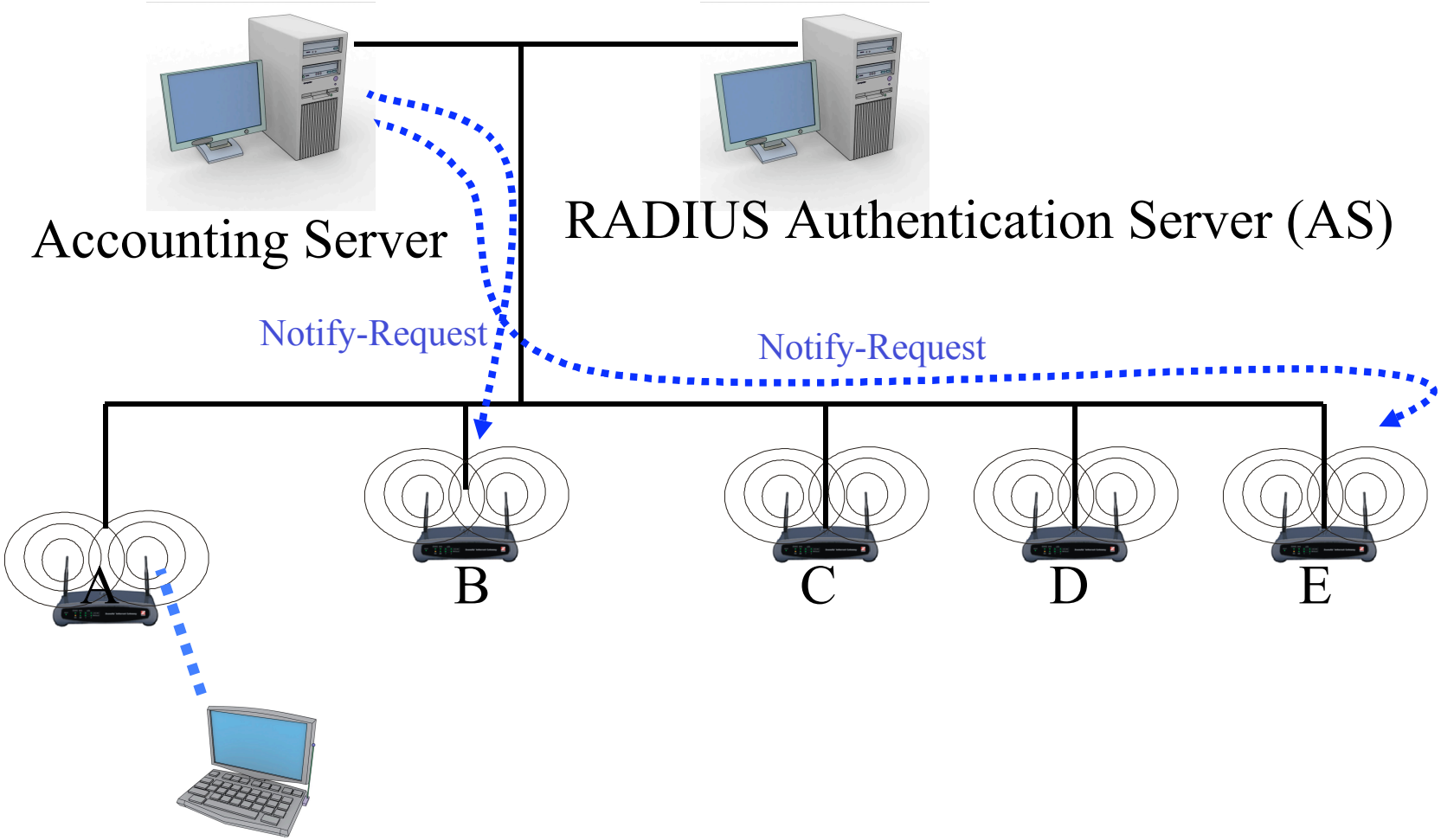




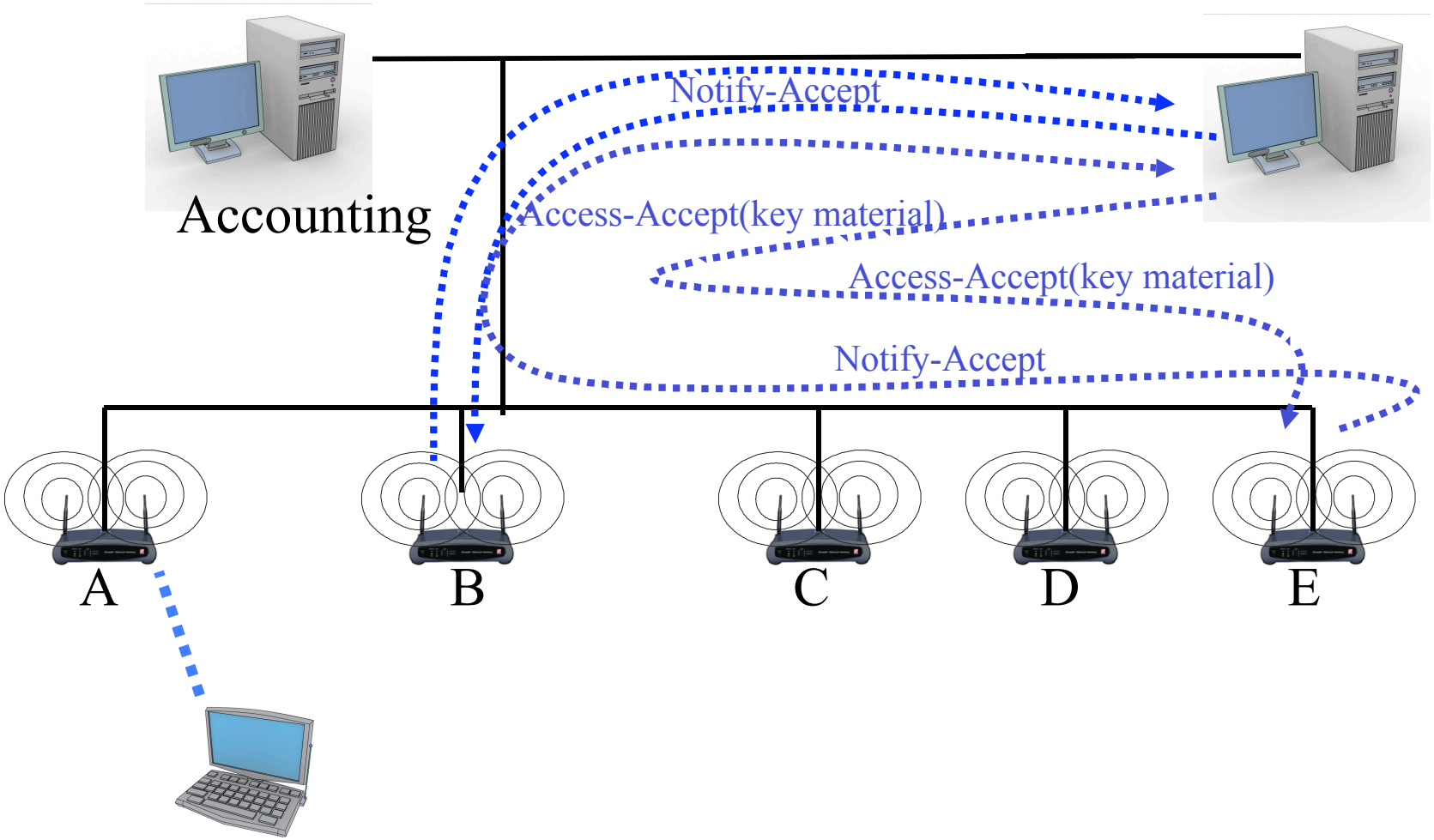
Post Authentication and 4-handshake



Proactive Key Distribution



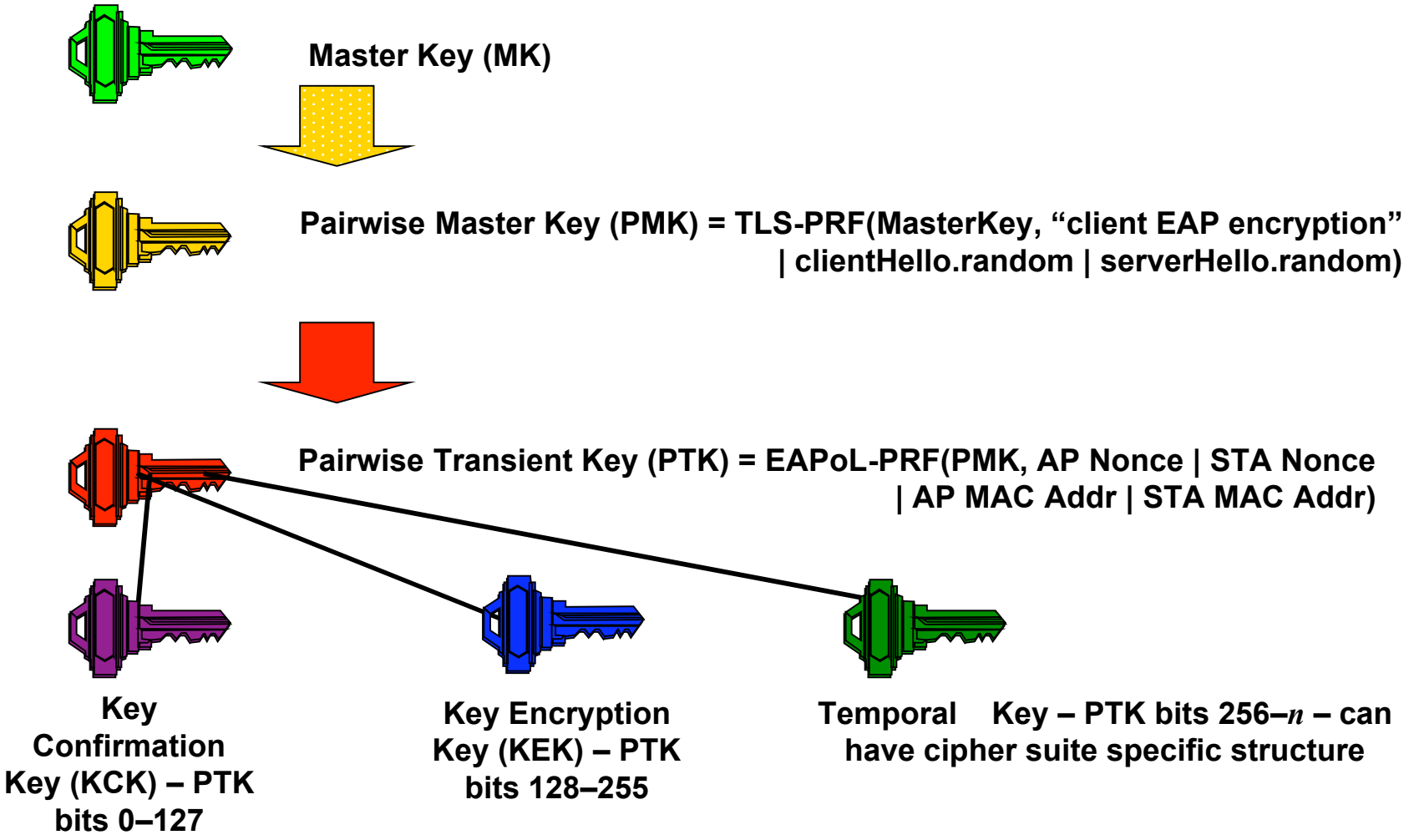
Proactive Key Distribution Post Authentication



AP Actions on Notify Request

- Dynamic Keys, i.e. PMK changes per roam.
  - AP MUST send an ACCESS-REQUEST to AS
- Static Key, i.e. PMK is unique per AP but never changes.
  - Nothing unless authorization is required.

TGi Pairwise Key Hierarchy Review



PMK Generation

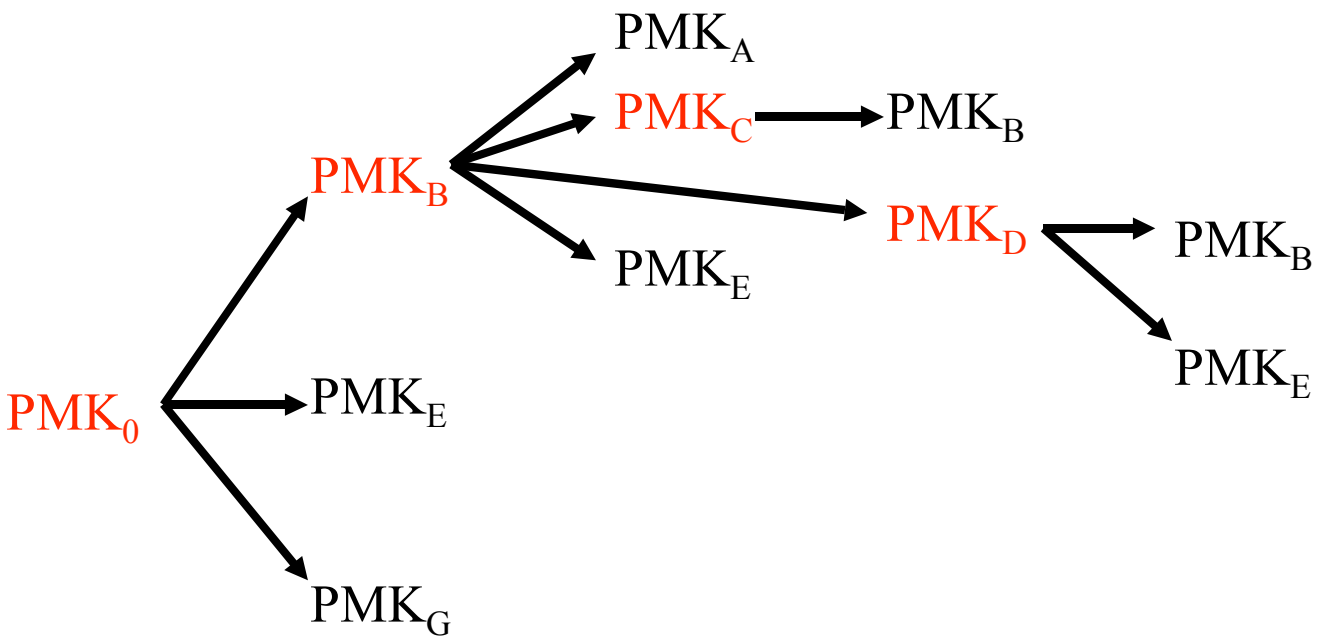
- Each AP is given a unique PMK per roam, or generation.
- The PMK for the AP for that generation becomes that generation's PMK.

PMK Generations

- Generation:
  - PMK_0 = TLS-PRF(MK,"client EAP encryption", client-Hello.random, serverHello.random)
  - $PMK_{1\text{-}B}$ = TLS-PRF(MK, PMK_0,AP_B-MAC-Addr)
  - $PMK_{1\text{-}E}$ = TLS-PRF(MK, PMK_0,AP_E-MAC-Addr)
  - $PMK_{2\text{-}C}$ = TLS-PRF(MK, $PMK_{1\text{-}B}$,AP_C-MAC-Addr)

Synchronization Example

STA roam pattern: A→B →C ≈ D



Synchronization

- One of two cases exist:
  - 1) AP and STA share correct generation of PMK, or
  - 2) They do not.
- STA ALWAYS has the correct PMK.
- The hand-shake determines which case exists and if the situation is case 2) then a full re-auth is completed. NOTE: The ~10ms it takes to determine the appropriate case is lost in the "noise" of a full EAP/TLS at ~800ms.
- Can be greatly improved by enforcing an invariant that a PMK for a STA only exists at current and neighboring AP's (requires new message to draft-irtf-aaaarch-handoff-03.txt).

How do the AP and STA know that Fast Roaming is supported?

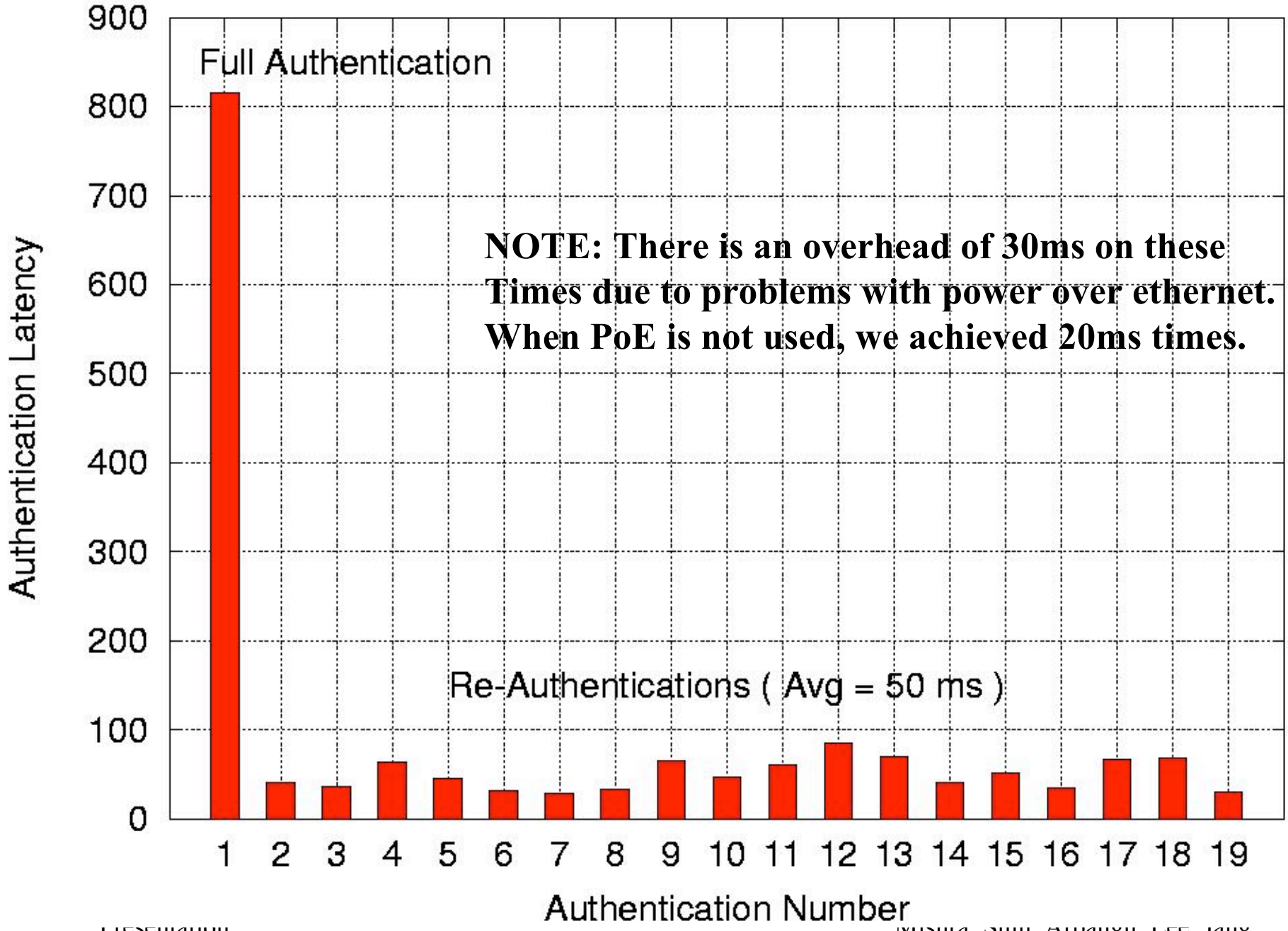
- STA asserts that it supports fast roaming by setting a bit (use of the current reserved bits) in the RSN information field element in the REASSOCIATION-REQUEST.
- AP asserts the same bit in the REASSOCIATION-RESPONSE *if and only if* the AP supports fast roaming and it is provisioned with a derived PMK for the STA.
- If either bits are unset, then a full reauthentication MUST be done.

Implementation Results

- Significantly easier to implement than expected.
- Little traffic increase on distribution system.
- Synchronization MUCH easier than expected.
- Reduced hand-off times from a full EAP/TLS by a factor of 40!
- Full paper available at:

  http://www.cs.umd.edu/~waa/pro-key.pdf

  – Please do not re-distribute paper


Full Authentication
NOTE: There is an overhead of 30ms on these Times due to problems with power over ethernet. When PoE is not used, we achieved 20ms times.
Re-Authentications (Avg = 50 ms)
Authentication Latency
Authentication Number
0
100
200
300
400
500
600
700
800
900
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Advantages

- Permits fast roaming with little to no changes within current draft.
- Provides similar level of security as a full re-authentication
  - New PMK
  - Interaction with AAA server to authorize access
- Actually has been implemented!

Disadvantages

- Does require server changes, but pre-auth can be used until server changes occur (besides server changes are minor).

What needs to be done?

- Adjust to EAP key hierarchy.
- Use PMK name (once defined) rather than generation.
- Add invariant message.
- Add fast-roam bit to IE.

Maximum STA Velocity

For the Notify and PMK install to occur in time, we need:

2 RTT + handshake < D/v

Where:

D = coverage diameter

v = STA velocity

RTT = round-trip time from AP to AAA server, including processing.

Assuming D=100 ft, handshake = 10 ms, and RTT = 100ms, we get:

v = 100 ft/ (200ms + 10 ms) ~ 500 ft/sec = **Mach 0.5!!**

Conclusions

- Provided an overview of various options
- Provided a protocol that:
  - Can support high speed roaming, meets IETF requirements (draft-arbaugh-radius-handoff-00.txt),
  - Is independent of the type of key management/derivation used (static or dynamic), i.e. can IEEE11-03-008r0-I or the method in the information slides of this presentation.
  - Is independent of the type of hand-shake used.

Acknowledgements

- Bernard Aboba assisted with the best way to integrate Proactive key distribution with RADIUS.
- The maximum STA velocity calculation is from Bernard Aboba.
- Jesse Walker pointed out potential synchronization problems in an earlier version of proactive key distribution.