Incentive Engineering in Wireless LAN Based Access Networks

Raymond R.-F. Liao
Rita H. Wouhaybi
Andrew T. Campbell

October 28, 2002

802.11 Explosive Growth

Introduction

- Explosive growth of 802.11 users and access points
- The need to differentiate traffic based on
  - Type of traffic
  - User

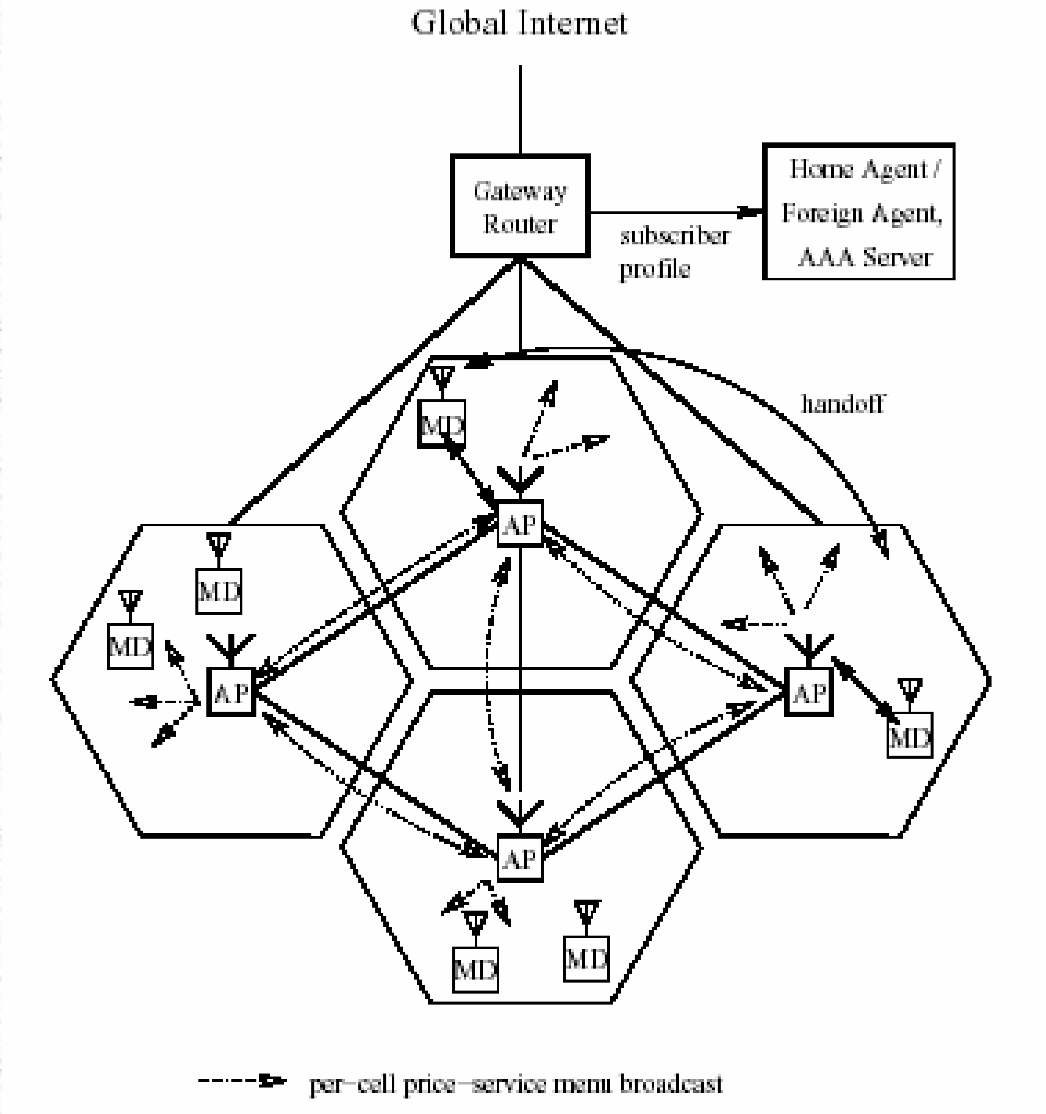
Desired System

- The need for a platform that provides
  - QOS and differentiation of traffic
  - Simple pricing mechanism for users and providers
  - Incentive for users to bid truthfully
  - Distributed Algorithm
  - Minimizing signal information

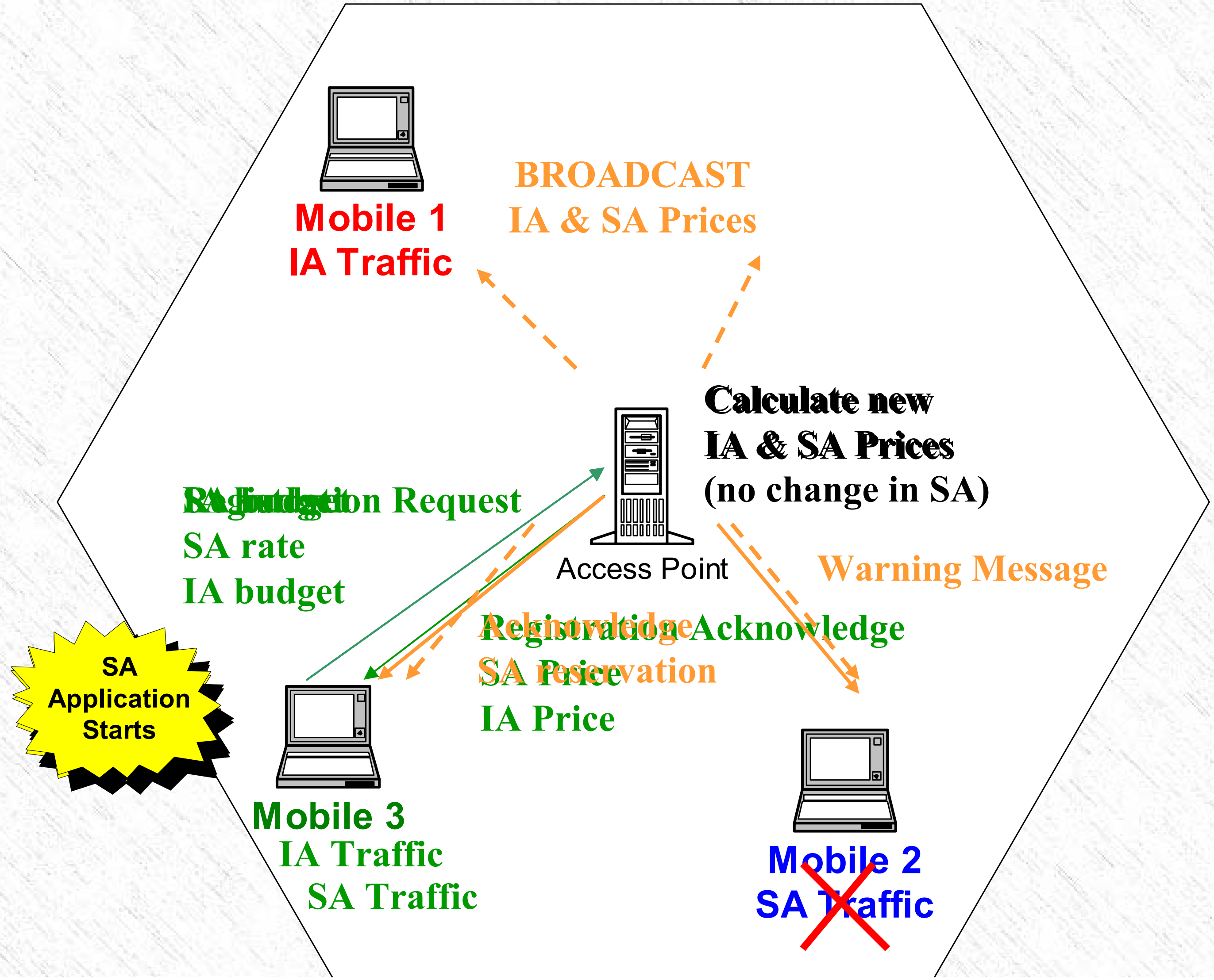
Our Approach

- The system provide 2 levels of service:
  - Instantaneous Allocation (IA)
  - Stable Allocation (SA)
- Users have a budget that they can divide between the 2 services
- The access point periodically
  - calculates the prices of the 2 services based on bandwidth demand and usage
  - broadcasts them to the users

Wireless LAN Network



Example



BROADCAST
IA & SA Prices
Mobile 1
IA Traffic
Calculate new
IA & SA Prices
(no change in SA)
Access Point
Warning Message
SA rate
IA budget
Acknowledge
IA Price
SA
Application
Starts
Mobile 3
IA Traffic
SA Traffic
Mobile 2
SA Traffic

Instantaneous Allocation (IA)

- Nash bargaining fair
  - allocation is proportional to a subscriber's service purchasing power
- Efficient
  - allocation is capped at the maximum consumption level, measured by base station
  - supports bursty data transactions
- Simple Protocol
  - One broadcast message from base station, no messages from mobile
  - Base station detects excessive consumptions via traffic monitoring.

IA Algorithm

$$\upsilon_{i,I} = \upsilon_i + \upsilon_{i,S}$$

υ_i : Service purchasing power of user i

$\upsilon_{i,I}$: Service purchasing power for IA Allocation

$\upsilon_{i,S}$: Service purchasing power for SA Allocation

$$\zeta_{i,I}(t) = \frac{\upsilon_{i,I}}{b_{i,I}^{\max}(t)}$$

$\zeta_{i,I}(t)$: IA unit "bid price" for a mobile device i

$b_{i,I}^{\max}(t)$: maximum bandwidth of IA class that mobile device i can consume

IA Algorithm: Example

	1	**2**	**3**
budget	100	200	100
b_{max}	100	100	100

$$\zeta_{1,I} = \frac{\upsilon_{1,I}}{b_{1,I}^{\max}} = \frac{100}{100} = 1 \quad \zeta_{2,I} = \frac{200}{100} = 2 \quad \zeta_{3,I} = \frac{100}{100} = 1$$

Sort the unit bid price $\Rightarrow (\zeta_{2,I}, \zeta_{1,I}, \zeta_{3,I})$

$$\theta_{all,I} = \sum_i \upsilon_{i,I} = 400 \qquad \theta_{k,I} = \sum_{i \in B(k)} \upsilon_{i,I} \quad \theta_{1,I} = 200 \quad \theta_{2,I} = 300 \quad \theta_{3,I} = 400$$

$$b_{k,I} = \sum_{i \in B(k)} b_{i,I}^{\max} \qquad b_{1,I} = 100 \quad b_{2,I} = 200 \quad b_{3,I} = 300$$

$$q_{k,I} = b_{k,I} + \frac{\theta_{all,I} - \theta_{k,I}}{\zeta_{(k),I}} \quad q_{1,I} = 200 \quad q_{2,I} = 300 \quad q_{3,I} = 300$$

$$q_{l,I} = 250; \;\; q_{l,I} \in (q_{j,I}, q_{j+1,I}] \Rightarrow j = 1 \qquad p_{l,I} = \frac{\theta_{all,I} - \theta_{1,I}}{q_{l,I} - b_{1,I}} = \frac{400 - 200}{250 - 100} = 1.33$$

$$b_{1,I} = \min\left(\frac{\upsilon_{1,I}}{p_{l,I}}, b_{i,I}^{\max}\right) = \min\left(\frac{100}{1.33}, 100\right) = 75$$

$$b_{2,I} = \min\left(\frac{200}{1.33}, 100\right) = 100 \quad b_{3,I} = \min\left(\frac{100}{1.33}, 100\right) = 75$$

IA Algorithm: Measurement-based

$$b_{i,I}^{\max}(t_n) = \min\left\{\gamma \overline{b_{i,I}}(t_n), b_{i,I}^{\max}\right\}$$

$$\gamma(t_n) = \begin{cases} \gamma(t_{n-1})(1+inc) & \rho > threshold \ \& \ \exists j; b_{j,I}^{\max}(t) < b_{j,I}^{\max} \\ \max\left\{1, \gamma(t_{n-1})(1-dec)\right\} & \rho < 0.7 * threshold \\ \gamma(t_{n-1}) & otherwise \end{cases}$$

Stable Allocation (SA)

- Base station maintains a ranked allocation list prioritized by bid price
  - Subscribers with higher bid price has more stable allocation
  - Supports handoff by broadcasting handoff price
- Incentive compatible
  - Users have no incentive to cheat in declaring more or less bandwidth than needed;
  - Dominant strategy for users prefer more throughput is to use IA allocation;
  - Dominant strategy for users prefer allocation stability will choose SA allocation.
- Simple Protocol
  - Reservation protocol: mobile submits bid for bandwidth, no needs to estimate reservation length
  - Reservation revocation is delayed for a warning interval for applications to respond with increasing bid price or reducing bid quantity

SA Algorithm

admission control { // arrival of reservation request

update $\overline{\lambda}(t)$

if $(b_{i,S} > (1-\rho_{l,S})C)$

reject(); // no enough bandwidth

elseif $(\zeta_{i,S} < p_{l,S})$

reject(); // bid price too low

else

Calculate_SA_price();

}

SA Algorithm

calculate_SA_price {

update $(1-\rho_{l,S})C$;

update t_{out};

search for the smallest p_k in the quantized price set $\{p_k\}$

$$\gamma t_{out} \sum_{i \geq k} \bar{\lambda}(t \mid p_i) \leq (1-\rho_{l,S})C;$$

$$p_k \geq \frac{\theta_{all,I}}{\rho_{l,S} C - \gamma t_{out} \sum_{i \geq k} \bar{\lambda}(t \mid p_i)};$$

$p_{l,S} = p_k$;

broadcast price-service menu;

}

SA Algorithm

$$\overline{\lambda}(t \mid p_k) = \alpha \frac{cnt\_b_k}{t - t_{n-1}} + (1-\alpha)\overline{\lambda}(t_{n-1} \mid p_k)$$

cnt_b_k: the sum of $b_{i,S}$ arrived within $(t_{n-1}, t]$

whose bid price $\zeta_{i,S} \in [p_k, p_{k+1})$

t_{out}: the minimum interval at the end of which additional SA bandwidth is guaranteed to be available

$$p_i = \begin{cases} \dfrac{p_K}{K-i+1} & i = 1, \ldots, K \\ 0 & i = 0 \\ \infty & i = K+1 \end{cases}$$

SA Algorithm

maintain SA allocation {

maintain sorted list of $\zeta_{i,S}$

if ($\zeta_{i,S}$ < $p_{l,S}$ AND *i* is not under probation)

put *i* under probation, start timeout timer;

if ($\zeta_{i,S} \geq p_{l,S}$ AND *i* is under probation)

move *i* out of probation;

if (*i* is under probation and timer expires)

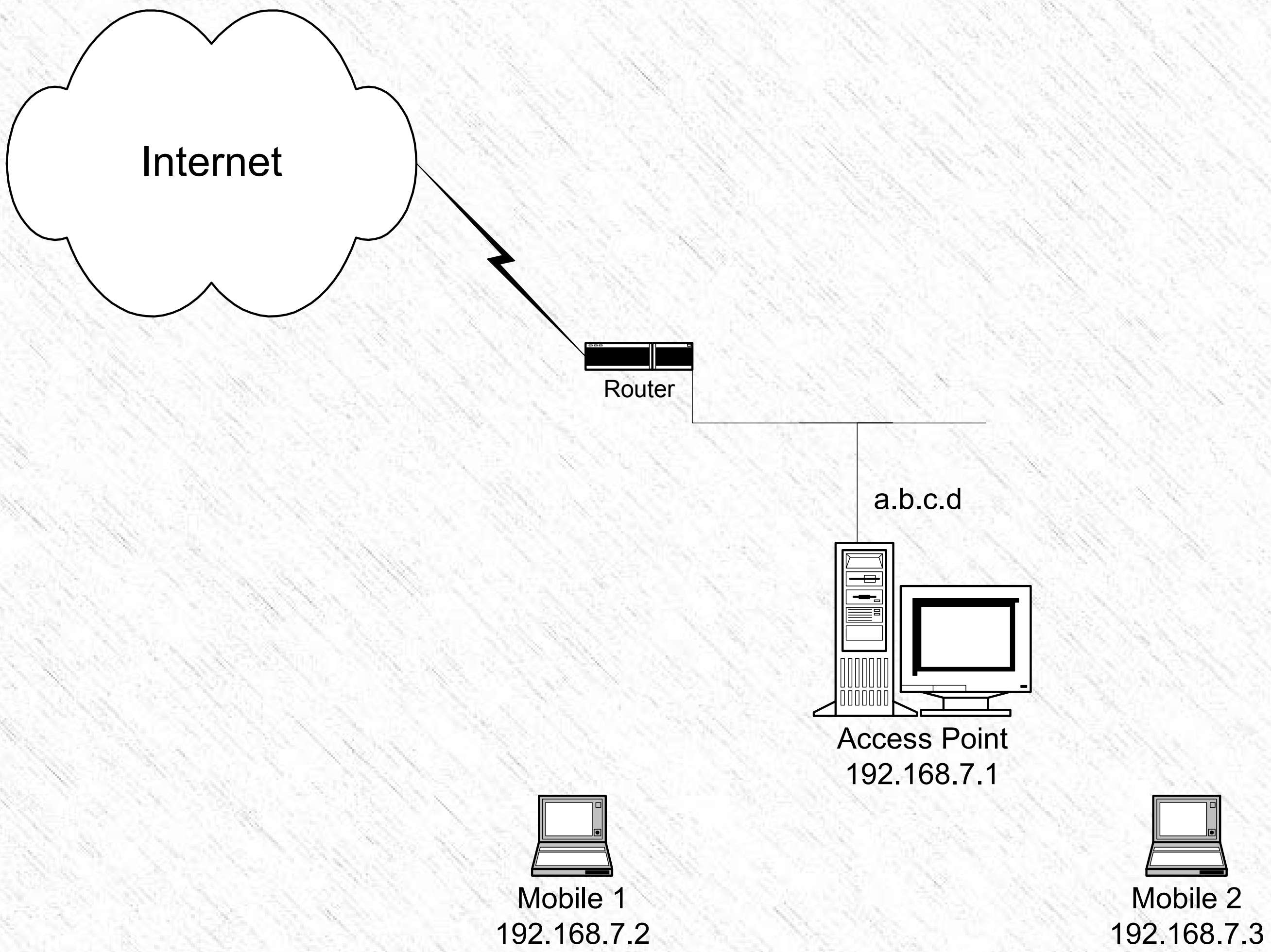
remove *i*'s SA reservation;

}

Testbed

- Modifying wireless driver to allow for traffic snooping
- Relying on TC for traffic shaping
- Implementing 2 modes: simulation and real-time testing

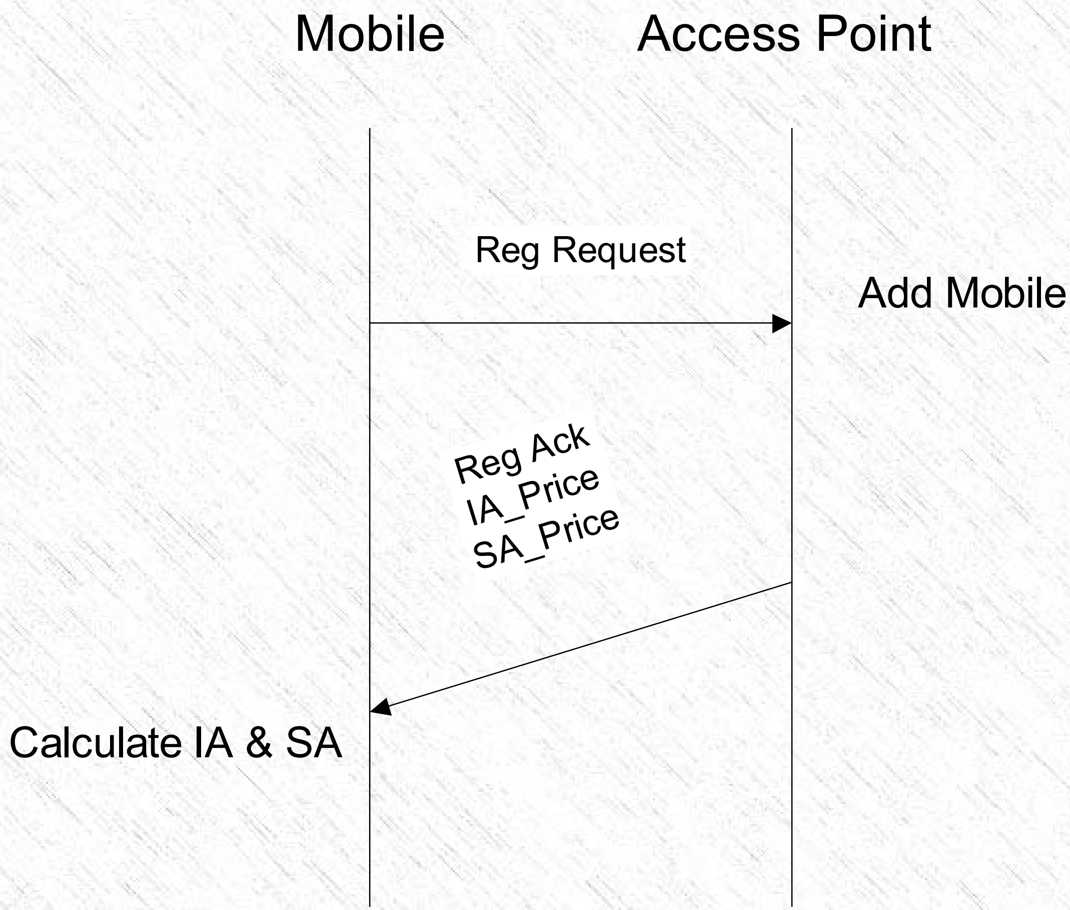
Testbed Setup



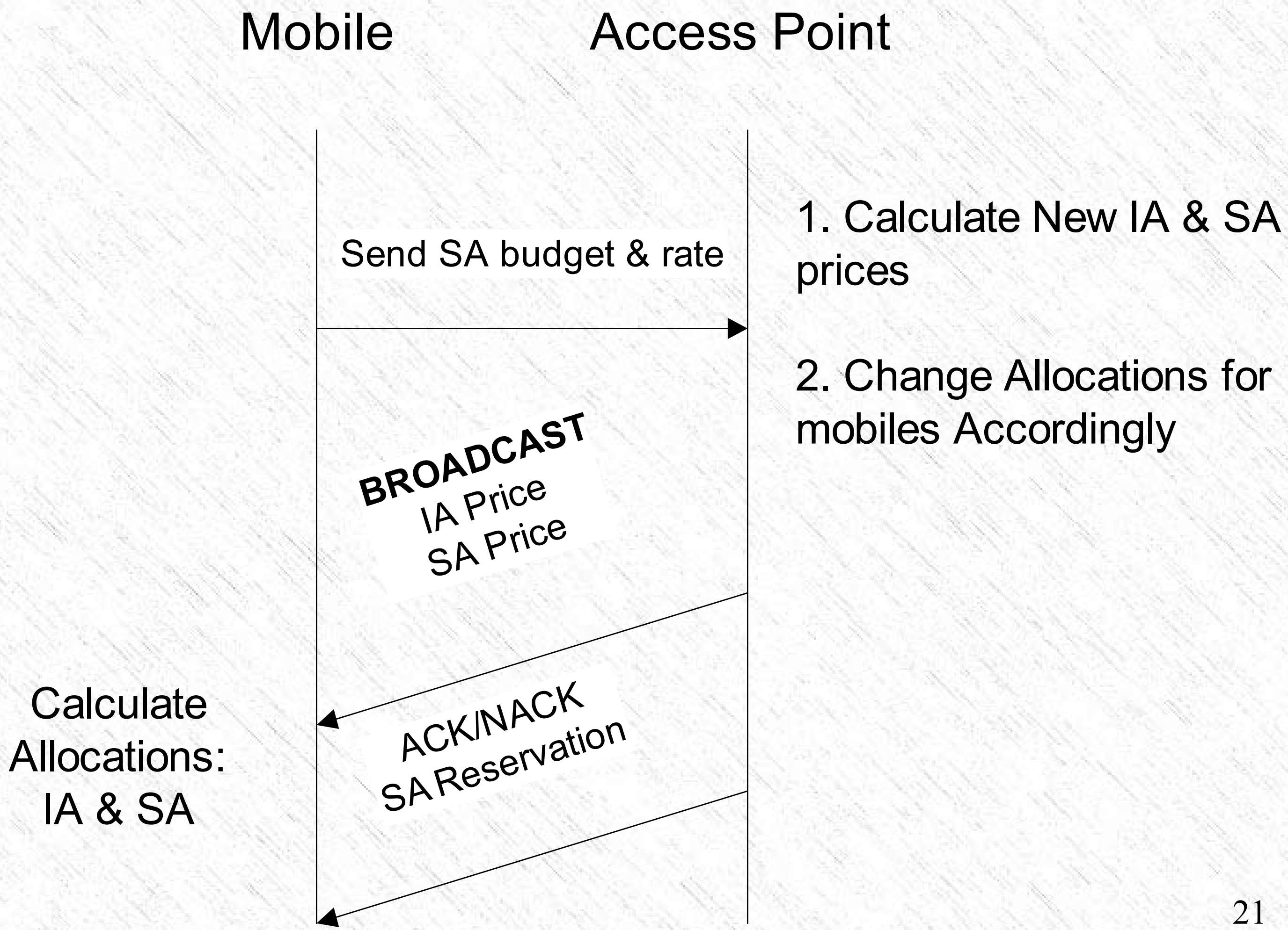
Testbed: Modules

- Access Point:
  - NAT
  - TC
  - Protocol
- Mobile:
  - TC
  - Protocol

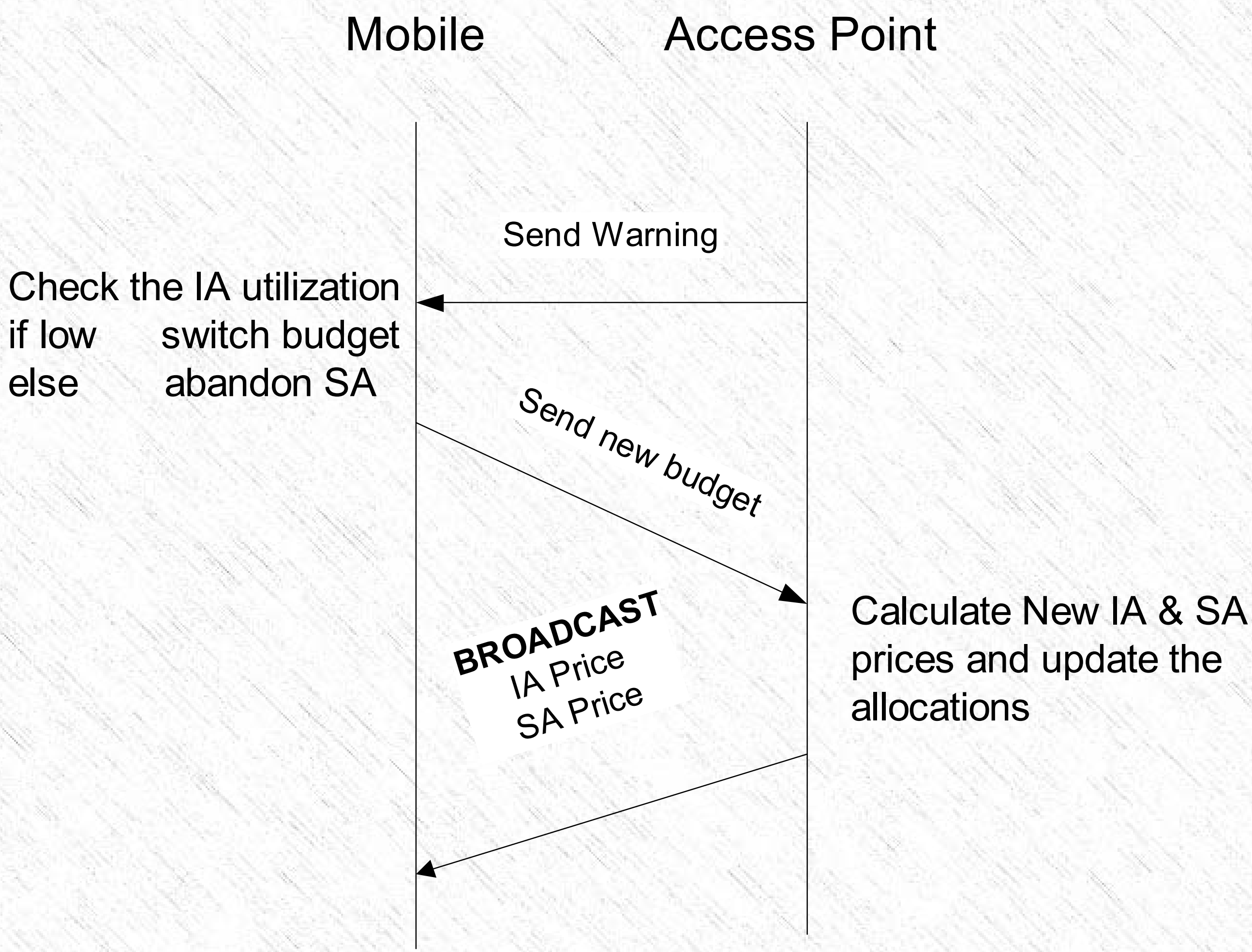
Algorithm: Registration



Algorithm: SA Reservation



Algorithm: SA Warning



Access Point

- Periodically:
  - Measures traffic (updated driver)
  - Calculate new prices
  - Updates the allocations (TC)

- Bandwidth can be switched between IA & SA when underutilized.

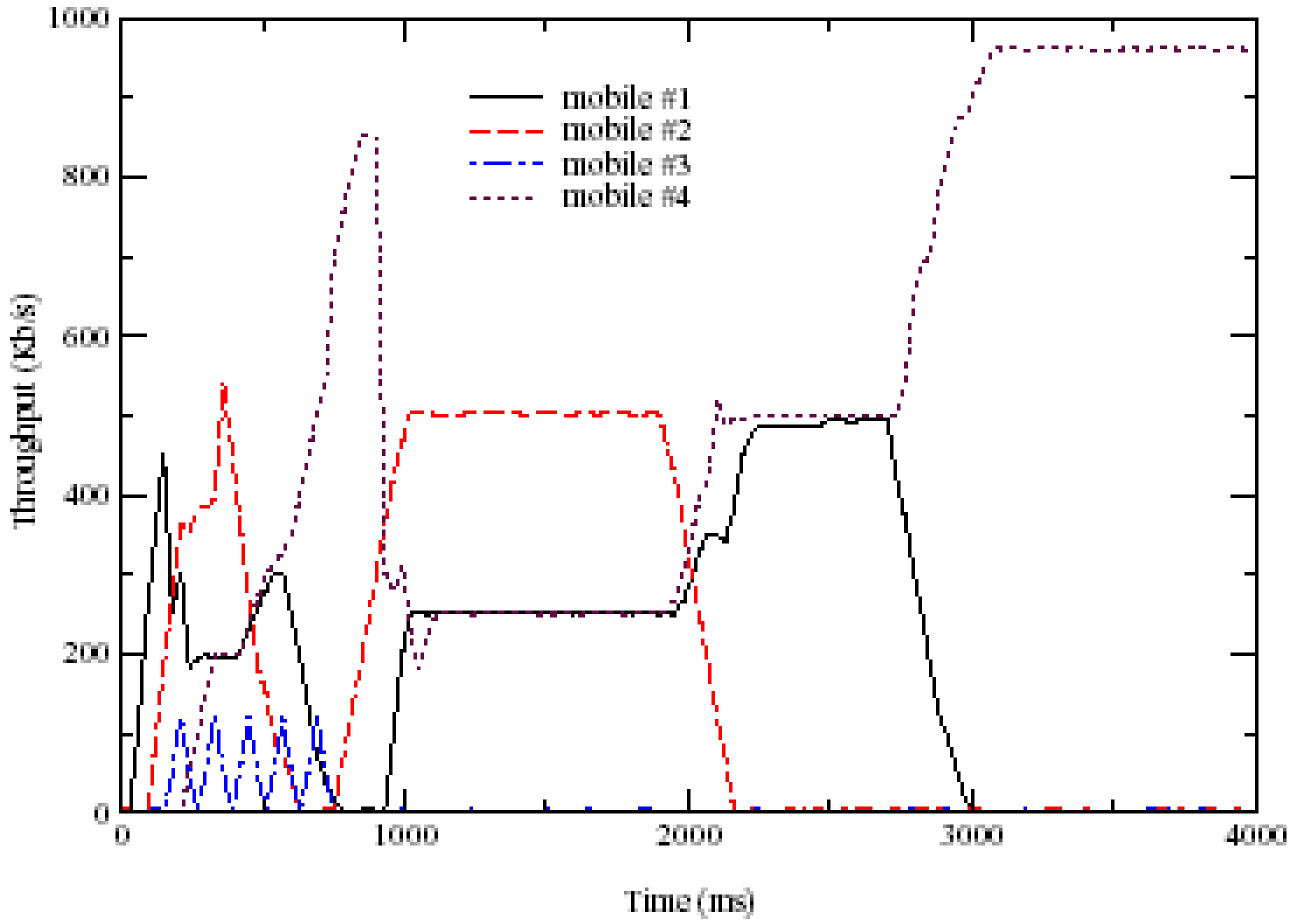
Mobile

- Periodically:
  - Measures traffic (updated driver)
  - Tries to maximize allocations
  - If needed:
    - Requests a new allocation
    - When acknowledged
      - Updates its bandwidth accordingly (TC)

Mobile

- An application starts
  - Traffic is assumed to be IA
  - Port is checked
  - If SA
    - SA budget and rate are updated
    - Request sent to Access Point
    - If Approved
      - Traffic is switched to SA
      - Allocations updated accordingly

Testbed Results



Testbed Results