A Native Measurement Technique for IPv6-based Networks

Francisco J. Garcia, Robert Gardner & Dimitrios Pezaros
AGILENT LABS

Contents

• Why IPv6 ?
• A Native Approach
• Functional Prototype
• Experimental Results
• Next Steps and Concluding Remarks
Why IPv6?

Address Shortage

- Success of the Internet being stifled by address shortage
  - ALWAYS CONNECTED phones, PDAs, cellular/mobile Systems, sensors of the future will need fixed, constantly available address(es)
  - Asia, Europe and Africa currently critically short of addresses
  - Class A,B,C system -> uneven distribution of addresses. CIDR not perfect.
  - IPv4 Routing tables are exploding resulting in loss of performance.
  - NAT extends the life of IPv4 but has serious drawbacks
Case

Mobile Communications

• 3GPP Rel. 5: Architecture for All-IP network
  • Radio Access Network: EDGE, UMTS
  • IP Multimedia Subsystem (IMS)
  • GPRS Core Network
  • Mandates use of IPv6 at Application Layer because of address shortage
• 3GPP Rel. 6: Inter-working with WLAN
• Adopting IPv6 throughout:
  • Reduce infrastructure costs
  • Native mobility support (Mobile IP)
  • Native security support (IPSec)

• All-IP Wireless Network characteristics
  • IP-based multimedia services
  • IP-based transport
  • Integration with IETF protocols for functions such as:
    • Wide area mobility (Mobile IP)
    • Signalling (SIP, SCTP)
    • Authentication, authorisation and accounting (Diameter)
  • Network meeting these characteristics is referred to as an all-IP network

Case

Wi-Fi and WiMAX

• Carrier class solutions presently poor in areas such as:
  • Manageability
  • Security
  • Mobility
• Carrier class and low cost could be facilitated through the adoption of IPv6 features:
  • Connectivity – Stateless node discovery
  • Native security support
  • Native mobility support

Emerging Wireless Standards

WAN
802.20
< 20km

MAN
802.16
< 10km

LAN
802.11
< 100m

PAN
Bluetooth
< 10m

Wi-Fi
802.11a/b/g

WiMAX
802.20
< 20km
Case

Monitoring “Mobile” Services

- When mobile and wireless worlds collide
  - Ubiquitous Internet access with global roaming
  - First commercial deployments of IPv6-based networks
- Devices, users and services are fast becoming mobile
  - Overlay models used to provision and dynamically adapt a delivered service over an existing transport topology
  - Service dissemination topology changes with time
  - Growing trends in Application Level Routing
- Vertical Handovers
  - Access technology changes
    - And so does monitoring infrastructure (e.g. access to Wi-Fi sensors)
    - How can we maintain same or similar service assurance functionality?
- Exploit IPv6 enhancements to natively introduce telemetry functionality

Approach

A Native Approach
Approach

Accessional Techniques

- **Issues with Passive Measurements**
  - Scalability – probes, correlation engines, data volume etc.
  - Link monitoring has its limits
    - New service or new link could trigger the need for complex re-engineering of link monitoring hardware
    - Growing demands for high capacity links of 100Gb/s or higher
  - Complexity involved in performing 2-point measurements
  - Challenging to infer end-to-end view
  - Further complicated with introduction of security

- **Issues with Active Measurements**
  - Assume measured performance for synthetic traffic reflects performance of user traffic
  - Measurement traffic itself may be a factor in performance degradation
  - Measurement mechanism tightly coupled with the measurement applications

Approach

Extension Headers

- **IPv6 Extension Headers**
  - Optional information encoded in separate headers between IPv6 header and upper-layer header.
  - Packet may carry zero or more EHs.
  - So far there are only a handful of standardised EHs:
    - Hop-by-Hop Options.
    - Routing (Type 0).
    - Fragment.
    - Destination Options.
    - Authentication.
    - Encapsulating Security Payload.
  - With one exception, EHs are only processed at the destination(s).
  - The exception is the Hop-by-Hop Options, processed at every node.
Approach

Extension Headers and Telemetry

- Exploit extension headers to introduce native measurement and management functionality
- Applying these notions to the instrumentation of measurements
  - In-line measurements—piggybacking triggers and measurement data onto real user traffic
  - Lowest level condition-event-action triggers for influencing measurements system behaviour
  - Multipoint measurement technique
- Examples using destination header options
  - One-way loss
  - One-way delay

TLV-encoded Options

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Option Length</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Destination Options Header

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Hdr Ext Len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Options

Timestamp Destination Options TLV

<table>
<thead>
<tr>
<th>Option type</th>
<th>Option data len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Reserved)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pointer</th>
<th>Overflow</th>
<th>Flags</th>
<th>Source timestamp: seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Destination timestamp: seconds |
|                                |
|                  |

| Destination timestamp: microseconds |
|                                  |
|                                |

Loss Destination Options TLV

<table>
<thead>
<tr>
<th>Option type</th>
<th>Option data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sequence Number

Approach

Applicability

- A native approach for introducing service measurements
  - Not intended as a replacement for active/passive techniques
  - Complementary technique, when to use depends on service characteristics being monitored (e.g. mobile services)
- Facilitates seamless and incremental deployment
  - Distribute telemetry intelligence to location where it is required, when it is required
  - Allows for the engineering of distributed monitoring solutions that dynamically adapt to the monitored service
## Approach

### Comparative Analysis

<table>
<thead>
<tr>
<th>Aspect/Property</th>
<th>Active Measurements</th>
<th>Passive Measurements</th>
<th>Inline Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impact on network (Measurement process)</td>
<td>Intrusive: Intrusion detection (which competes for resources)</td>
<td>Passive: No impact on network</td>
<td>Intrusive: Marginal load increase and minor delay might be incurred</td>
</tr>
<tr>
<td>Impact on network (Measurement data)</td>
<td>Load generated at one end point</td>
<td>Load generated at one or both ends</td>
<td>Load generated at one end point</td>
</tr>
<tr>
<td>Confidence</td>
<td>Probability of injected traffic used to infer/predict experience of real traffic</td>
<td>Measures real user traffic</td>
<td>Measures real user traffic (Possibility that instrumented traffic is distinguishable and treated differently)</td>
</tr>
<tr>
<td>Controlability</td>
<td>Can test any traffic, path, protocol, etc. – at any time.</td>
<td>Can only measure available traffic</td>
<td>Can only measure available traffic (Requires an accommodating protocol)</td>
</tr>
<tr>
<td>Security/Privacy issues</td>
<td>Private, rejected traffic</td>
<td>Real data not examined</td>
<td>Observation and modification of real traffic</td>
</tr>
<tr>
<td>Scalability issues</td>
<td>Can be dynamically deployed on a per interface basis</td>
<td>Can inject a chosen amount of traffic</td>
<td>Can be dynamically deployed on a per node or per interface basis</td>
</tr>
<tr>
<td>Complexity and Processing</td>
<td>Correlation not required</td>
<td>Correlation of large quantities of data from ingress and egress is computationally intensive and doesn’t scale well</td>
<td>No correlation</td>
</tr>
<tr>
<td>Major application areas</td>
<td>Two-point measurements: Quality of Service testing, such as available bandwidth, hop delay, and packet loss.</td>
<td>One-point measurements: Packet filtering and counting to obtain traffic type, source/destination, etc.</td>
<td>End-to-end path-based measurements, active troubleshooting, packet loss, delay, routing, packet/flow foot printing.</td>
</tr>
<tr>
<td>Other comments</td>
<td>Eavesdropping not possible</td>
<td>Eavesdropping possible</td>
<td>Eavesdropping possible (not applicable in all traffic types e.g. real-time, max MTU traffic)</td>
</tr>
</tbody>
</table>

## Ubiquitous Measurements

- The need for ubiquitous measurements is ever growing and this is exemplified by current IETF activities:
  - Passive Sampling – standard set of capabilities for network elements to select subsets of packets by statistical and other methods that may assist in baseline measurements, performance measurements, troubleshooting, etc. [http://www.ietf.org/html.charters/psamp-charter.html](http://www.ietf.org/html.charters/psamp-charter.html)
  - The case for an Internet Measurement Protocol – allowing it to be handled by the forwarding path rather than the router CPU. [http://www.irtf.org/charters/imrg.html](http://www.irtf.org/charters/imrg.html)
  - IP Flow Information Export – network elements exporting flow information in a standard way so that it can be fed directly into mediation, accounting/billing and network management systems. [http://www.ietf.org/html.charters/ipfix-charter.html](http://www.ietf.org/html.charters/ipfix-charter.html)
- Are these approaches flexible, adaptable and scalable enough to handle mobility?
- **Issues**
  - Standard processes could be lengthy
  - Potential for lots of non-service specific data
  - New service, new set of metrics – potentially long time before the measurements could be deployed
Functional Prototype

Applying Programmable Networking Concepts

- Dynamically programmable network architectures
  - Supporting fast service creation and deployment
  - "network aware applications and application aware networks"
- Applying these concepts to the telemetry of mobile services
- Other lessons could be learnt from peer-to-peer, ubiquitous and pervasive computing
  - Deploy agents to perform specific functionality
  - Exploit user equipment
  - Agent-based approaches facilitate immediate deployment
- Telemetry Agents
  - Lowest level deployable component
  - Remotely managed
  - Dynamically linked into application
- Advantages
  - Transparently introduced
  - Dynamically deployed
  - Engenders flexibility, extendibility and scalability
Test-Beds

- Test-Beds at Lancaster and Agilent Labs are based predominantly on Linux systems (Kernel 2.4.x)
  - Telemetry agents implemented as Loadable Kernel Modules (LKM)
    - Can be linked with a running kernel at run-time
  - Distributed control, security, communications and management frameworks built using Java technologies
    - Including streaming of service data records
  - Test-beds consist of a combination of PCs to act as servers, wireless access points (802.11b) and routers
  - Mobile nodes based on laptops, Sharp Zaurus, and iPAQ
  - Mobile IPv6 used for managing mobility

Prototype General Architecture

- Local Telemetry Agent Management
- Instrumented Node
- Service Data Records
- Distributed Telemetry Agent Management
- OSS Application
- Streamer
- Cache
- Event
- Label for type & instance
- Time Stamp
- Computed data
- OSS Application
- Consumer

Fetch, load, execute, configure, etc.
Adding Telemetry Headers

Prototype

User Space

Kernel Space

From Device

To Device

Ip6_rcv
Ip6_forward
Ip6_input
Ip6_output
Ip6_input
Ip6_output

Filter & Sample
Compute & Insert

Virtual Device

Telemetry Agent

Local Telemetry Agent Management

Streamer

Key

Kernel Hooks

Removing Telemetry Headers

Prototype

User Space

Kernel Space

From Device

To Device

Ip6_rcv
Ip6_forward
Ip6_input
Ip6_output

Filter & Sample
Compute & Remove

Virtual Device

Telemetry Agent

Local Telemetry Agent Management

Streamer

Key

Kernel Hooks

Service Data Records
Experimental Results

Example Measurements

- These are very basic measurements over our own “small” test-beds
  - Not very exciting, involves running our own applications and injecting our own traffic
  - Applications include:
    - Video streaming
    - Interactive TCP (Telnet, SSH)
    - Web browsing
    - Bulk TCP transfers
- Initial experiments applied end-to-end
  - On all traffic
  - Filtered and sampled traffic
  - To evaluate the efficacy of this native measurement approach
- Recently started looking at applying the technique to signaling protocols
  - SIP
  - Mobile IPv6
TCP Goodput

### Approach

<table>
<thead>
<tr>
<th>Service Port</th>
<th>Client Port</th>
<th>Client Security</th>
<th>Conv. Setup Time</th>
<th>Conv. Duration</th>
<th>Packets</th>
<th>Completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>32809</td>
<td>59.805</td>
<td>111</td>
<td>111</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>3774</td>
<td>32811</td>
<td>60.222</td>
<td>76.9</td>
<td>43.21</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>31769</td>
<td>32812</td>
<td>60.598</td>
<td>5.9</td>
<td>42.95</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>8153</td>
<td>32813</td>
<td>0.9</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14253</td>
<td>32814</td>
<td>60.731</td>
<td>54.4</td>
<td>443</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>53855</td>
<td>32815</td>
<td>57.805</td>
<td>121</td>
<td>111</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

*Blue: FTP Control Channels
*Black: Data transfer from server to client (GET)
*Red: Data transfers from client to server (PUT)

Example results obtained over operational broadband network through ADSL Connection

Graph for MQET
- X-points: Data packets from the server
- Red Dots: Acks from the client

Graph for MPUS
- X-points: Acks from the server
- Red Dots: Data packets from the client

Video Streaming

### Approach

Min: 8 / Max: 89 / Ave: 20.9 (msec)

One-way Delay over Time (Video/UDP Stream)

Inter-Packet Variation (Jitter) Vs. Time (Video/UDP Stream)

Min: -68 / Max: 58 / Ave: -0.00049
SIP Signalling

- Applied to
  - SIP call set-up
  - SIP client registration
- Example telemetry agents developed
  - Sip_Delay: application-agnostic agents adding simple timestamps to all SIP/UDP messages sent/received
  - Sip_Filter: stateless application-aware agent that can be configured at runtime to filter on specific SIP/UDP messages (e.g. INVITE)
  - Sip_Register_Time: stateful application-aware agents applied to the measurement of SIP client registration with register proxies.
Next Steps & Concluding Remarks

- Study and quantify the performance/cost in deployment and operation of this scheme
- Engage with interested parties
  - Larger trials on operational or test-bed IPv6 networks
- Evolve SIP and mobile service monitoring
- Study other application domains

Acknowledgements

- Dimitrios Pezaros from Lancaster University, is an Agilent sponsored PhD student studying the efficacy and applicability of the presented native approach to telemetry
- Evangelos Bouzianis from Lancaster University, was sponsored by Agilent during his MSc dissertation and developed the SIP monitoring capabilities (http://www.lancs.ac.uk/postgrad/bouziani/)
- Prof David Hutchison from Lancaster University, is Dimitrios academic supervisor and has supported him on this work and provided continual and valuable input.
- Prof Joe Sventek, was formerly head of Agilent Research Labs in Scotland before accepting his current chair with Glasgow University. This project was initiated under his supervision and he has again provided continual support and valuable input.