SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks

Ishwar Ramani

Stefan Savage

INFOCOM 2005

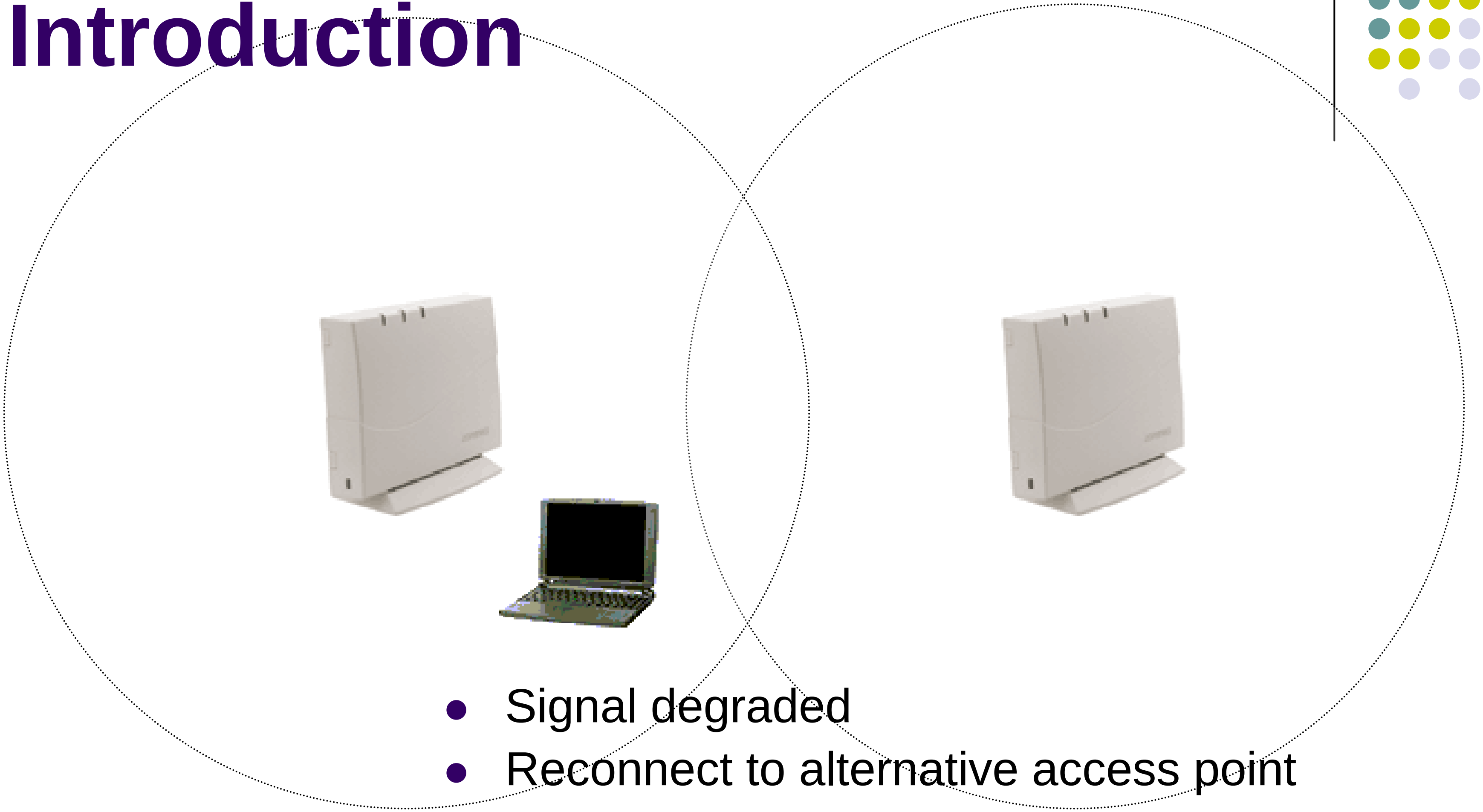
Outline

- Introduction
- SyncScan algorithm
- Implementation and Evaluation
- Conclusion

Introduction

- The 802.11 standards dictate that handoff should be managed autonomously and independently by each client
- 802.11-based system only monitor the signal to their current access point and handoff only after service degrades below an acceptable threshold

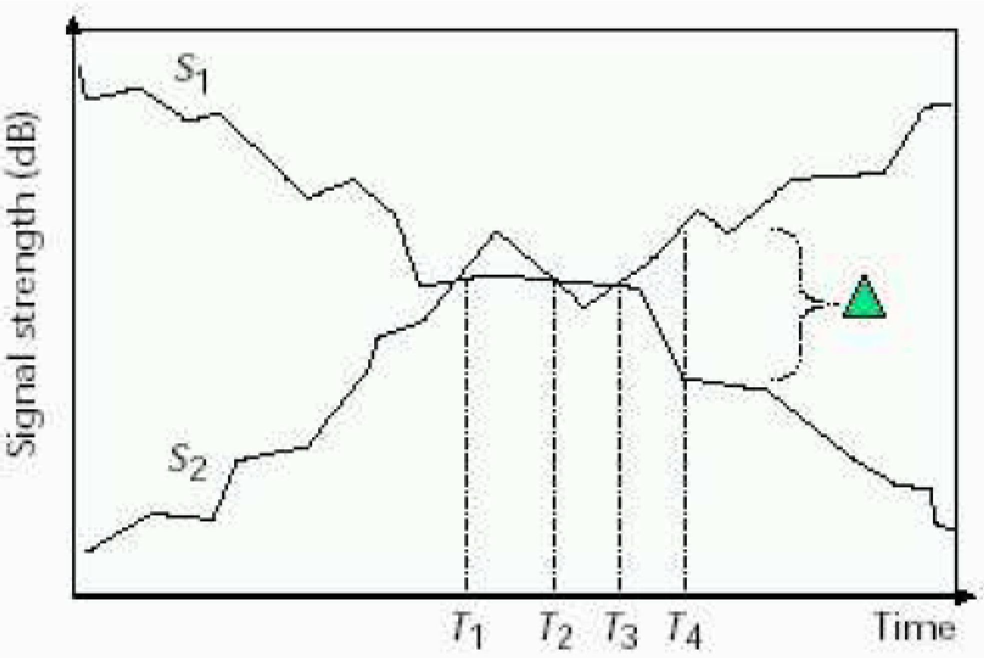
Introduction



- Signal degraded
- Reconnect to alternative access point

Introduction

- Monitor the current signal-to-noise (SNR)
- Initiate the scanning phase
- The 802.11 handoff procedure
    - Scanning
    - Authentication
    - Association

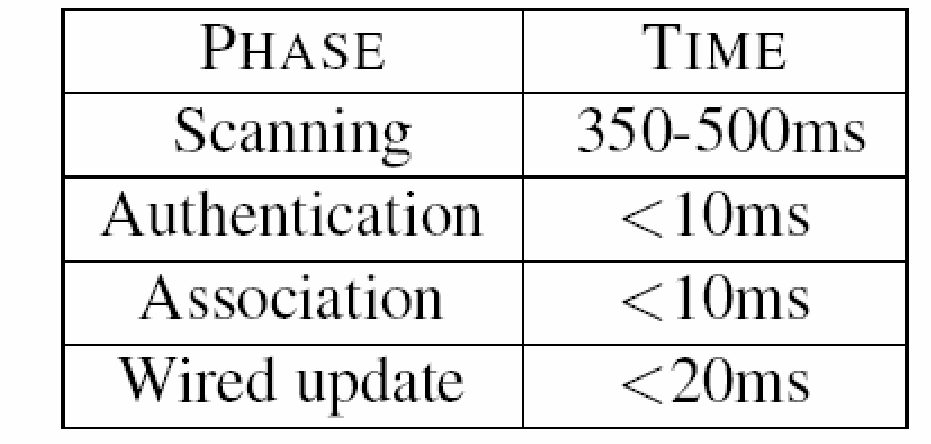


Introduction

- The 802.11 does not provide a shared control channel
- The client must explicitly “scan” each channel for potential access points
- The client switches to a candidate channel and listens for periodic beacon packets generated by access point
- $ScanDelay = NumChannels \cdot MaxBeaconInterval$

Introduction

- Slow to handoff
- Produce “gaps” in connectivity
- The scanning phase completely dominates the cost of handoff - usually contributing more than 90 percent of the overhead



PHASE	TIME
Scanning	350-500ms
Authentication	<10ms
Association	<10ms
Wired update	<20ms

Introduction

- The delay incurred in completing an active scan is regulated by some parameters:
  - ProbeDelay
  - Media contention time
  - MinChannelTime
  - MaxChannelTime
- $$\sum_{c=1}^{c=NumChannels} (1-P(c))\cdot Min + P(c)\cdot Max$$
- P(c) is the probability of one or more access point operating in channel C

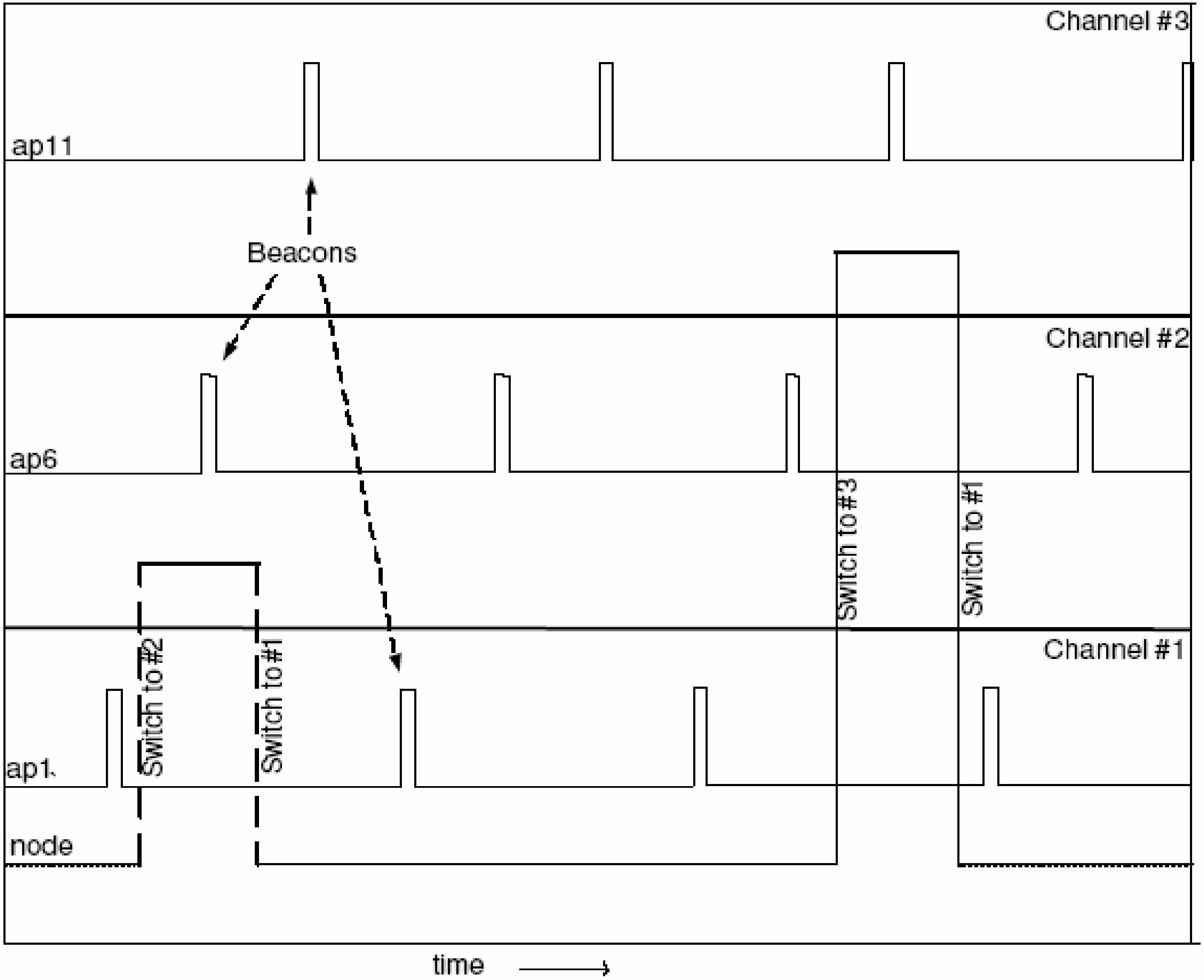
Introduction

- How to continuously monitor the proximity of nearby 802.11 access points?
- The potential disruption of channel switching is minimized by synchronizing short listening periods at client with regular periodic transmission from each access point

SyncScan

- Basic Algorithm
  - All access points operation on channel 1 will broadcast beacons at time t
  - While access points on channel 2 will do the same at time t+d ms
  - Channel 3 access points will send at time t+2d
- A mobile client associated with an access point operating on channel C can detect APs operation on channel C+x at time t+xd

SyncScan


Channel #3
ap11
Beacons
Channel #2
ap6
Switch to #3
Switch to #1
Channel #1
Switch to #2
Switch to #1
ap1
node
time

SyncScan

- SyncScan does not add several complexities
- The accuracy of clock in access points is critical to the global synchronization of beacon timing
- The simplest approach is to leverage the wide availability of Network Time Protocol (NTP) over the internet
- The most enterprise access point product already support

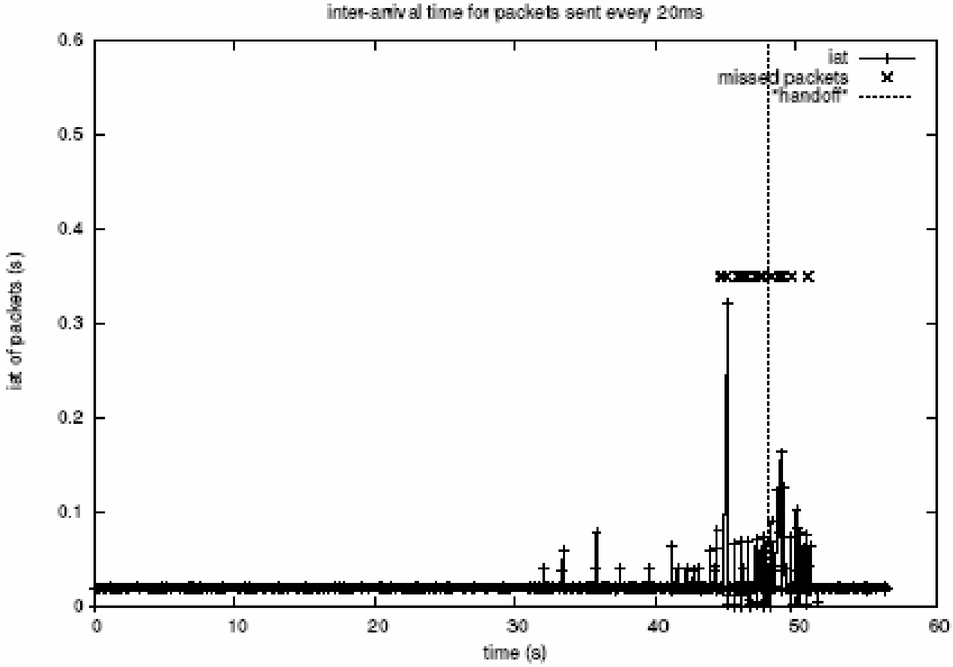
SyncScan

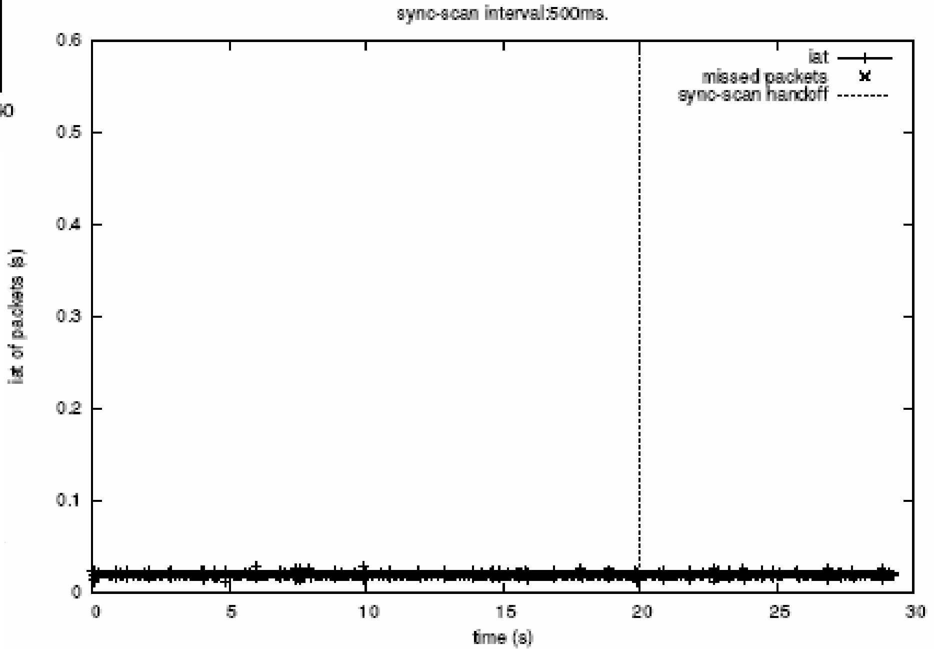
- The SyncScan procedure has a hidden cost
- It removes overhead of the scanning phase, replace it with a regular overhead
- For each session, the client must switch channels, wait for any beacon and then switch back
    - $\text{SyncScanDelay}=2 \cdot \text{SwitchTime}+\text{WaitTime}$

SyncScan

- A continuous scanning can discover the presence of access point with stronger SNRs even before the associated access point’s signal has degraded below its threshold
- SyncScan also provides an opportunity for continuous location tracking

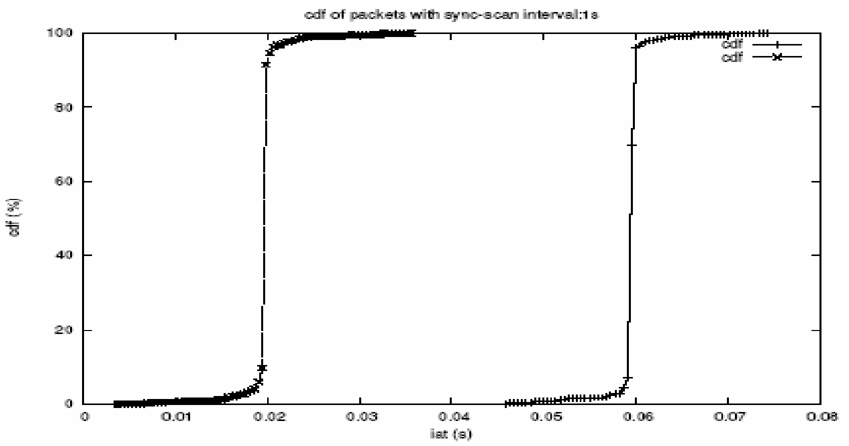
Implementation and Evaluation

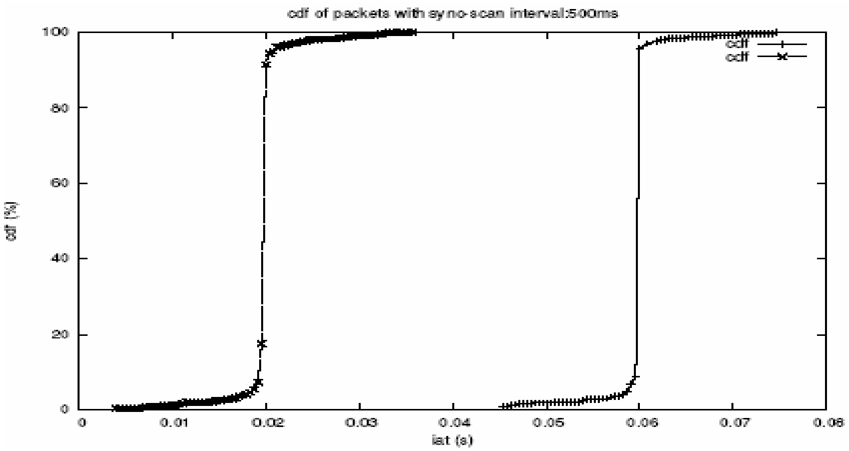


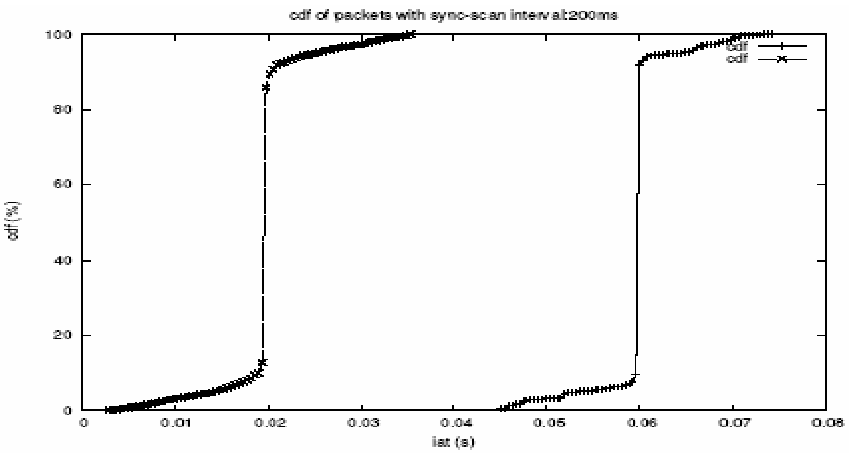


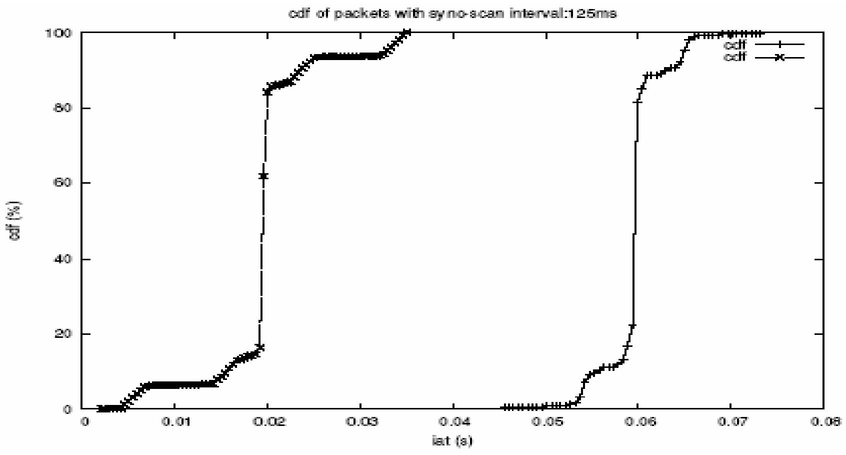
ChihJen

Implementation and Evaluation

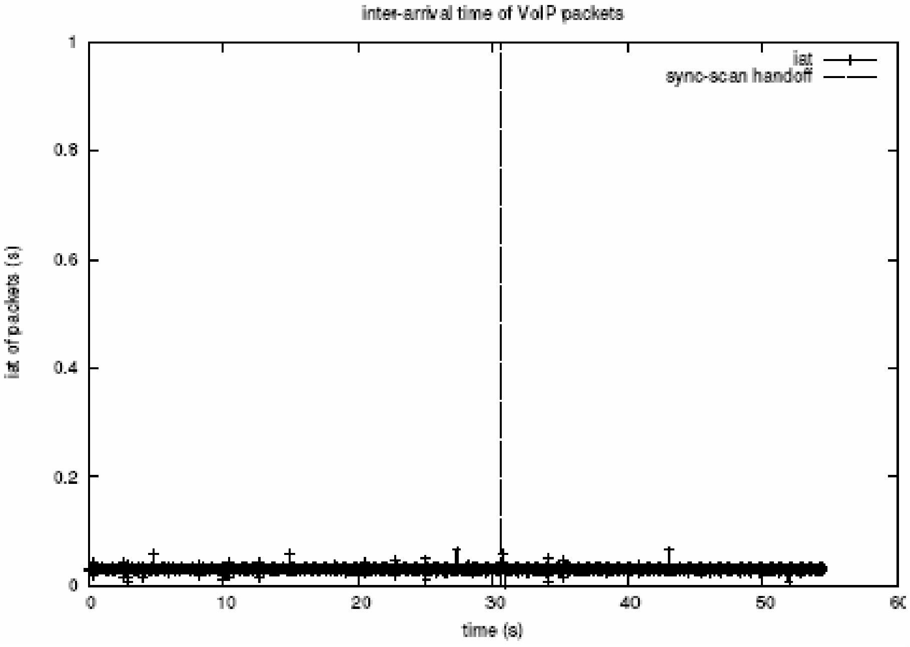


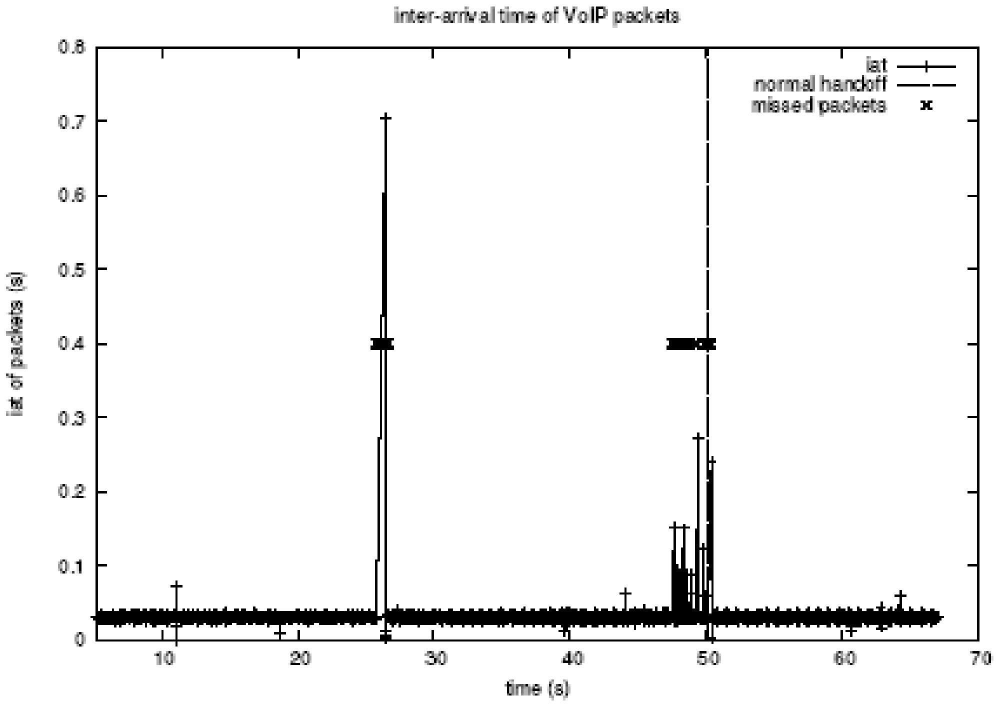




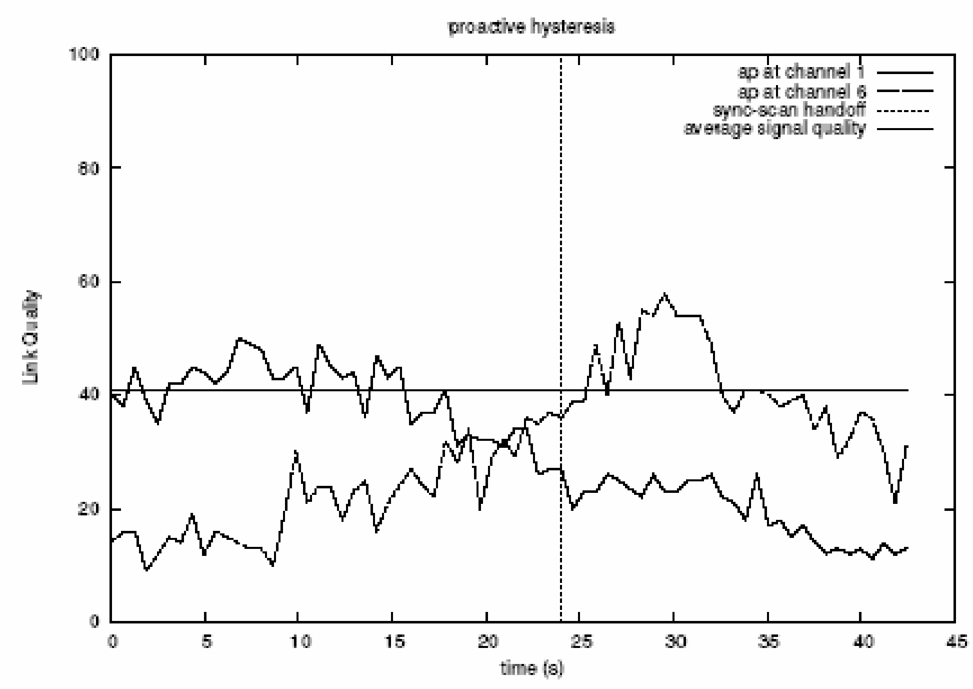


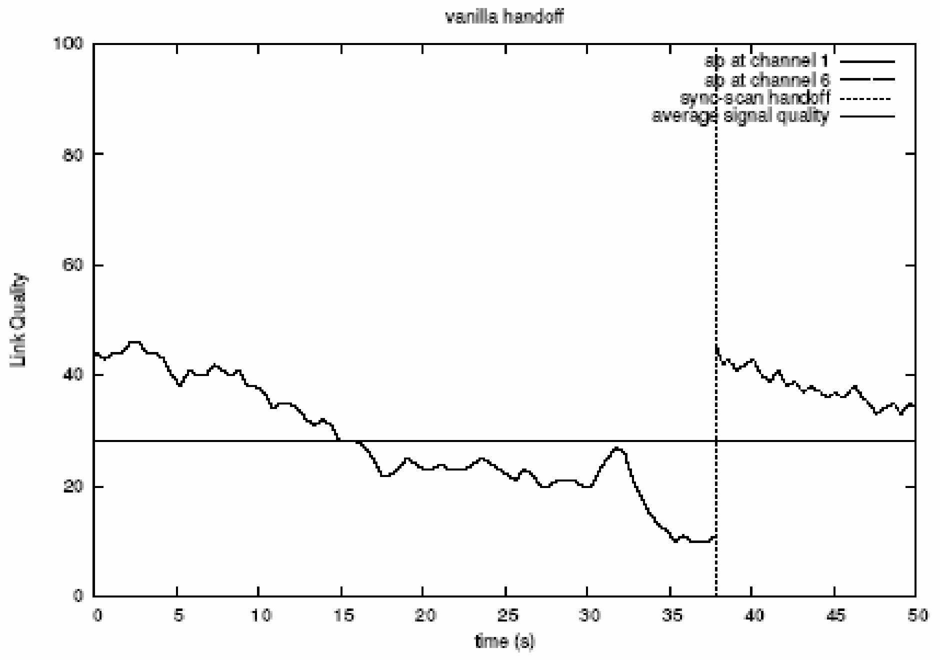
Implementation and Evaluation



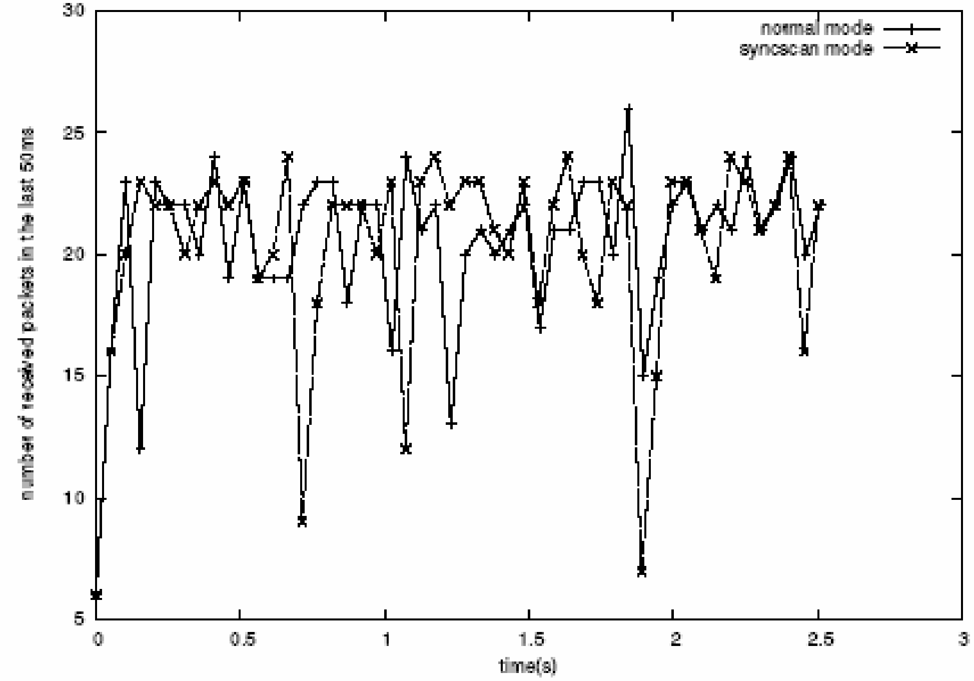


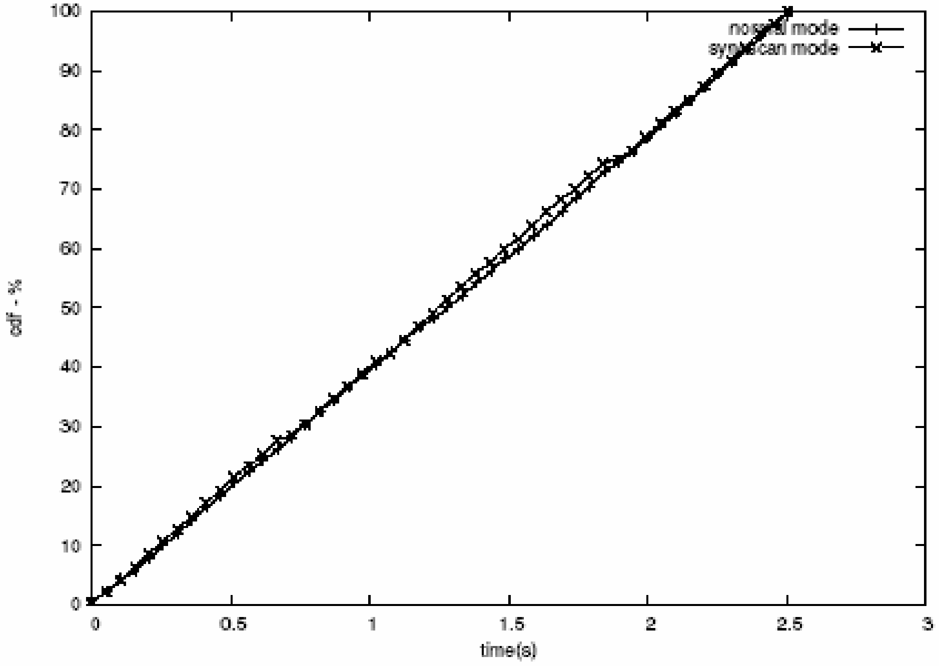
Implementation and Evaluation





Implementation and Evaluation





Conclusion

- Mobile voice application are the next challenge for 802.11-based wireless networks
- By synchronizing the announcement of beacon packets, a client can arrange to listen to other channel with very low overhead