# Seamless Mobility on OpenFlow-enabled Networks

**Intermediate Project Presentation**

Nikhil Handigol & Wei Wei

October 29, 2008

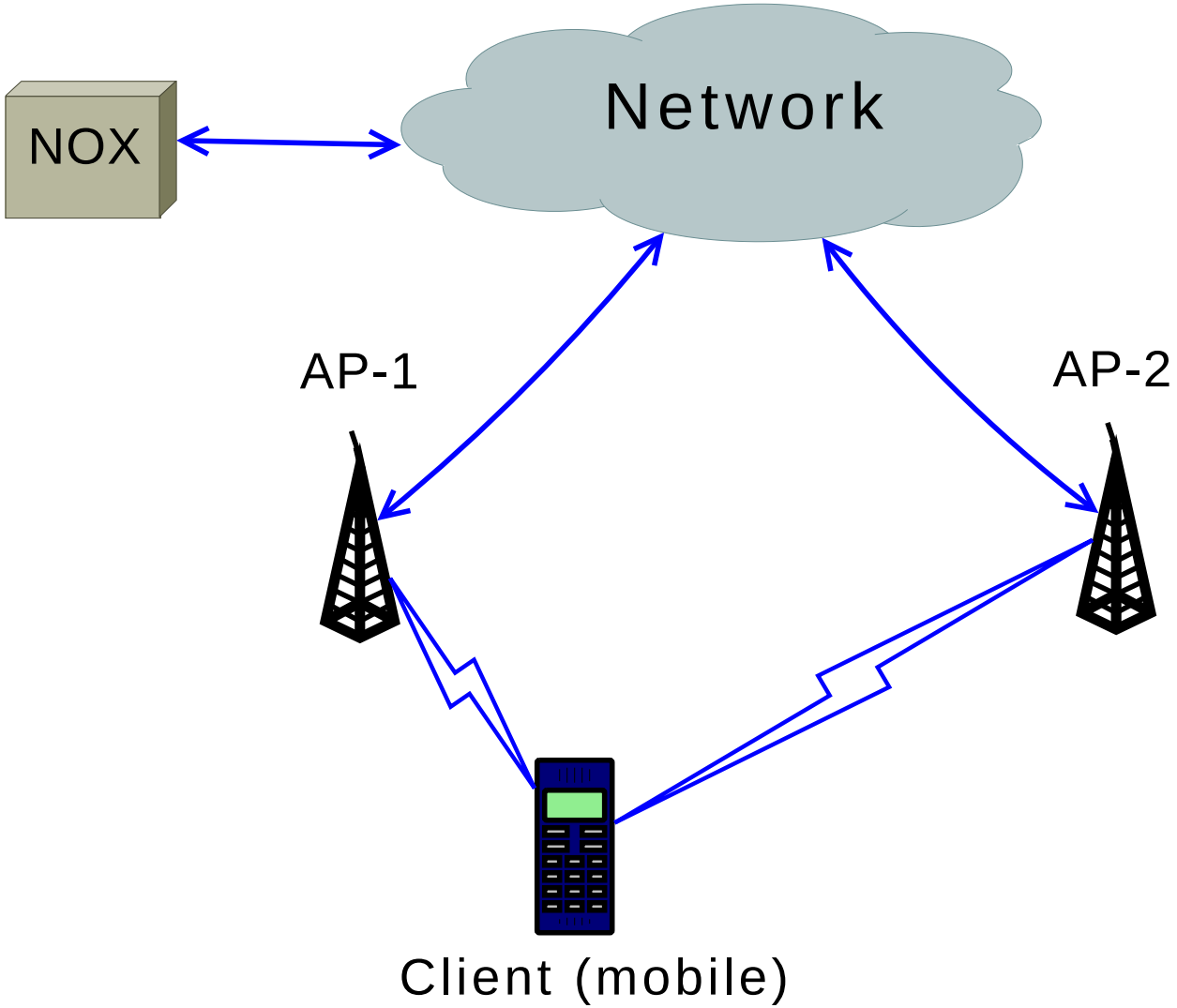
Outline

# Goals

- **Seamless Mobility**
  - Exploit two radios on the client to minimize handoff latency
  - Completely oblivious to transport and application layers
- **Lossless Handoff**
  - No undue loss of packets
- **No Performance Degradation**
  - No side-effects such as packet-reodering, high latency, etc.
- **User-determined Policy**
  - Client gets to achieve handoff based on his/her own policies

# Design Overview





# Fundamental Questions

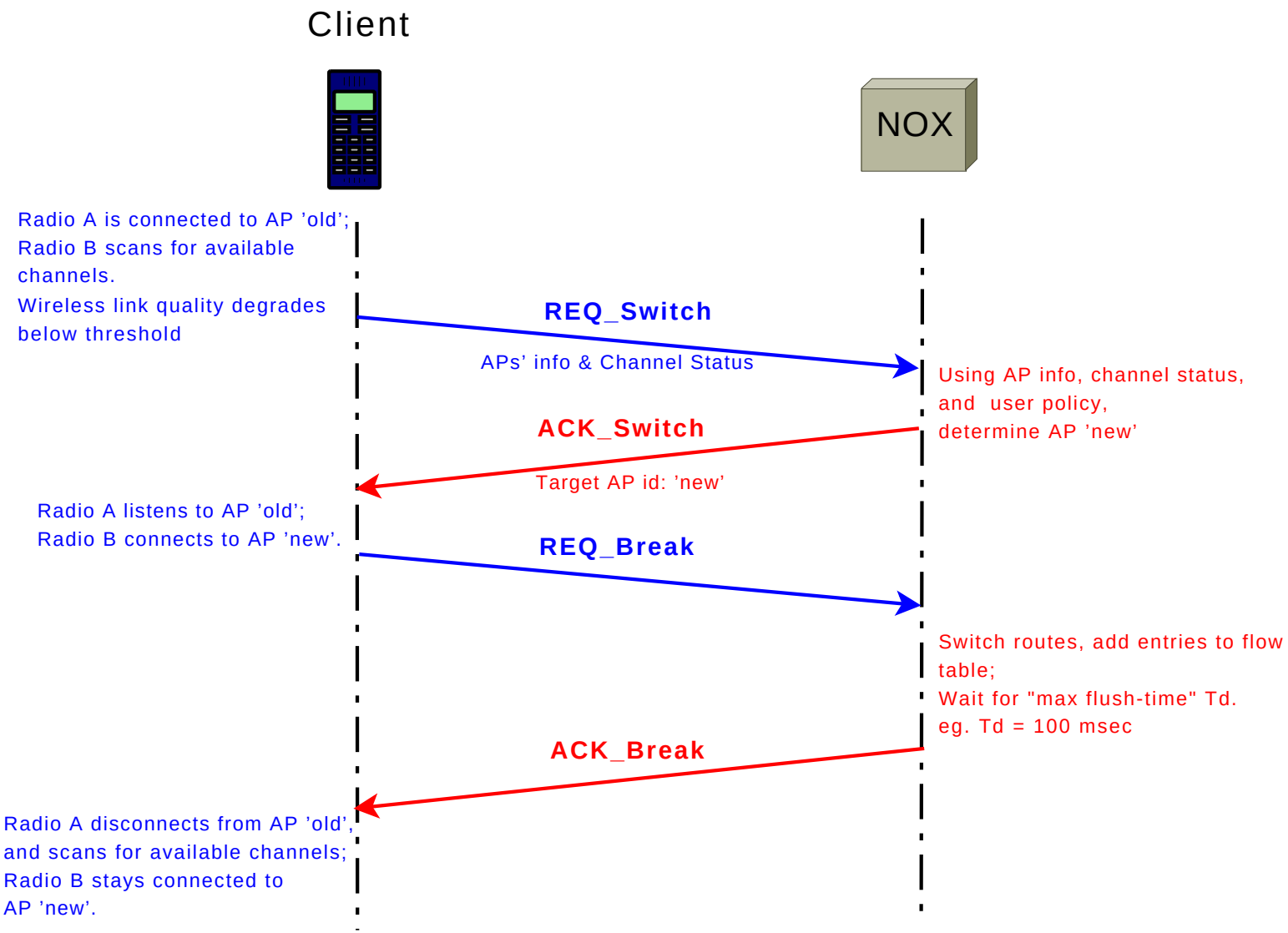


- **When** to switch APs?
  - Eg. degrading signal strength
- **Where** to switch to?
  - Eg. random, max-strength, max increase in strength, etc.
- **How** to switch?
  - Exploit two radios on the client - make-before-break [1]
  - Lossless handoff
  - Efficiency through packet-buffering
- **Who** makes the decisions?
  - Client + NOX (NOX runs policies decided by the user)
  - Client - "when", "how"
  - NOX - "where", "how"

[1] "Eliminating handoff latencies in 802.11 WLANs using Multiple Radios: Applications, Experience, and Evaluation" - Vladimir Brik, Arunesh Mishra, Suman Banerjee

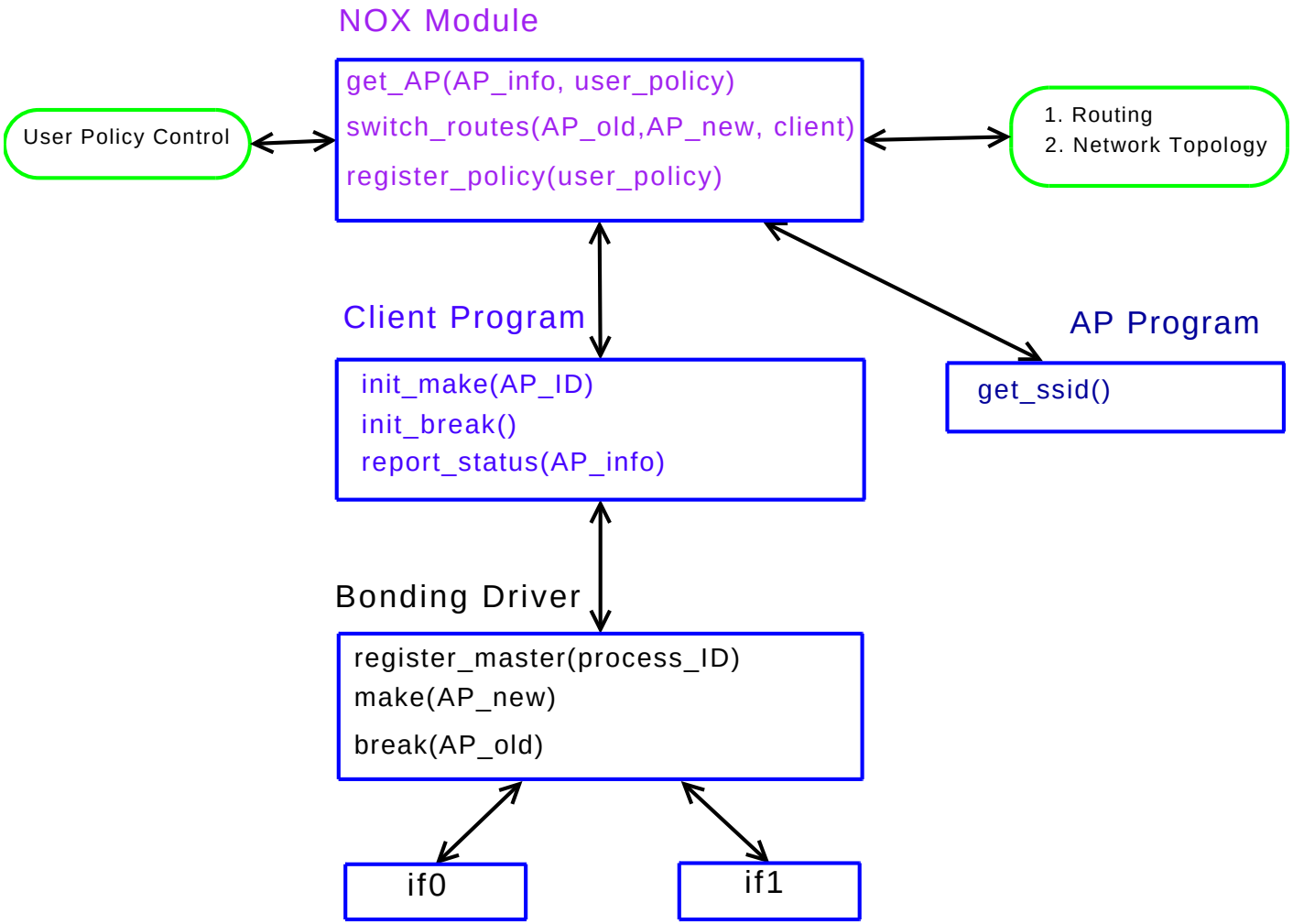
# Protocol




Client
NOX
Radio A is connected to AP 'old';
Radio B scans for available channels.
Wireless link quality degrades below threshold
REQ_Switch
APs' info & Channel Status
Using AP info, channel status, and user policy, determine AP 'new'
ACK_Switch
Target AP id: 'new'
Radio A listens to AP 'old';
Radio B connects to AP 'new'.
REQ_Break
Switch routes, add entries to flow table;
Wait for "max flush-time" Td.
eg. Td = 100 msec
ACK_Break
Radio A disconnects from AP 'old', and scans for available channels;
Radio B stays connected to AP 'new'.

# Software Architecture





# Deliverables



- **NOX module**
  - Core network functionalities - choose AP, switch routes, etc.
- **Client-side functionality**
  - **Bonding driver** - modified to provide handoff specific functionality
  - **Client program** - the master program, that communicates with both NOX and the bonding driver
- **AP program**
  - Reports SSID and other information to the NOX module
- **Performance evaluation** in the simulation environment
  - Throughput, buffer space, packet-loss, etc.
- **Deployment** (hopefully)

# Approximate Timeline



- November 15: Bonding Driver (Nikhil)
- November 20: NOX module (Wei)
- November 25: Client Program (Nikhil), AP Program (Wei)
- November 30: Debugging and testing in the simulation environment (Nikhil and Wei)
- December 10: Deployment attempt (Nikhil and Wei)

# Thank you!