FreeSurf: Application-centric Wireless Access with SDN

IRTF SDNRG
Prague, July 2015

Zhen Cao, Jürgen Fitschen, Panagiotis Papadimitriou
Leibniz Universität Hannover
Public Wi-Fi with a Traditional setup

Network highly underutilized
Only ONE connection per user per month *
10% active customers

No way to provide ubiquitous connectivity for customers

Operator

Poor user satisfaction
Always reluctant to pay, and impatient
12 clicks are required for a captive portal authentication
25% of users abandon access after 4 seconds #
50% of users abandon access after 10 seconds
**Application-centric Wireless Access**

- **Authentication:** users authenticate to the network using their SP accounts, e.g., Amazon
- **Access:** users are allowed to access the SP domain after a successful authentication
- **Billing:** SP is accountable for their customers' access, and users are left to **FreeSurf**
1. EAP Identity Request

2. EAP Identity Response
   alice@amazon.com

EAP-TTLS Start

Server authentication and TLS tunnel setup

EAP-TTLS Phase 2 (e.g., MSCHAP), user authenticates using the SP account
   (alice@foo.com)

Authentication forwarding table

<table>
<thead>
<tr>
<th>ID</th>
<th>FreeSurf SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>@amazon.com</td>
<td>Amazon AAA IP</td>
</tr>
<tr>
<td>@ebay.com</td>
<td>ebay AAA</td>
</tr>
<tr>
<td>.</td>
<td>Operator AAA</td>
</tr>
</tbody>
</table>

Radius EAP-TTLS Start
FreeSurf Access Control

Authentication

Openflow table

Flows towards SP domain

Bloom Filter

Flows towards outside of SP domain

Tablesize vs. Lookup efficiency

CDN, Multihoming, Cloud

BF assisted
Operation Modes

- **Direct mode:**
  - SPs take over the authentication procedure and provide Internet access to their clients
    - Direct control of clients’ credentials
  - Better suited to large SPs

- **Broker mode:**
  - A broker takes over the client authentication on behalf of multiple SPs
  - Better suited to small SPs
FreeSurf Prototype

Client: an iOS device

**Flow 1: FreeSurf authflow**

1. AuthFlow
2. DataFlow

**Flow 2: non-FreeSurf authflow**

1. AAA Server
2. Operator AAA Server

FreeSurf AP: Linux Laptop + USB Wireless Dongle + hostapd

In AWS

Within same LAN

FreeSurf: Application-Centric Wireless Access

Zhen Cao
**Minimal increase in authentication delay with FreeSurf**
- 1.7% additional delay with EAP-TTLS
- 2.4% additional delay with EAP-PEAP

**BF promotes lookup efficiency**
- The larger the flow table is, the more BF helps
- Lookup with the BF is constant irrespective of flow table size
Thank you

For more information, checkout our paper and code at https://github.com/freesurf
BF-assisted Access Control in FreeSurf

### Initialization

<table>
<thead>
<tr>
<th>i</th>
<th>B[i]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>K-3</td>
<td>0</td>
</tr>
<tr>
<td>K-2</td>
<td>0</td>
</tr>
<tr>
<td>K-1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Construction

1. $i \leftarrow$ Hash(pkt$_i$.hdr)
2. $i \leftarrow$ Hash(pkt$_i$.hdr)
3. $K-3 \leftarrow 1$
4. $K-2 \leftarrow 0$
5. $K-1 \leftarrow 0$

### Lookup & Decision

1. $m \leftarrow$ Hash(pkt$_m$.hdr)
2. $k \leftarrow$ Hash(pkt$_k$.hdr)
3. Decision:
   - True: Forward
   - False: Drop

**Complexity:** Size $K$ irrespective of the number of flows; Lookup $O($number of hash functions$)$

*False positive can be tuned by use of larger bit vector and more hash computations*
BF False Positives

- **No false negatives:**
  - Discarded packets are absolutely not permitted

- **Existence of false positives:**
  - Impact: allowing some traffic from authenticated users to unallowed addresses
  - Rescue: twisting the false positive rate as small as possible by a choice of larger bit vector and use of more hash functions