Introduction and Motivations

The Internet is evolving from the Network-Centric view to the User-Centric view. The user increasingly takes an active role in the network, promoting peer-to-peer (P2P) and many-to-many interactions, sharing his wide-band connection and providing both contents and network functionalities. The transition to the User-Centric Internet (UCI) is fostering the development of multi-channel applications. Such applications provide a single interface to perform heterogeneous activities exploiting many communication channels.

Characterizing multi-channel applications has implications in many networking fields:
- Capacity planning and provisioning
- Traffic engineering
- Fault diagnosis
- Policy enforcement
- Intrusion and anomaly detection
- Billing
- Network neutrality

Traditional network traffic analysis methodologies are less and less effective [1]
- ignore relations among nodes and among communication channels
- cannot usually deal with obfuscation, encapsulation and encryption

Therefore, it is necessary to find new techniques and analysis methodologies purposely designed for the properties of emerging applications.

The Proposed Methodology

We propose the definition of a novel methodology for the characterization of multi-channel applications based on a multi-layer traffic inspection and a decomposition approach, as depicted in figure, counting four layers:
- **Host** - aggregates the whole traffic pertaining to a single host
- **Service** - groups together packets having the same transport protocol and IP address-port pair.
- **Biflow** - aggregates packets having the same 5-tuple, where source and destination can be swapped
- **Packet** - looks at the properties of each packet (e.g. size, inter-packet time, payload, ...)

Combining information collected at these layers can reveal useful patterns otherwise not visible. For instance:
- by looking at many biflows belonging to the same application it is possible to detect the application itself
- being aware of an application running on a particular host/service can help in associating a new flow to it, and to identify the related activity

Experimental Analysis: a Proof of Concept

To validate our methodology we applied it to Skype
- multi-channel application
- widespread
- hybrid super-peer based P2P architecture
- intensive use of encryption (AES + RSA)
- proprietary protocols and algorithms

We used TIE [2] to gain knowledge of the traffic associated to each Skype communication channel (see Tab. 1), and we discovered several patterns at different layers

### Table 1: Skype traffic at biflow layer.

<table>
<thead>
<tr>
<th>Activity</th>
<th>proto</th>
<th>src port</th>
<th>dst port</th>
<th>up fast</th>
<th>down port</th>
<th>up bytes</th>
<th>down bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super-peer signaling</td>
<td>udp</td>
<td>33837</td>
<td>20159</td>
<td>2</td>
<td>1</td>
<td>21</td>
<td>29</td>
</tr>
<tr>
<td>Normal video signaling</td>
<td>udp</td>
<td>5000</td>
<td>33837</td>
<td>104</td>
<td>1</td>
<td>91</td>
<td>135</td>
</tr>
<tr>
<td>File transfer</td>
<td>udp</td>
<td>13524</td>
<td>33837</td>
<td>243</td>
<td>247</td>
<td>6 k</td>
<td>123 k</td>
</tr>
<tr>
<td>Call</td>
<td>udp</td>
<td>55879</td>
<td>15024</td>
<td>3 k</td>
<td>4 k</td>
<td>403 k</td>
<td>484 k</td>
</tr>
</tbody>
</table>

**Host layer**
- port numbers are used more than once on a short period
  - listening services (i.e. port 33837)

**Service layer**
- same TCP and UDP listening port number
  - detect also TCP communications

**Biflow layer**
- **signaling traffic** is mostly composed by many short UDP biflows revealing a few different patterns
- **file transfers** present almost the same number of packets in both directions, but most bytes fall only in one of them
- **voice calls** reveal a symmetric pattern in transferred data
  - patterns peculiar to activities

**Packet layer**
- activities reveal peculiar payload-sizes distribution patterns
- more patterns peculiar to activities

Combining the previous observations allows to identify Skype and its activities. For instance, once inferred Skype random port number, we were able to detect different communication channels involved in the same activity.

References
