Toyshark’s migration to Kotlin

이창환
lipisoft@gmail.com

차례

- Toyshark 프로젝트 소개
- Kotlin 소개
- 개발언어를 Java에서 Kotlin으로 변경
- 배운 점

발표자 소개

- Toyshark 프로젝트 설립자
- LG전자, SK텔레콤에서 안드로이드 단말의 네트워크 관련 개발 및 분석

프로젝트 개요

- Android application
- 네트워크 데이터 실시간 모니터링 시스템
- IP, TCP, UDP
- https://github.com/LipiLee/toyshark.git

패킷 캡쳐 필요성

- 네트워크 전문가
- 보안 전문가
- Android 관련 개발자
- 기존 tcpdump명령어로 사용
- 루트 권한 필요
- 루팅이 잘 못 될 경우에는 벽돌로 전락

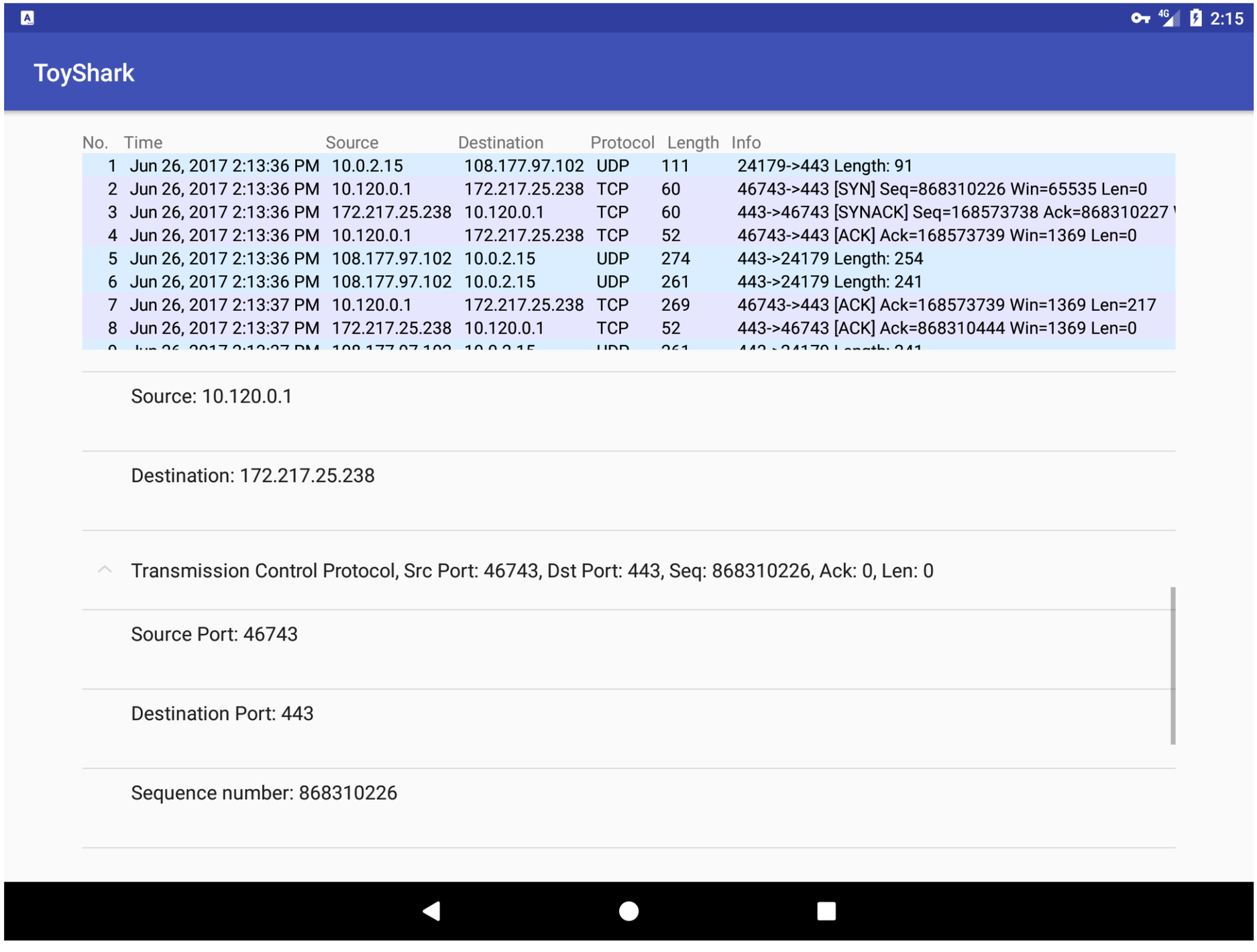
사용 기술

- Android VPN Framework 사용
- IceCream Sandwich 4.0
- API 14 이상
- 관련 샘플 코드
  https://android.googlesource.com/platform/development/+/master/samples/ToyVpn/

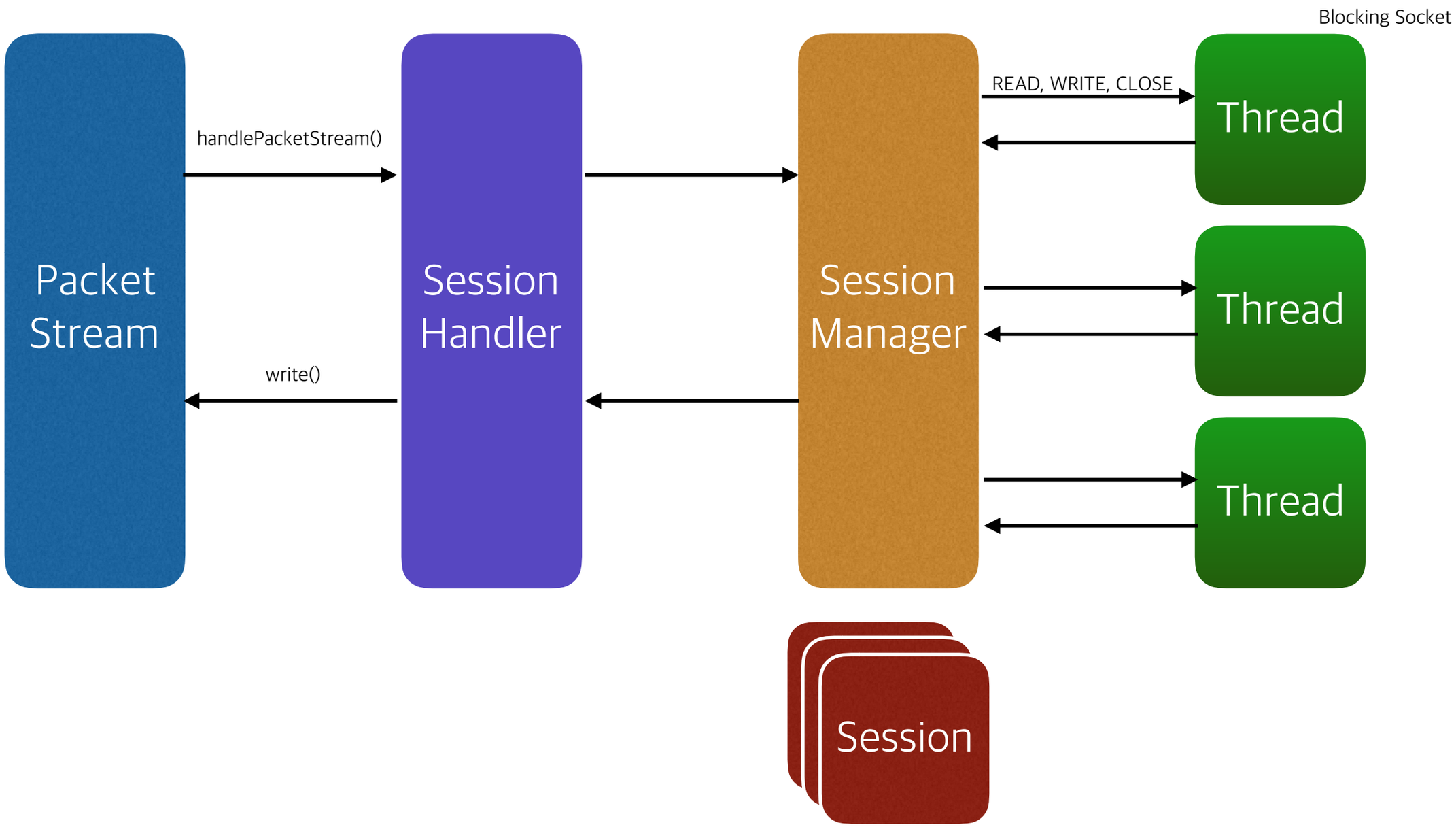
특징

- 처음 VPN기능 앱 사용시 사용자의 승인 필요
- VPN연결은 동시에 하나만 생성
- 시스템 노티가 항상 보임
- 시스템에서 관리되는 대화 상자가 보이고 연결 종료 가능

App UI


2:15
ToyShark
No. Time Source Destination Protocol Length Info
1 Jun 26, 2017 2:13:36 PM 10.0.2.15 108.177.97.102 UDP 111 24179->443 Length: 91
2 Jun 26, 2017 2:13:36 PM 10.120.0.1 172.217.25.238 TCP 60 46743->443 [SYN] Seq=868310226 Win=65535 Len=0
3 Jun 26, 2017 2:13:36 PM 172.217.25.238 10.120.0.1 TCP 60 443->46743 [SYNACK] Seq=168573738 Ack=868310227
4 Jun 26, 2017 2:13:36 PM 10.120.0.1 172.217.25.238 TCP 52 46743->443 [ACK] Ack=168573739 Win=1369 Len=0
5 Jun 26, 2017 2:13:36 PM 108.177.97.102 10.0.2.15 UDP 274 443->24179 Length: 254
6 Jun 26, 2017 2:13:36 PM 108.177.97.102 10.0.2.15 UDP 261 443->24179 Length: 241
7 Jun 26, 2017 2:13:37 PM 10.120.0.1 172.217.25.238 TCP 269 46743->443 [ACK] Ack=168573739 Win=1369 Len=217
8 Jun 26, 2017 2:13:37 PM 172.217.25.238 10.120.0.1 TCP 52 443->46743 [ACK] Ack=868310444 Win=1369 Len=0
Source: 10.120.0.1
Destination: 172.217.25.238
Transmission Control Protocol, Src Port: 46743, Dst Port: 443, Seq: 868310226, Ack: 0, Len: 0
Source Port: 46743
Destination Port: 443
Sequence number: 868310226

SW 구조


Blocking Socket
Packet Stream
handlePacketStream()
write()
Session Handler
Session Manager
READ, WRITE, CLOSE
Thread
Thread
Thread
Session

Kotlin

맛보기

```
data class Person(val name: String,
                  val age: Int? = null)

fun main(args: Array<String>) {
    val persons = listOf(Person("Alice"),
                         Person("Bob", age = 29))
    val oldest = persons.maxBy { it.age ?: 0 }
    println("The oldest is: $oldest")
}

// The oldest is: Person(name=Bob, age=29)
```

"data" class

Nullable type (Int?); the default value for the argument

Top-level function

Named argument

Lambda expression; Elvis operator

String template

Autogenerated toString

- Listing 1.1 An early taste of Kotlin in 'Kotlin in Action'

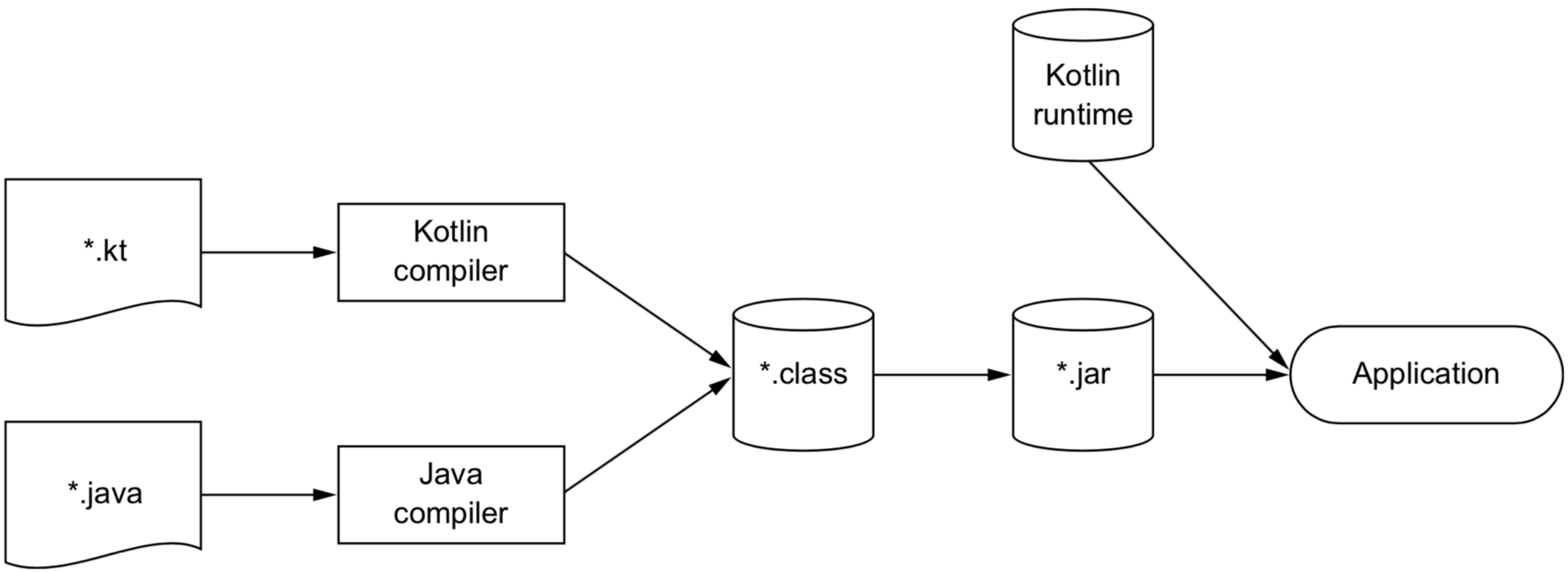
특징

- server-side, Android, iOS, anywhere Java runs
- Statically typed
- Functional and object-oriented
- Free and open source

철학

- Pragmatic
- Concise
- Safe
- Interoperable

Build Process



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP Header Format

https://tools.ietf.org/html/std7#page-15

Java Primitive

type	bit size	range	hex value
byte	8	$-128(-2^{7}) \sim 127(2^{7}-1)$	-128(0x80) -1(0xFF) 0(0x00) 127(0x7F)
short	16	$-32,768(-2^{15}) \sim 32,767(2^{15}-1)$	-32,768(0x8000) -1(0xFFFFF) 0(0x00) -32,767(0x7FFF)
int	32	$-2^{31} \sim 2^{31}-1$	-2^{31}(0x80000000) -1(0xFFFFFFFF) 0(0x00000000) $2^{31}-1$(0x7FFFFFFF)
long	64	$-2^{63} \sim 2^{63}-1$	-2^{63}(0x8000000000000000) -1(0xFFFFFFFFFFFFFFFF) 0(0x0000000000000000) $2^{63}-1$(0x7FFFFFFFFFFFFFFF)

Shift operation

```
jshell> byte aByte = -128
aByte ==> -128
```

0x80

```
jshell> byte anotherByte = aByte >> 4
```

0x08 or 0xF8

Shift operation

```
jshell> byte aByte = -128
aByte ==> -128

jshell> byte anotherByte = aByte >> 4
|  Error:
|  incompatible types: possible lossy conversion from int to byte
|  byte anotherByte = aByte >> 4;
|                     ^--------^
```

Shift operation

```
jshell> int aInteger = aByte >> 4
aInteger ==> -8
```

Shift operation

```
jshell> byte anotherByte = (byte) aByte >> 4
|  Error:
|  incompatible types: possible lossy conversion from int to byte
|  byte anotherByte = (byte) aByte >> 4;
|                     ^---------------^
```

Shift operation

```
jshell> byte anotherByte = (byte) (aByte >> 4)
anotherByte ==> -8
```

Implicit widening and explicit shortening

- byte aByte = -128; **// 0x80**
- int aInteger = aByte; **// 0xFFFFFFF80**
- int anotherInteger = aByte >> 4; **// 0xFFFFFFF8**
- byte anotherByte = (byte) (aByte >> 4); **// 0xF8**
- byte anotherByte = (byte) ((aByte >> 4) & 0xF) **// 0x08**

toUnsignedInt

```
public static int toUnsignedInt(byte x)
```

Converts the argument to an `int` by an unsigned conversion. In an unsigned conversion to an `int`, the high-order 24 bits of the `int` are zero and the low-order 8 bits are equal to the bits of the `byte` argument. Consequently, zero and positive `byte` values are mapped to a numerically equal `int` value and negative `byte` values are mapped to an `int` value equal to the input plus 2^8.

Parameters:

x - the value to convert to an unsigned int

Returns:

the argument converted to int by an unsigned conversion

Since:

1.8

Numbers

Type	Bit width
Double	64
Float	32
Long	64
Int	32
Short	16
Byte	8

- Kotlin에서는 Java의 primitive와 그 각각의 wrapper class와의 구별 없이 사용
- byte, java.lang.Byte
- int, java.lang.Integer

https://kotlinlang.org/docs/reference/basic-types.html

bitwise operation

Kotlin	Java
shl(shift left)	<<
shr(shift right)	>>
ushr(unsigned)	>>>
and	&
or	|
xor	^
inv	~

available for Int and Long only

https://kotlinlang.org/docs/reference/basic-types.html

bitwise operation(Kotlin)

```
>>> val aByte : Byte = -128
>>> val anotherByte = aByte shr 4
error: unresolved reference: shr
val anotherByte = aByte shr 4
                        ^
```

bitwise operation(Kotlin)

```
>>> val anotherByte: Byte = ((aByte.toInt() shr 4) and 0xF).toByte()
>>> println(anotherByte)
8
```

Non Blocking Mode

```
/**
 * Time between polling the VPN interface for new traffic, since it's non-blocking.
 *
 * TODO: really don't do this; a blocking read on another thread is much cleaner.
 */
private static final long IDLE_INTERVAL_MS = TimeUnit.MILLISECONDS.toMillis(100);
```

```
// If we are idle or waiting for the network, sleep for a
// fraction of time to avoid busy looping.
if (idle) {
    Thread.sleep(IDLE_INTERVAL_MS);
    final long timeNow = System.currentTimeMillis();
```

https://android.googlesource.com/platform/development/+/master/samples/ToyVpn/src/com/example/android/toyvpn/ToyVpnConnection.java

Change to Blocking Mode

```
private fun bringUpInterface() {
    val builder =  Builder()
            .addAddress( address: "192.168.0.1",  prefixLength: 32)
            .addRoute( address: "0.0.0.0",  prefixLength: 0)

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP)
        builder.setBlocking(true)

    vpnInterface = builder.establish()
}
```

https://github.com/LipiLee/ToyShark2/blob/master/app/src/main/java/com/lipisoft/toyshark/ToySharkService.kt

Blocking Mode

```
generic_x86:/ $ netstat -nt | grep CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:56336  ::ffff:172.217.24.202:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:44260  ::ffff:216.58.197.106:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:46006  ::ffff:216.58.200.14:44 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:44270  ::ffff:216.58.197.106:4 CLOSE_WAIT
generic_x86:/ $ netstat -nt | grep CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:56336  ::ffff:172.217.24.202:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:44265  ::ffff:216.58.197.106:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:56337  ::ffff:172.217.24.202:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:33213  ::ffff:172.217.25.10:44 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:49887  ::ffff:172.217.18.3:443 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:45993  ::ffff:216.58.200.14:44 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:46542  ::ffff:172.217.24.46:44 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:44260  ::ffff:216.58.197.106:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:46006  ::ffff:216.58.200.14:44 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:44270  ::ffff:216.58.197.106:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:44266  ::ffff:216.58.197.106:4 CLOSE_WAIT
tcp        1      0 ::ffff:10.0.2.15:46000  ::ffff:216.58.200.14:44 CLOSE_WAIT
```

Reference

- Kotlin in Action, Dmitry Jemerov and Svetlana Isakova
- Efficient Java 2nd, Joshua Bloch
- How “Effective Java” may have influenced the design of Kotlin, Lukas Lechner(https://hackernoon.com/how-effective-java-may-have-influenced-the-design-of-kotlin-part-1-45fd64c2f974)

QnA