Encrypted Tunnel Through Virtual Network Interface

Safiqul Islam
INF -9090 – Project Presentation
University of Oslo

Outline

- Introduction
- Background
  - Virtual Private Network
  - Virtual Network Interface
  - Link Local Addressing
  - Cryptography
    - Asymmetric Key Cryptography
    - Symmetric Key Cryptography
  - Design
  - Evaluation
  - Conclusion and Future Work

Introduction

- Virtual Private Network(VPN) provides secure communication over the insecure public network.
- Most of the current open source methods do not support *Mobility* - such as : Vtun and OpenVPN
- Some proprietary methods: Cisco VPN, and Netmotion support mobility
- Designing a system that uses a virtual network interface and supports mobility is the primary goal of this system.

Virtual Private Network

- Provides secure communication over the insecure public network via
  - Authentication
  - Encryption
  - Compression
  - Tunneling
- IPSec
  - Tunnel Mode
  - Transport Mode

Virtual Network Interface

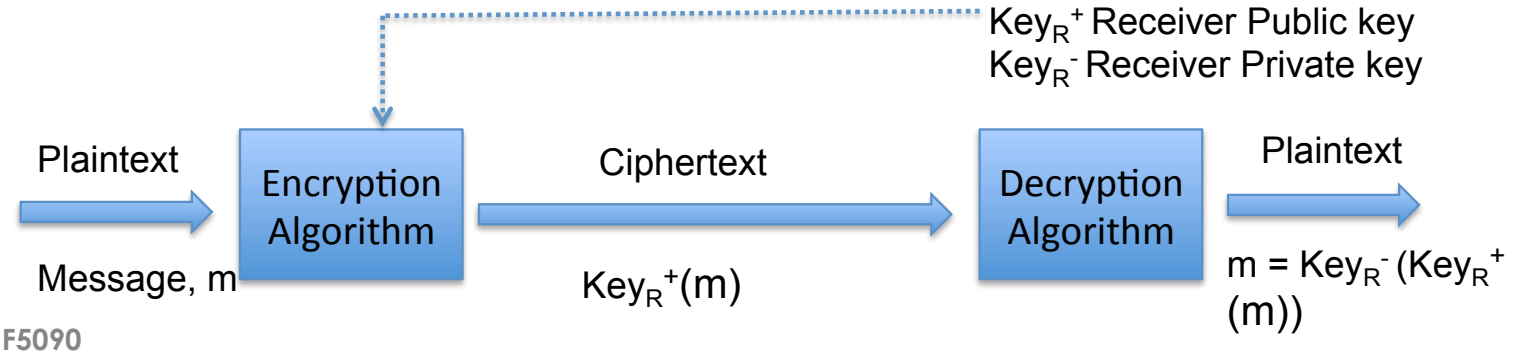
- An Ethernet like device
  - Receives packets from the userspace program
  - Sends them to the userspace program before sending it via physical media.
- TUN/TAP driver is used to create Virtual Network Interface
  - TUN is used for reading and writing IP packets
  - TAP is used for reading and writing Ethernet frames
- By using TUN/TAP for making connection with the other end, we can add the support of mobility when the connection is moved to different location.

Cryptography

- An art of science for transforming intelligible text to an unintelligible one and vice versa.
  - Intelligible text is plain text
  - Unintelligible text is cipher text
- Public-key cryptography
  - Have a pair of cryptographic keys
    - Public and private – mathematically linked

Public-key Cryptography

- Public key is publicly known, and private key has to be kept secret.
- Encryption is done using the public key of the user, and decryption is done using the private key,
- Digital signature is also performed using this cryptography.

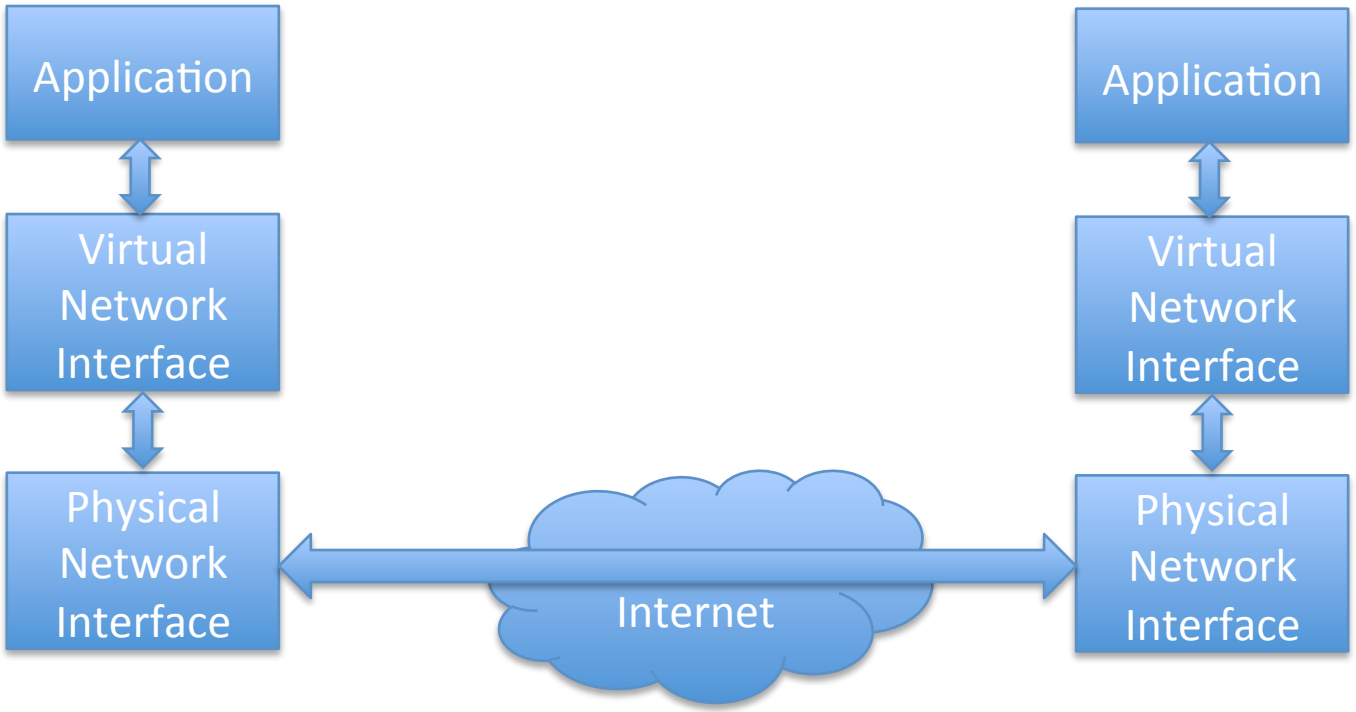


Link Local Address

- Intended for addressing on a single link or for a Local Area Network
- Routers do not forward such packets
- Both IPV4 and IPV6 have reserved a block for link local addresses.
  - 169.254.0.0/16 for IPV4
  - Fe80::/64 for IPV6

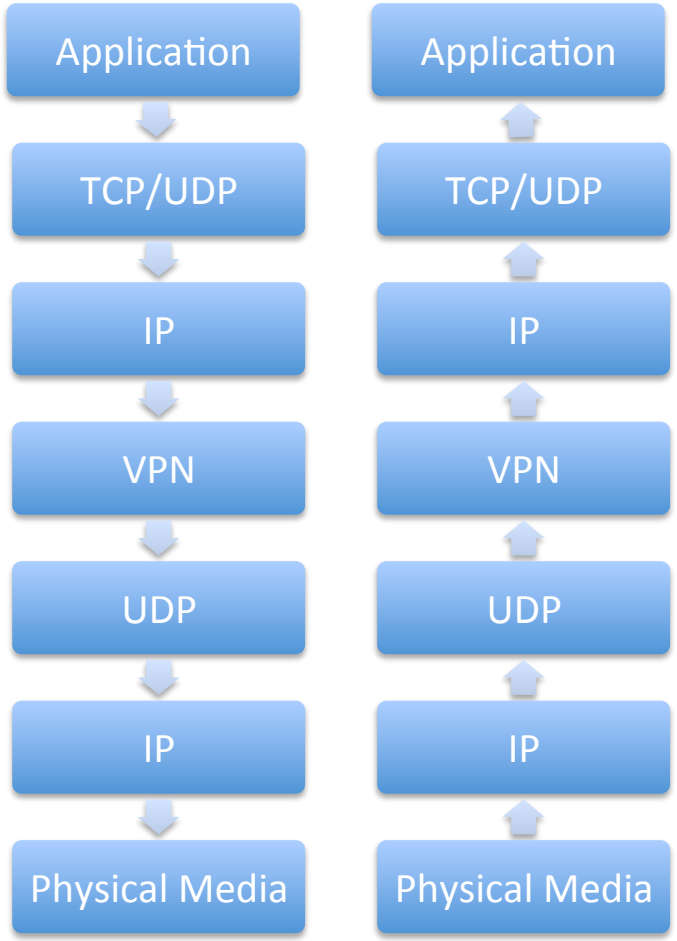
Design

- Provides Server/Client functionality
- Uses TUN for virtual network interface



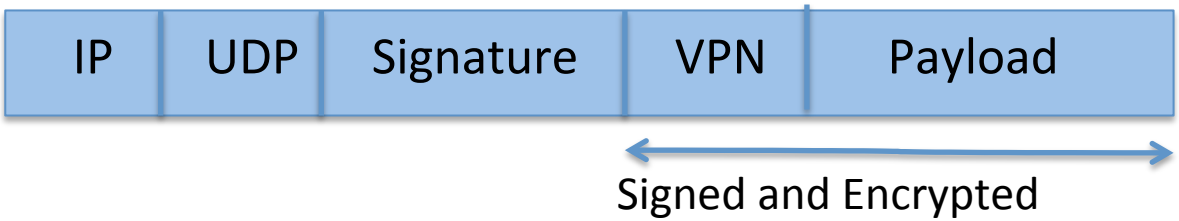
Design

- IPv4 link local addresses are used for configuring the TUN interfaces.
- To successfully traverse the network packet is encapsulated into an UDP packet.



Design

- Encryption
- Integrity checking
- Mobility



Challenges

- Transport Protocols
  - UDP – TCP over TCP problems
  - Simpler methods and higher success rates
- Kernel Space vs User Space
  - Portability
  - Efficiency

Evaluation

- Metrics
  - Throughput
  - Latency
- Mobility Test

Testbed 1

Device	Configurations
Host 1	Processor: Intel Pentium 4, 1.60GHz RAM : 0.5 GB OS : lubuntu Network interface: Realtek RTL-8169 Gigabit Ethernet
Host 2	Processor: Intel Pentium 4, 1.60GHz RAM : 0.5 GB OS : lubuntu Network interface: Realtek RTL-8169 Gigabit Ethernet
Switch	8 port Gigabit desktop switch Vendor: DLINK Model : DGS -1008D

Testbed 2

Device	Configurations
Host 1	Processor: Intel core i7, 2.93 GHz RAM : 8 GB OS : Fedora 18 Network interface: Intel 82578DM Gigabit Network Connection
Host 2	Processor: Intel core i7, 2.93 GHz RAM : 8 GB OS : Fedora 18 Network interface: Intel 82578DM Gigabit Network Connection
Switch	8 port Gigabit desktop switch Vendor: DLINK Model : DGS -1008D

File Transfers over SSH

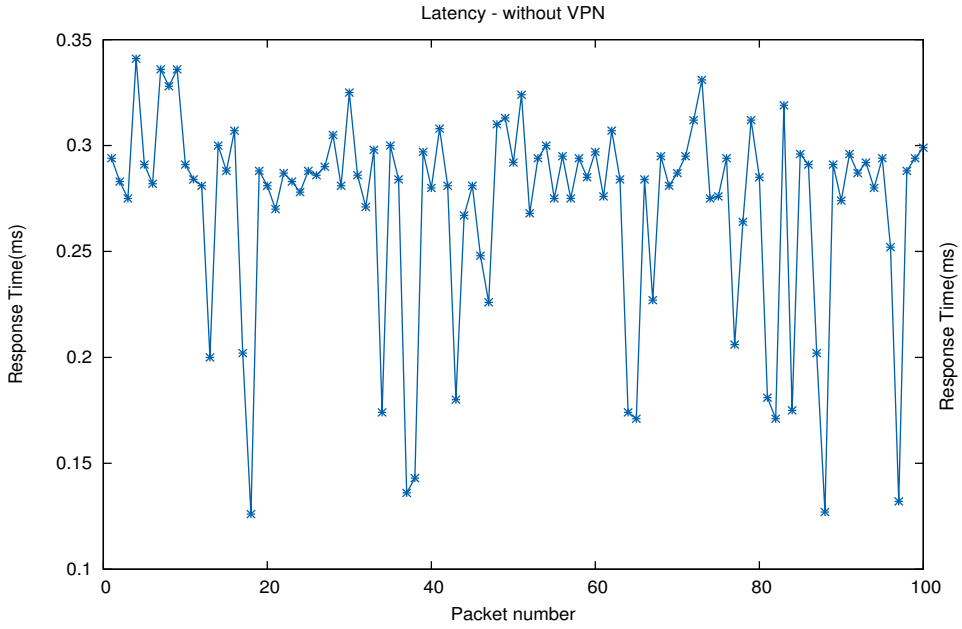
File Size (KiB)	Throughput (KBps)
32	1.1
64	1.5
128	1.4

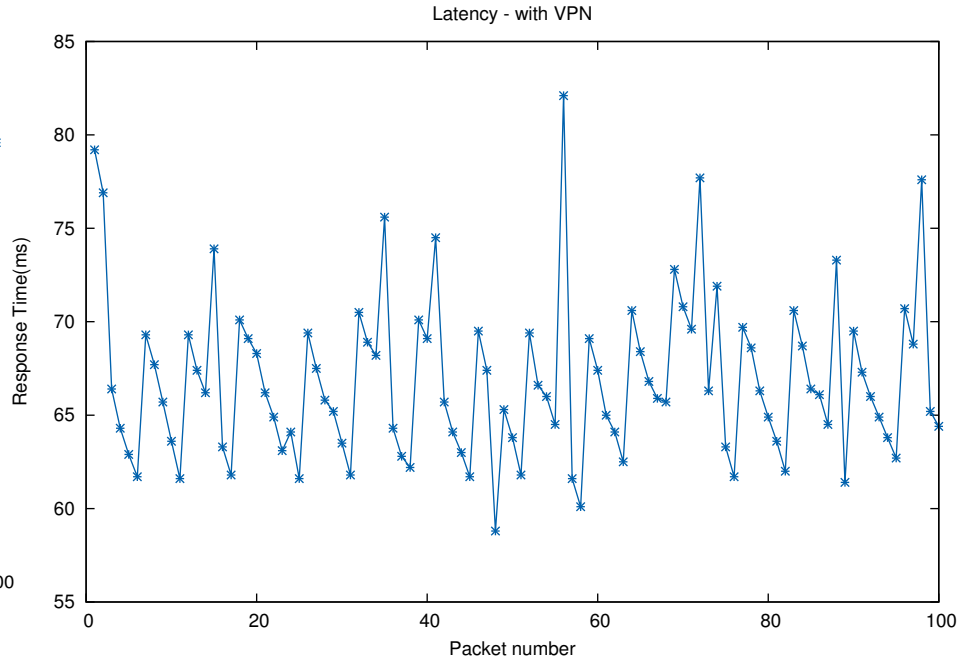
Table: File Transfers over SSH for testbed 1

File Size (KiB)	Throughput (KBps)
32	16
64	21.3
1024	18.6
10240	17.7

Table: File Transfers over SSH for testbed 2

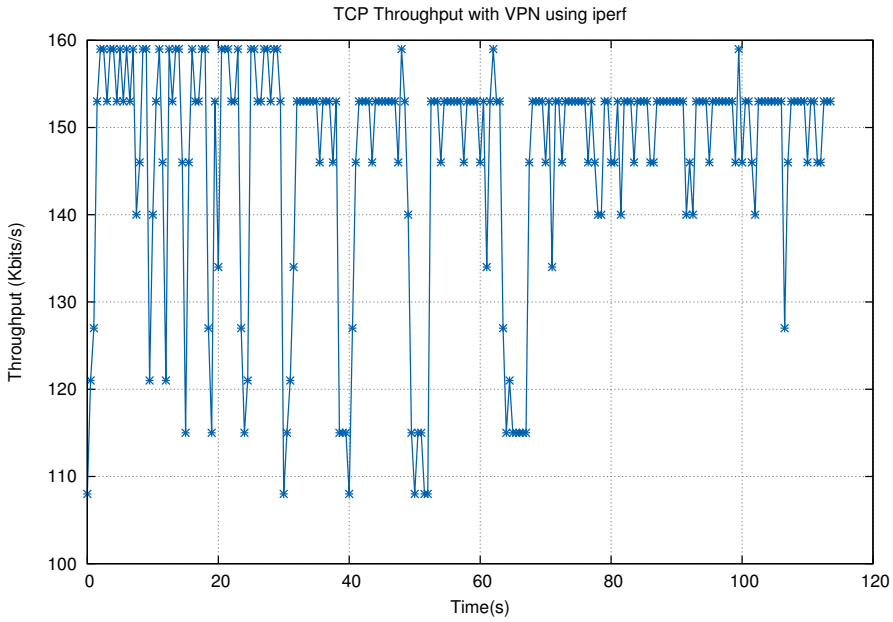
Latency

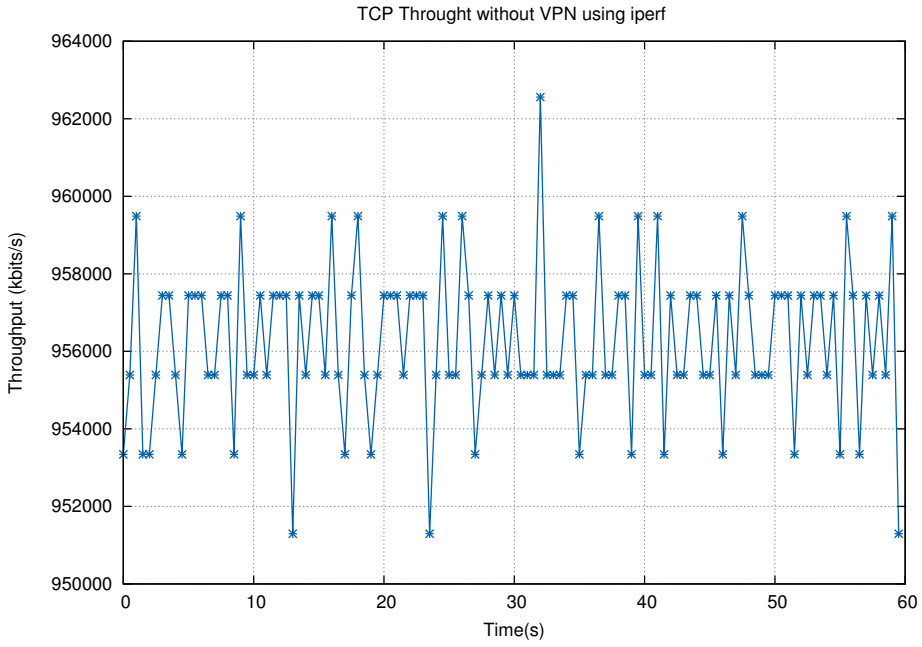




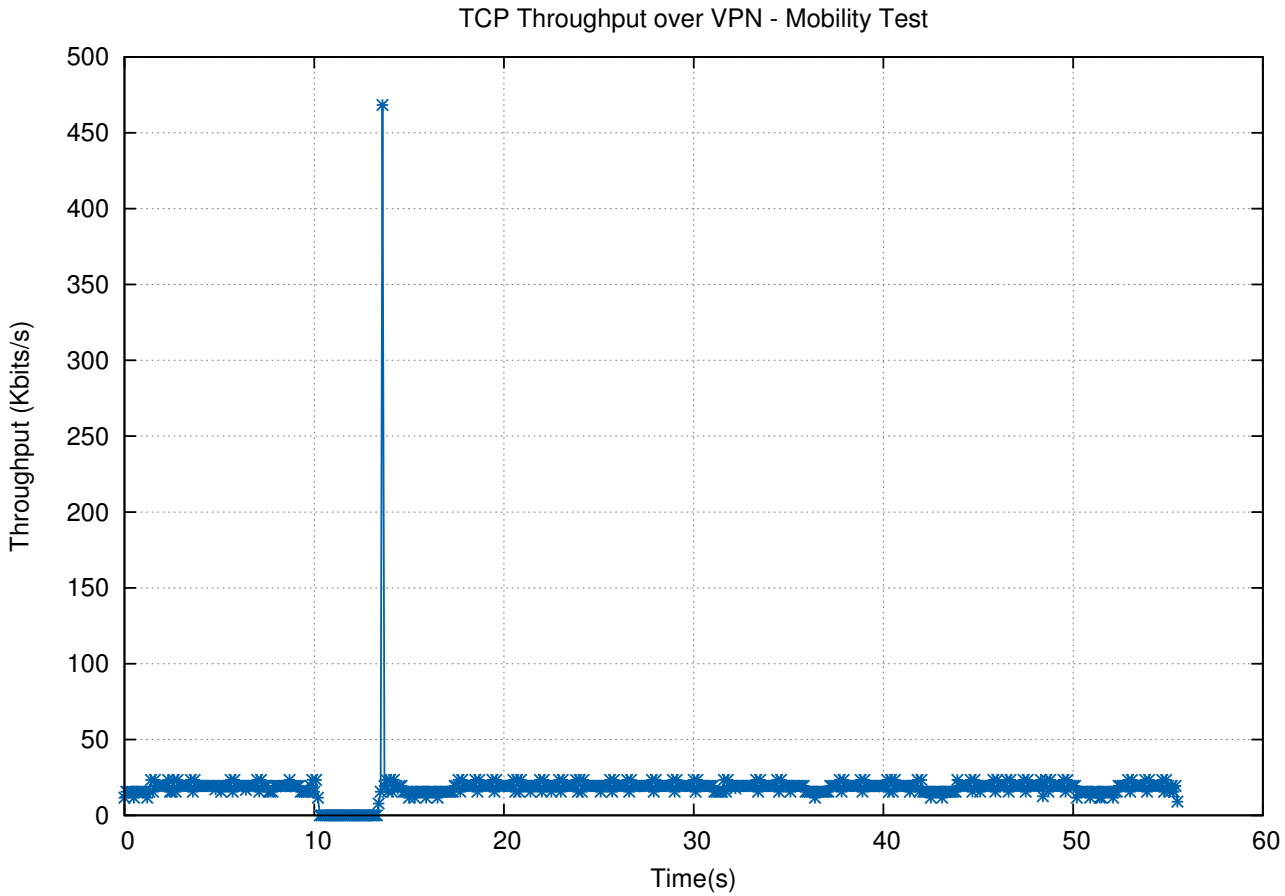
	Total Packets	Minimum	Average	Maximum
Default	100	0.126	0.270	0.341
Our System	100	58.141	66.908	82.157

Throughput


TCP Throughput with VPN using iperf
Throughput (Kbits/s)
Time(s)
160
150
140
130
120
110
100
0
20
40
60
80
100
120


TCP Throught without VPN using iperf
Throughput (kbits/s)
Time(s)
964000
962000
960000
958000
956000
954000
952000
950000
0
10
20
30
40
50
60

Mobility


TCP Throughput over VPN - Mobility Test
Throughput (Kbits/s)
Time(s)
0
50
100
150
200
250
300
350
400
450
500
0
10
20
30
40
50
60

Conclusion

- Implemented and evaluated an encrypted tunnel where we used virtual network interface.
- Supports mobility
- However, regular system outperforms our system
- There are some future works :
  - Symmetric key cryptography.
  - CPU performance.
  - IP address derivation from the public key

Acknowledgement

- We would like to thank Hans for helpful discussion and valuable feedback.

Thanks and Questions ? ☺