Performance of Host Identity Protocol on Lightweight Mobile Devices

Andrey Khurri

Helsinki Institute for Information Technology

firstname.lastname@hiit.fi

4th FRUCT seminar

Tampere, Finland

October 29-31, 2008

Outline

- Research problem
- Host Identity Protocol (HIP)
- Device specifications & network setup
- Performance metrics
- Results and analysis
- Concluding remarks

Research Problem

- Moving TCP/IP stack to lightweight platforms
  - Adjusting for constrained devices such as PDA, phone, sensor, microcontrollers
    - Examples: µTCP/IP, µIPv6, lightweight IKE
  - Running existing ”desktop” solutions if performance is acceptable
    - Example: Elliptic-Curve Cryptography on mobile healthcare devices
- Are unmodified IP mobility and security solutions ready to be used on lightweight devices?
  - Limited hardware resources
  - Computationally expensive software-based cryptography

Host Identity Protocol

- Host Identity Protocol –

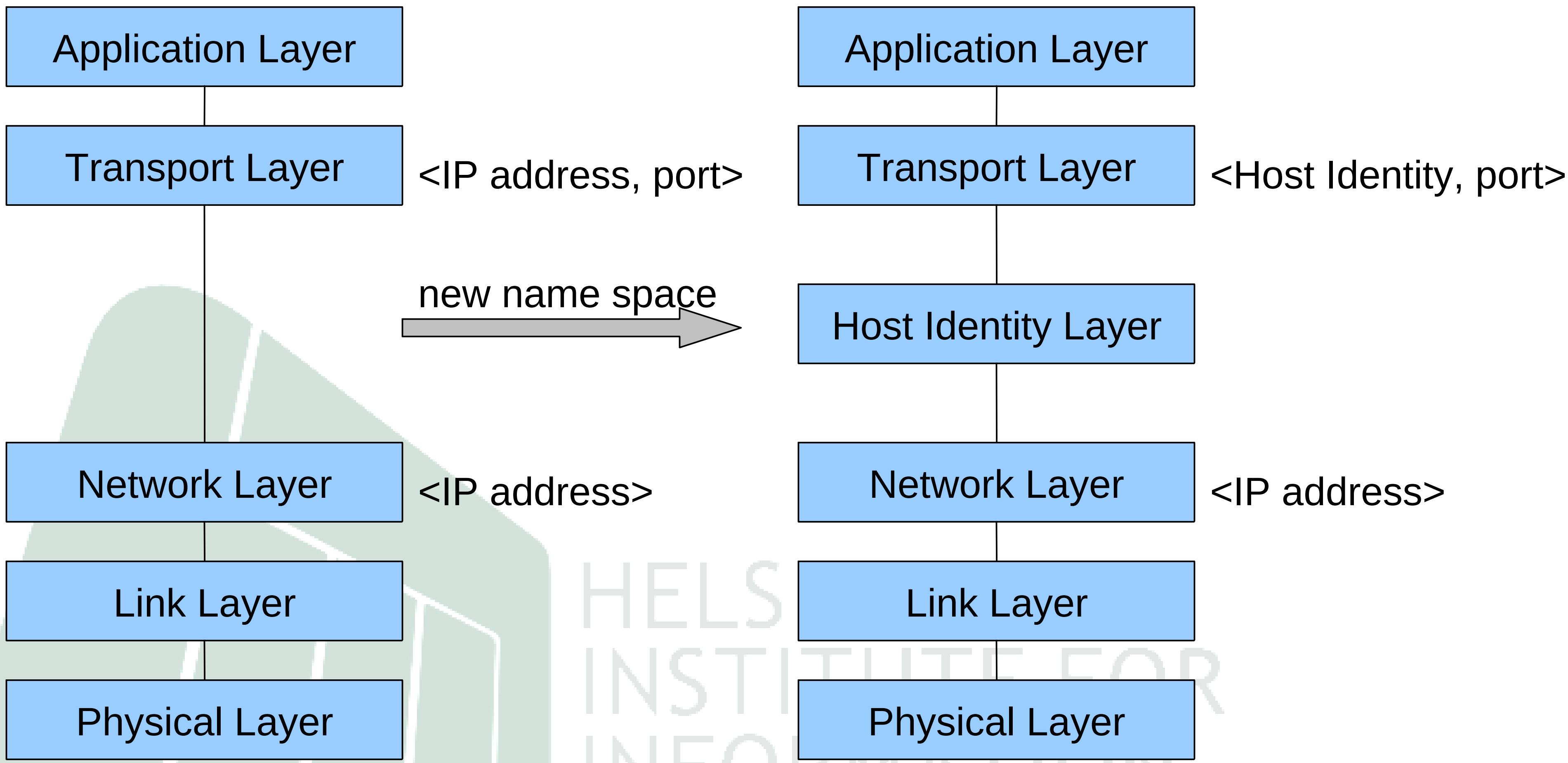
  a ”universal” solution to many Internet problems

  - Three open-source implementations
  - No experience with running it on lightweight devices
  - Concept similar to other security and mobility protocols
    - Assymetric key pair cryptography
    - IPsec ESP for data protection

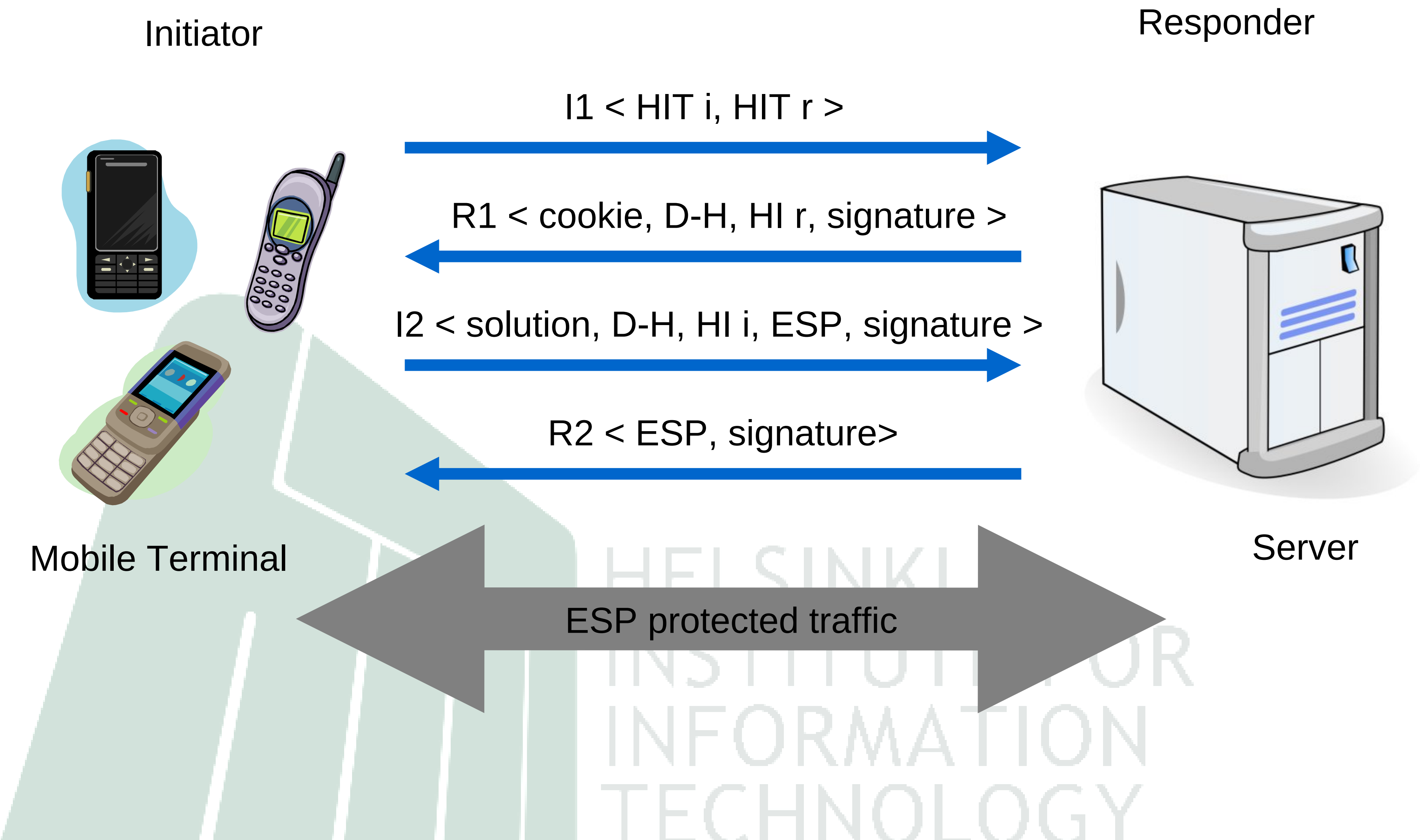
Host Identity Protocol (cont'd)

- Specified by IETF (RFC 5201-5207)
- Decouples IP layer from the above layers
  - Locator/identifier split
- Public-private key pairs to authenticate hosts
- IPsec ESP protocol to protect user data
- Provides
  - End-to-end security
  - Authentication
  - Mobility
  - Multihoming
  - NAT traversal

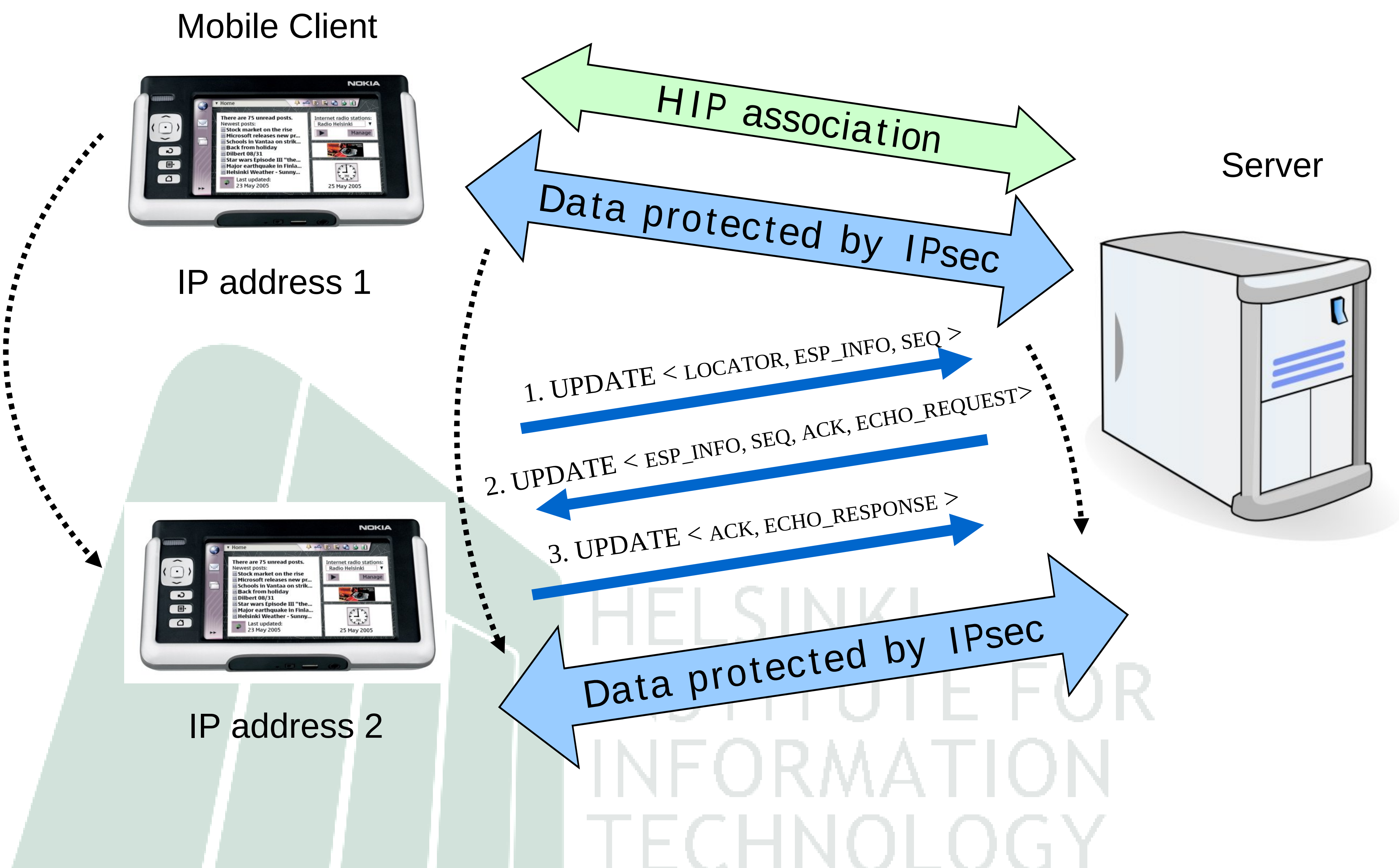
HIP Protocol Stack



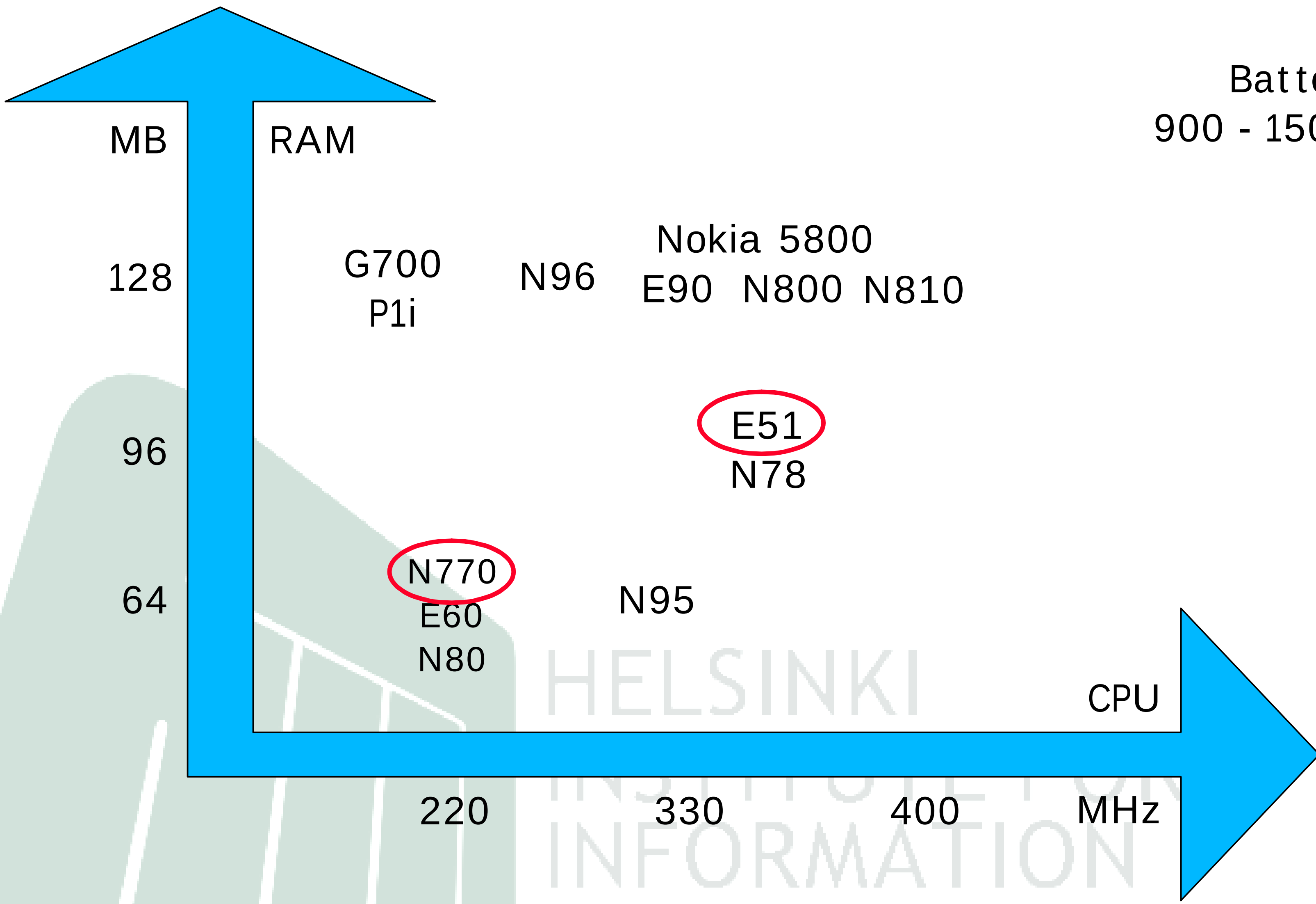
HIP Base Exchange



HIP Mobility



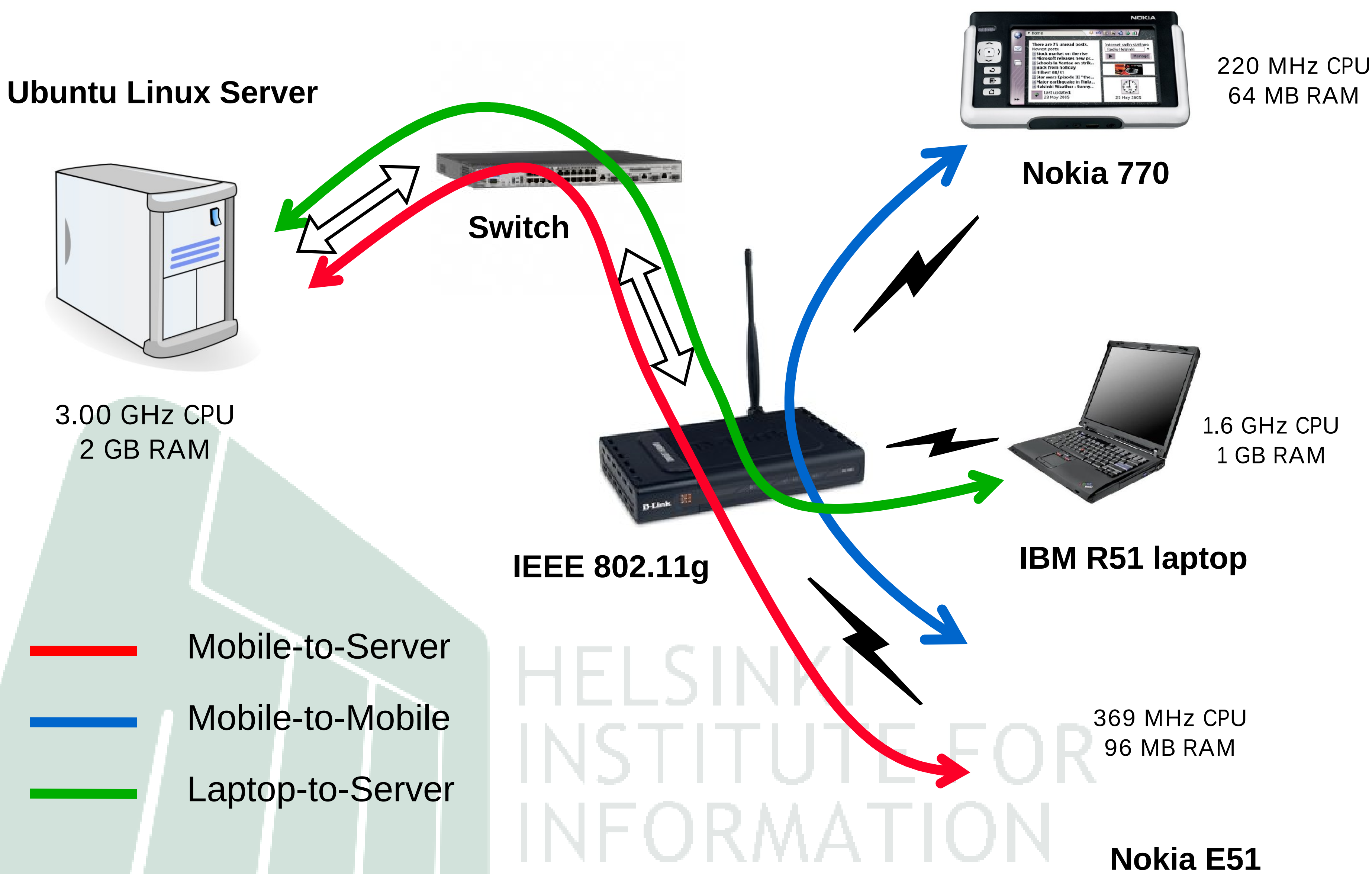
Mobile Device Specs Evolution



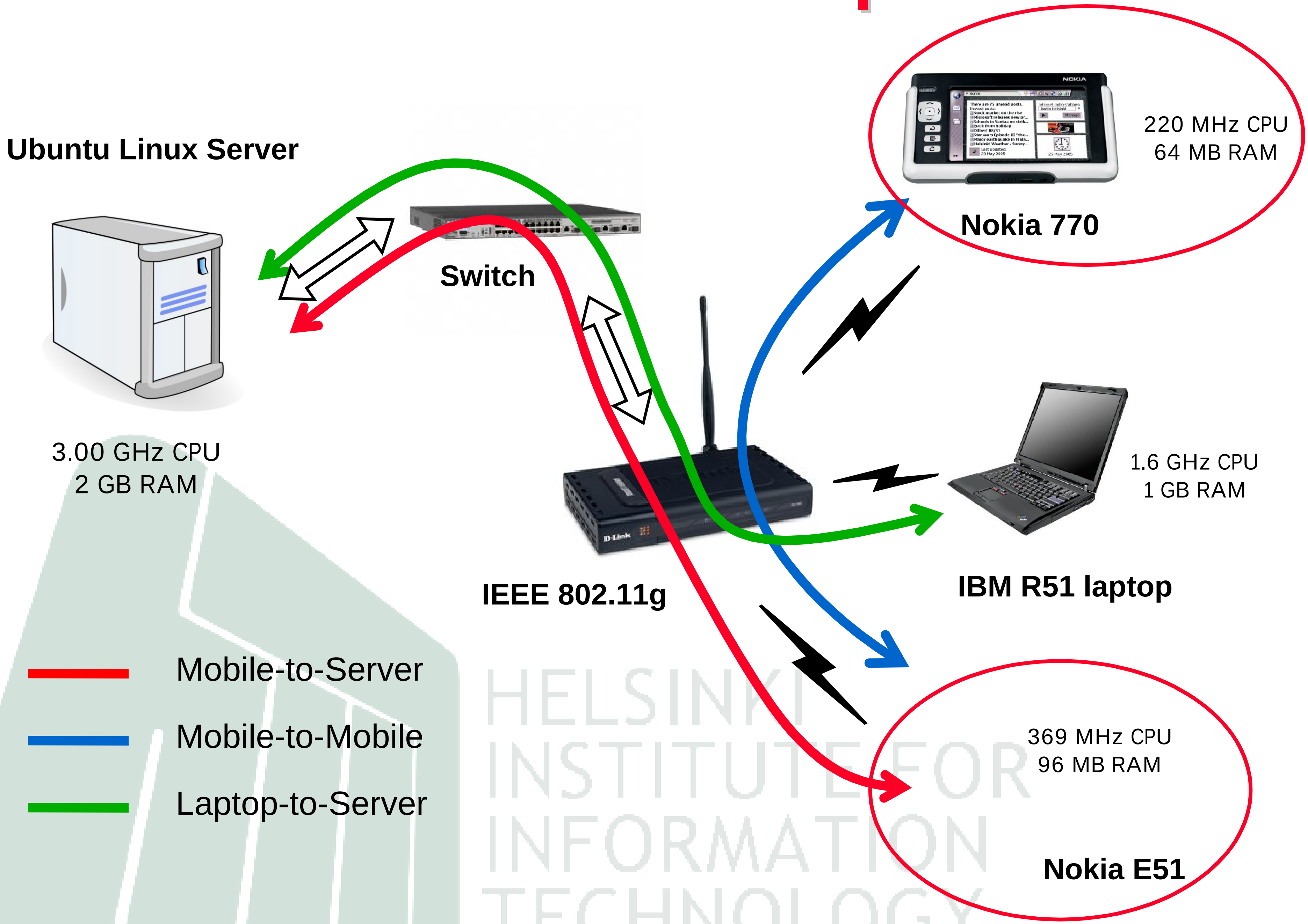
Device Specifications

	Nokia 770 Internet Tablet	Nokia E51 smartphone
CPU, MHz	220	369
RAM, MB	64	96
Battery, mAh	1500	1050
Connectivity	WLAN, Bluetooth	3G, WLAN, Bluetooth
Operating System	Linux Debian, Maemo	Symbian, S60 3rd Edition

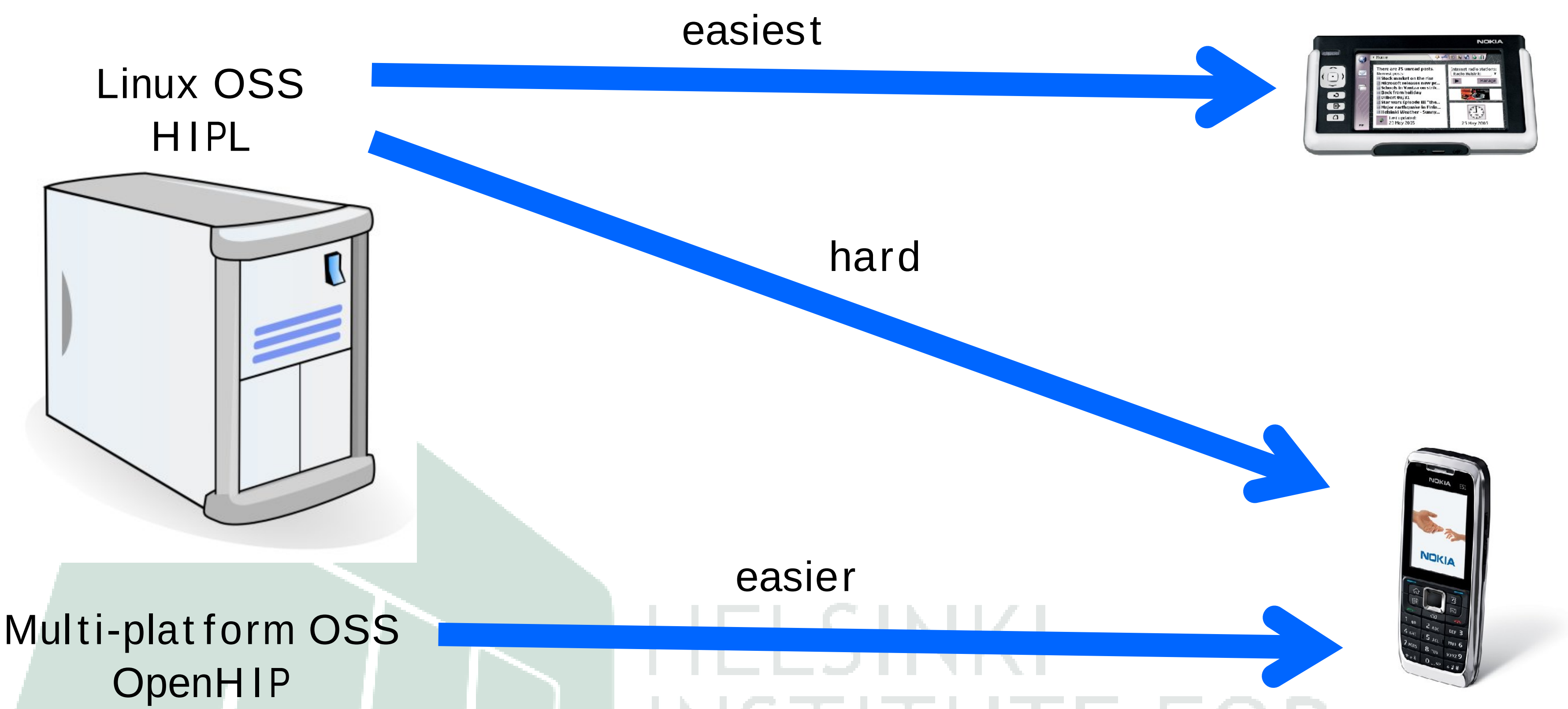
Network Setup



Network Setup



Porting from Desktop to Mobile



Performance Indicators

- HIP Base Exchange duration
- Mobility Update duration
- TCP throughput
- Power consumption
- CPU and memory load

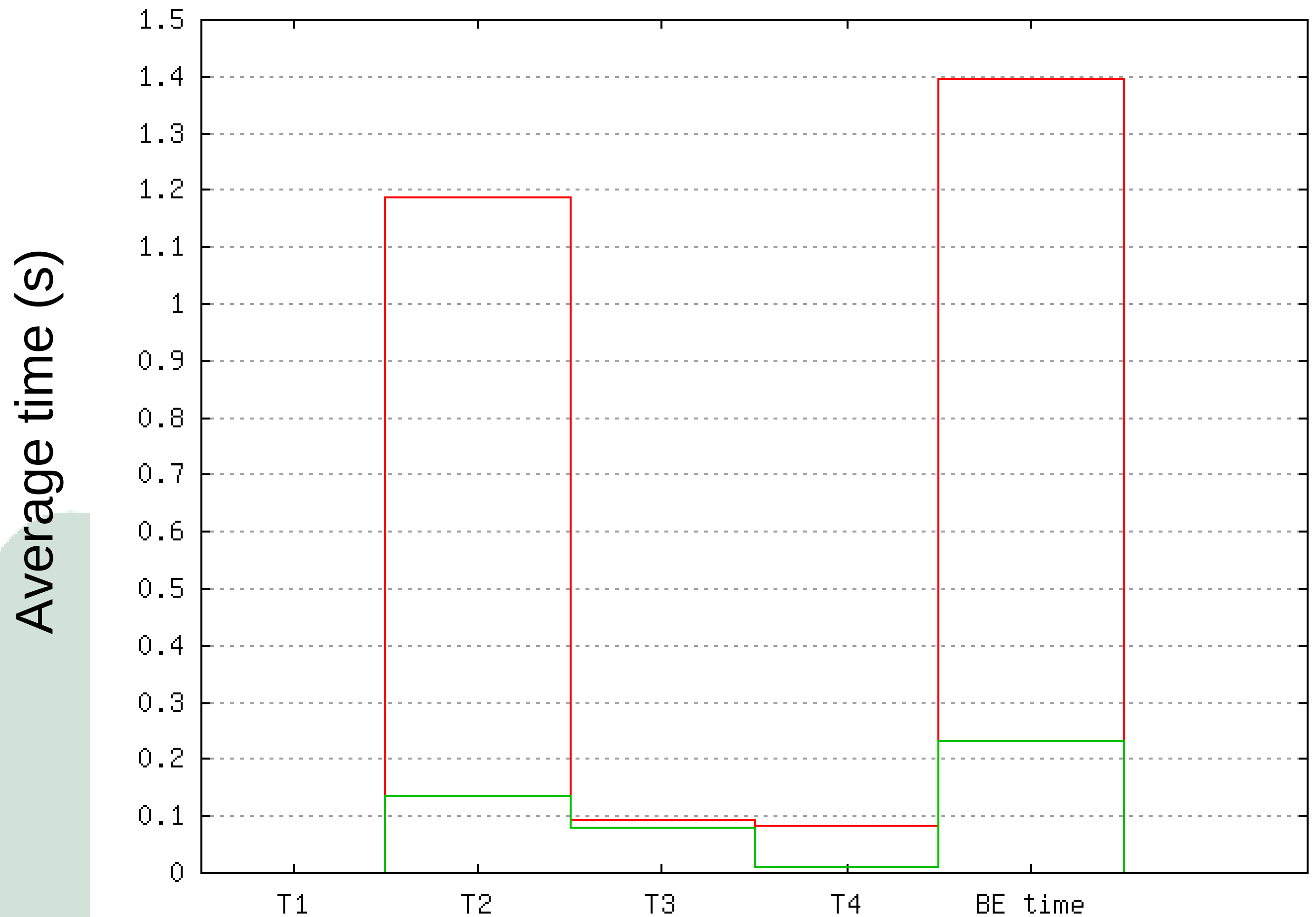
Results

HIP

Nokia 770

Nokia E51

Duration of HIP Base Exchange

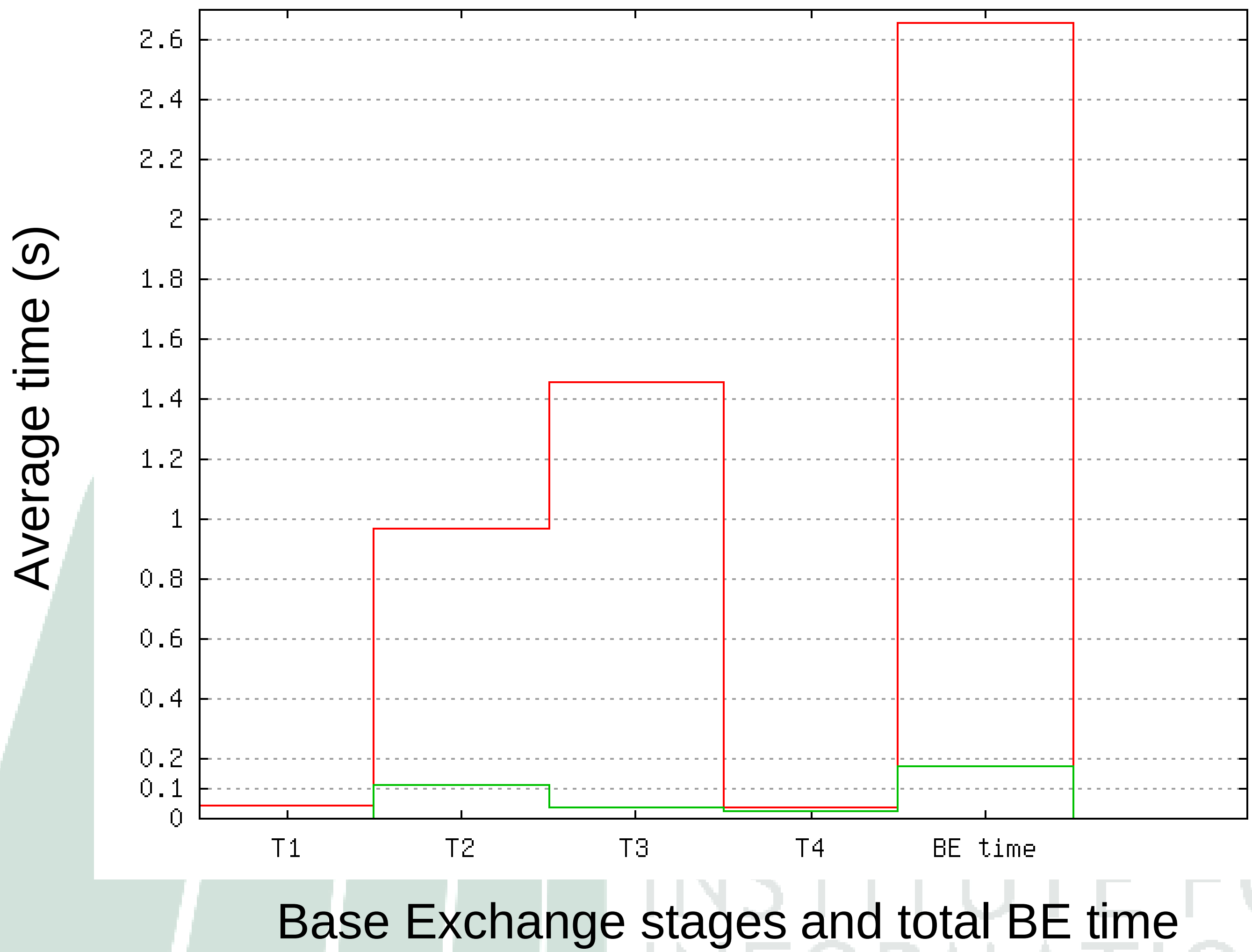


1024-bit RSA keys
1536-bit DH Group

Base Exchange stages and total BE time

Mobile Client → Server

Duration of HIP Base Exchange (cont'd)



Mobile Client → Mobile Client

Base Exchange Duration with HIPL and OpenHIP

Nokia E51	**Mean /** Standard Deviation (s)	
Scenario / Implementation	*HIPL*	*OpenHIP*
Phone → Server (Active)	**3.169** / 0.108	**3.089** / 0.170
Phone → Server (Standby)	**1.677** / 0.063	**1.895** / 0.122
Server → Phone (Active)	**3.313** / 0.104	**2.758** / 0.106
Server → Phone (Standby)	**1.759** / 0.138	**1.851** / 0.074
Phone → Phone (Active)	**6.416** / 0.712	**4.297** / 0.073
Phone → Phone (Standby)	**3.781** / 0.125	**3.501** / 0.123

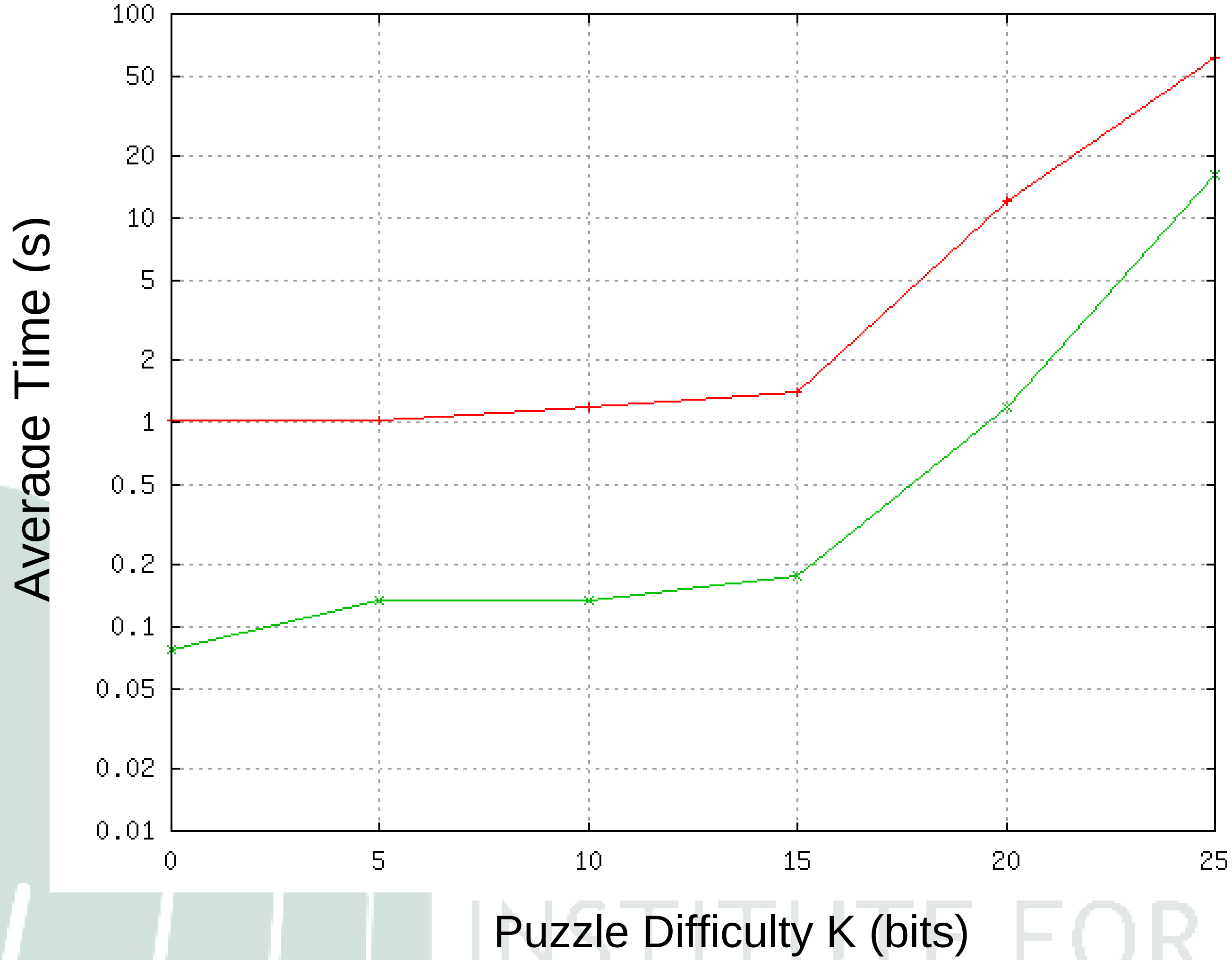
- Surprisingly, we found a significant difference in performance measured in *Active* and *Standby* phone states

Key Pair Creation of Different Size on Nokia E51

Nokia E51	**Mean /** Standard Deviation (s)		
Key Length (bits) →	512	1024	2048
DSA	**4.90** / 1.46	**31.48** / 16.54	**389.99** / 308.61
RSA	**0.51** / 0.13	**3.56** / 1.28	**40.73** / 31.20

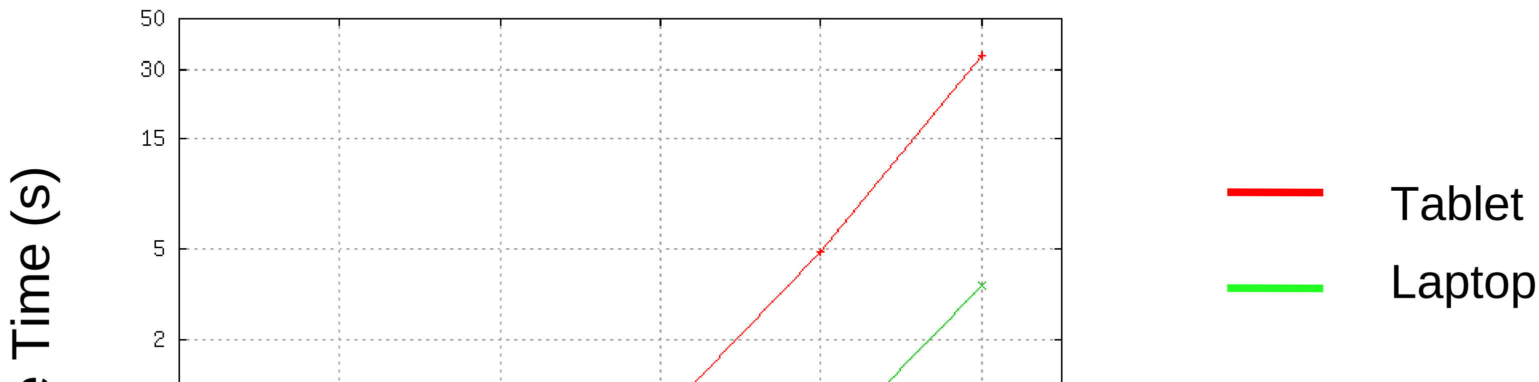
- The public-private key pair generation might stress the cell phone
  - Especially with key length > 1024 bits

Puzzle Difficulty Impact



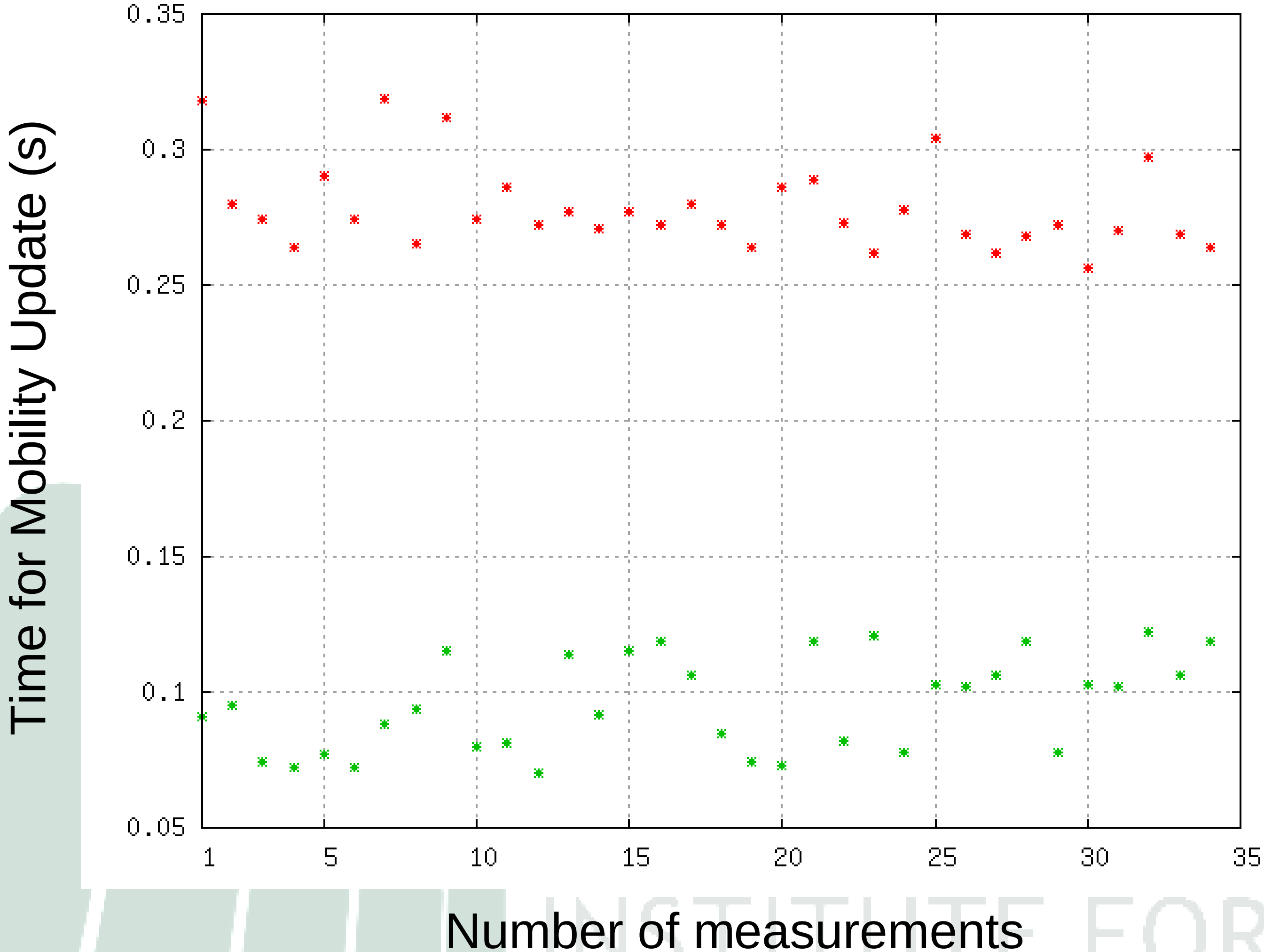
T2 processing time dependence on K

Influence of Diffie-Hellman Group ID



- With the 768-bit DH Group HIP association establishment with a server might be reduced up to 0.35 sec

Duration of Mobility Update



Tablet

Laptop

Average time: Tablet – 287 ms; Laptop – 100 ms

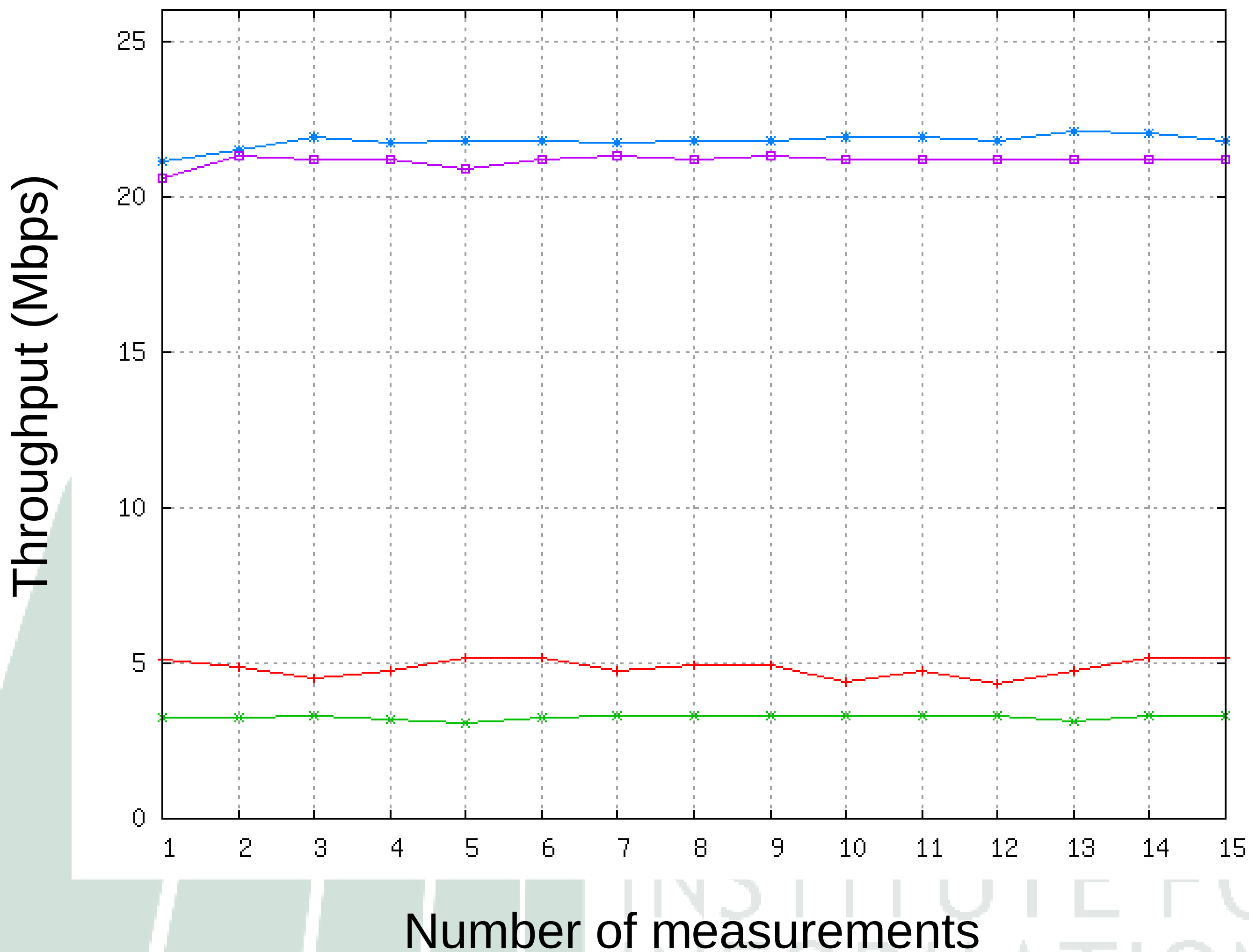
TCP Throughput

Average TCP throughput with Tablet and Laptop

Throughput	Mean / Standard Deviation (Mbps)			
	TCP	TCP + HIP	TCP + WPA	TCP + HIP + WPA
Tablet → PC	**4.86** / 0.28	**3.27** / 0.08	**4.84** / 0.05	**3.14** / 0.03
Laptop → PC	**21.77** / 0.23	**21.16** / 0.18		

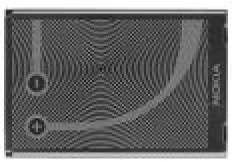
- Surprisingly, tablet only achieves 4.86 Mbps in a IEEE 802.11g WLAN (our laptop achieves 21.77 Mbps over the same link)
- WPA encryption has minor impact on the throughput
  - In contrast, ESP encryption involved with HIP reduces TCP throughput by 32%

TCP Throughput (cont'd)



Power consumption – Nokia 770

1500 mAh



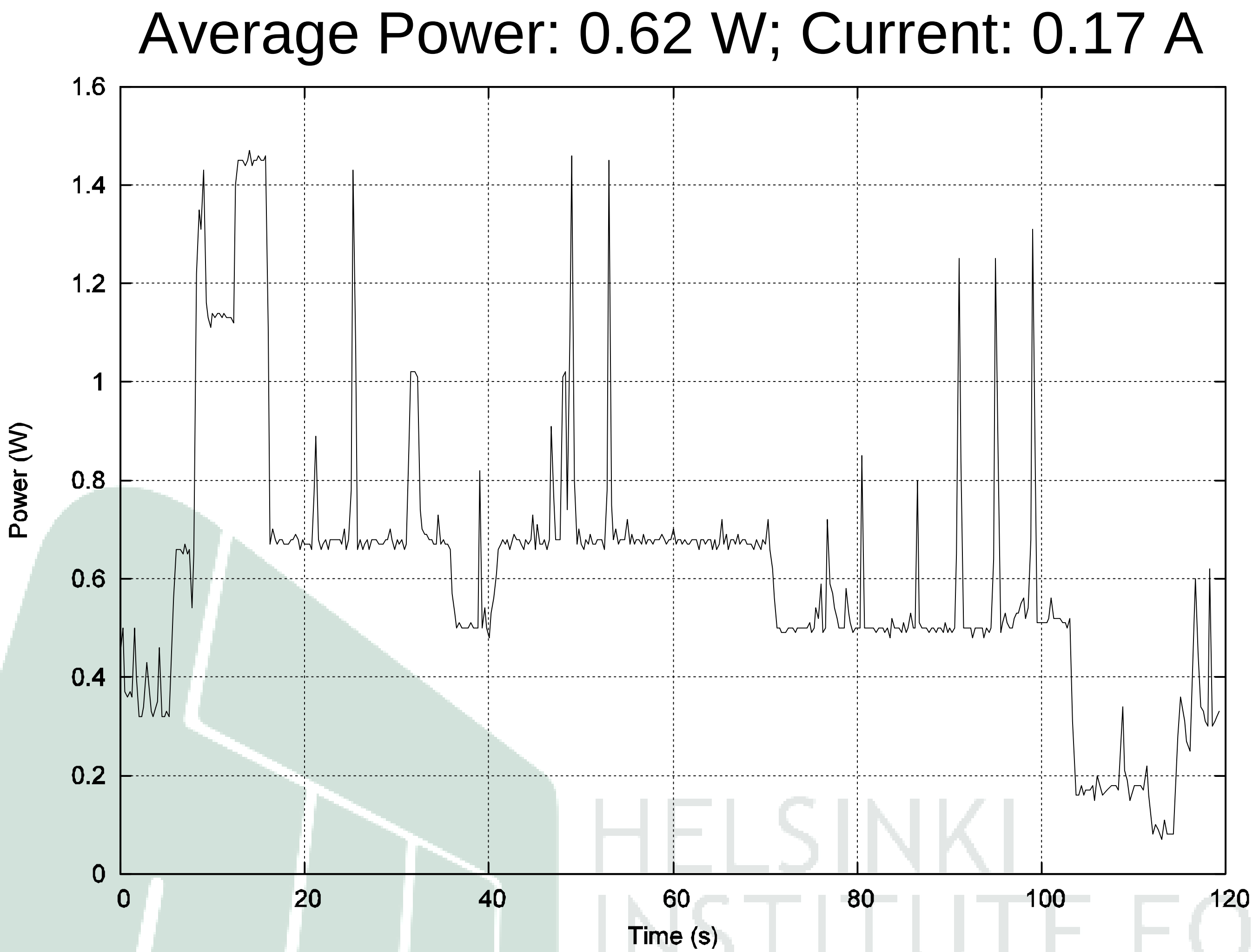
Application / Mode	Current (A)	Power (W)
HIP Base Exchange	0.36	1.33
ESP traffic (an app with HIP)	**0.38**	**1.41**
Plain TCP (an app without HIP)	**0.38**	**1.41**
Video stream from a server	> 0.50	1.85
Local video	0.27	0.99
Audio stream from a server	0.40 – 0.50	1.66
Local audio	0.20	0.74
Browsing (Active WLAN)	0.35 – 0.50	1.57
Passive WLAN	0.12	0.44
Standby mode	< 0.01	0.04

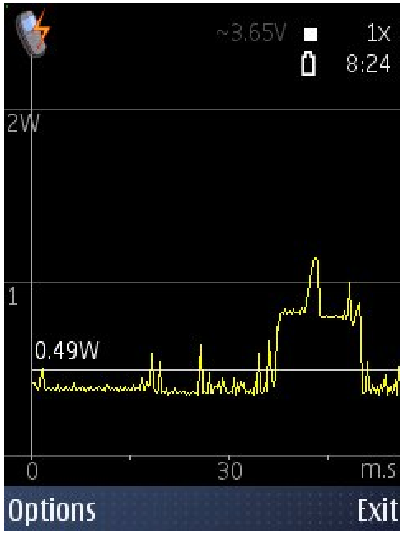
- The use of HIP does not noticeably affect the speed of battery depletion
- BUT energy cost per byte is higher with HIP due to reduced throughput

Power consumption (cont'd)

- Almost no difference between HIP-enabled and non-HIP applications
  - Tablet's CPU is kept busy always upon data transmission over WLAN
- HIP consumes more energy per byte than plain TCP/IP
  - IPsec data encryption requires a notably longer CPU utilization for a data bulk to be transferred
  - Longer CPU utilization causes more energy consumption for this particular task

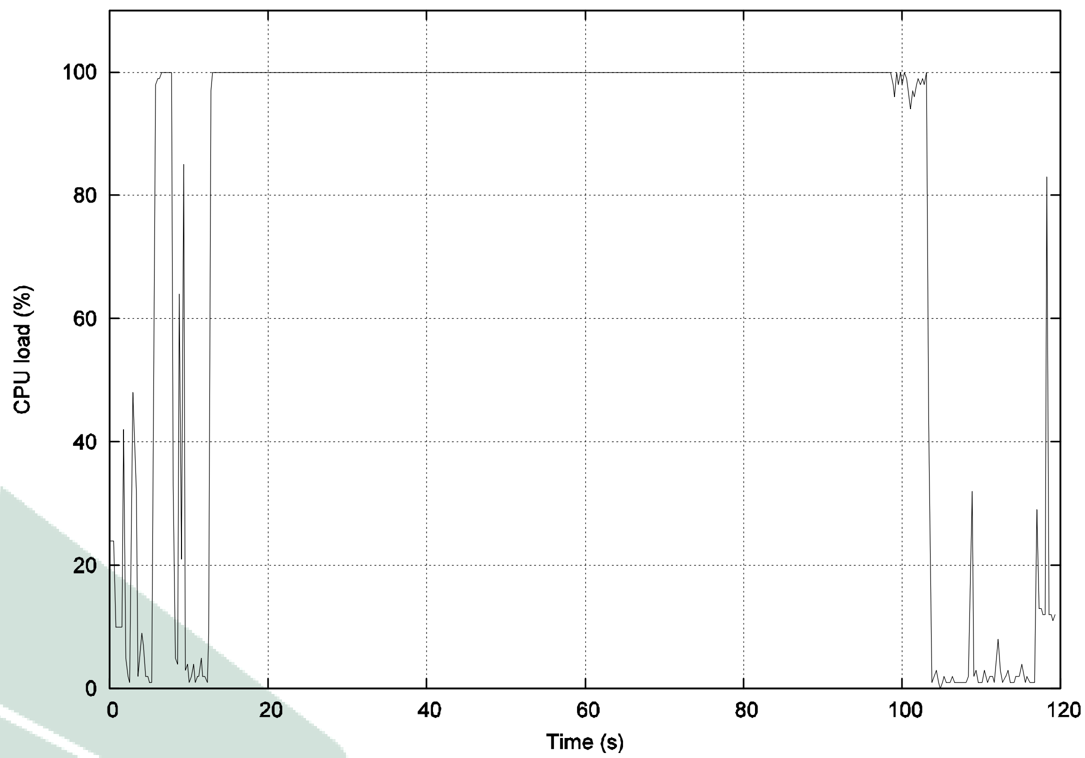
Power Consumption – Nokia E51





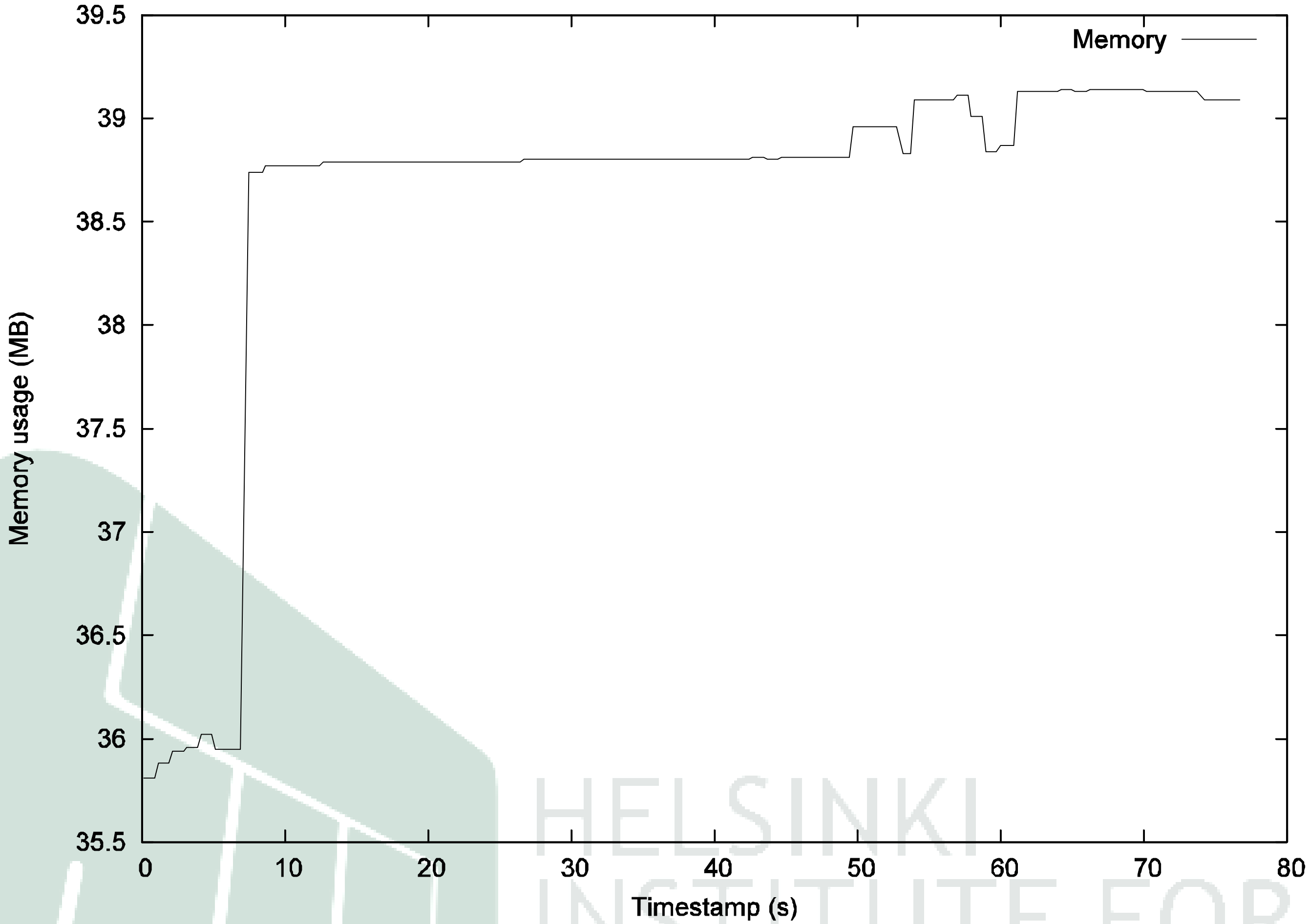
No HIP daemon: 200mW/60mA (18 h) and HIP BEX: 340mW/90mA (12 h)

OpenHIP Daemon Initialization CPU Load on Nokia E51



- CPU usage is close to 100% at the initialization phase but low in the idle mode

OpenHIP Daemon Initialization with BEX RAM Usage on Nokia E51



- HIP increases memory usage by 3 MB

Conclusions

- Unmodified HIP
  - might be used in a number of scenarios with a lightweight device communicating via a single proxy server
  - BUT is too heavy for two mobile hosts and/or multiple parallel HIP associations

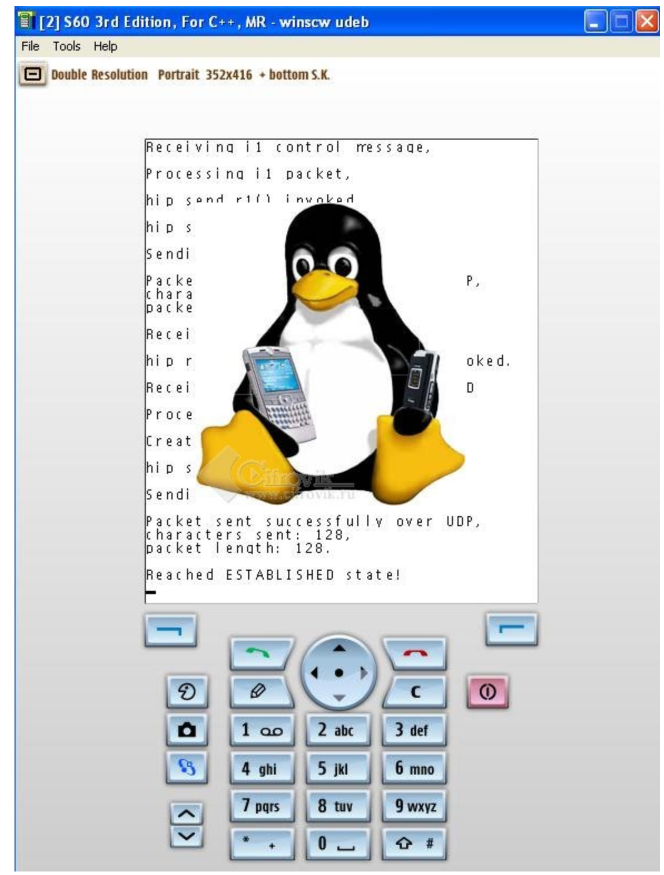
BEX, sec	Nokia 770	Nokia E51 (standby)	Nokia E51 (active)
Mobile → Server	1.4	1.7	3.2
Mobile → Mobile	2.6	3.5	6.4

Conclusions (cont'd)

- OpenHIP implementation has been a lot more portable (works now on many OS: Linux, Win, MacOS) and showed slightly better performance
- HIP implemented natively using Symbian C++ would have better performance

- Applicability of the measurement results to
  - A wide range of mobility and security protocols
    - most such protocols are based on similar public key and IPsec ESP operations like HIP
  - Other models of smartphones with similar hardware

Thank You!



HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY