

CHARACTERISING USER EXPERIENCE AND CRITICAL PATH IN MOBILE APPLICATIONS

Internet Measurement Conference (IMC 2017)

Queen Mary University of London – Nov 1/3

N. Rattanavipanon♦, J. Newman●, A. Finamore■, D. Perino■, C. Soriente◆, N. Vallina-Rodriguez*
 ♦University of California Irvine, ●Northwestern University, ■Telefonica Research, ◆NEC Labs Europe, *IMDEA Networks/ICSI

MOTIVATION: QoE in Mobile Traffic

- Users **Quality of Experience (QoE)** analysis is paramount for telcos to drive business, and for users to verify their SLAs with the operators.
- QoE on web traffic is “standardized” via Page-Load-Time (PLT), Google Speedy index, etc.

...but...

“What is the PLT for a generic mobile app?”
“What’s the impact of DNS/TLS/etc. ?”

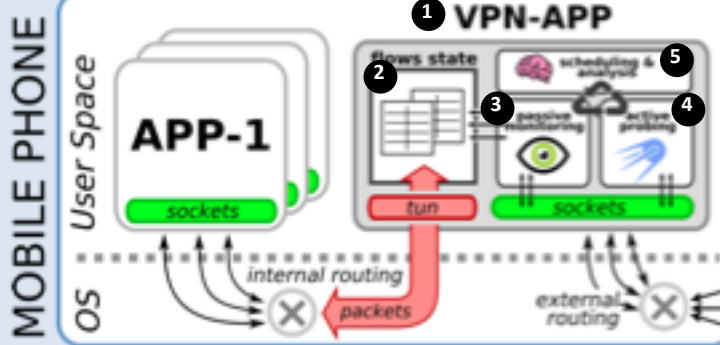
GOAL: From Bottlenecks to Solutions

- Analyze apps in different scenarios capturing all **traffic and user engagement**, e.g., application startup, user interaction, application background/foreground
- Build the **traffic waterfall**, i.e., flows pattern and content exchanged over time
- Dissect the **critical path**, i.e., identify which flows (if shortened) can improve QoE, and assess the impact of DNS, TCP handshake, TLS, etc., for the flows

HOW: Active and Passive On-Device Measurements

Networking Layer

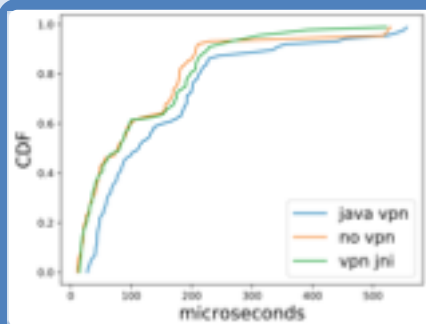
- Use VPI APIs to study traffic on-device (internal forward to a tun interface)
- Flow reassembly from raw packets to both investigate content, and extract measurements



Analysis Layer

- Passive analysis to collect per-flow & per-packet metrics
- Active probing to create a comparison baseline & do root cause analysis
- Supervisor to control & schedule activities, and perform analysis

Networking Layer: First Implementation



- A critical implementation detail is the extra overhead imposed by on-device traffic acquisition and processing
- Java requires to poll `tun` separately from the remaining sockets while using a C++ Java Native Interface (JNI) we can handle all sockets at once (with a single `select()` call)
- A JNI implementation reduces the kernel /user-space packets copy and processing overhead

Analysis Layer: Early Results

Fraction of critical path spent on a given activity

Network Activity	App 1	App 2	App 3
DNS (%)	1 - 2	0 - 0	0 - 3
TCP handshake (%)	4 - 12	1 - 2	1 - 11
TLS handshake (%)	2 - 35	2 - 5	16 - 40

Lower bound: sum of times a given activity is alone on the critical path.

Upper bound: sum of the total time of each activity on the critical path.

