**Лабораторная работа № 3.2**

**Разработка ICMP приложений**

Рассматривается задача разработки приложения посылающего ICMP пакеты на языке GO. Необходимо реализовать ping приложение. Для реализации возможно использовать один из следующих пакетов go-fastping (github.com/tatsushid/go-fastping) или go-ping (github.com/sparrc/go-ping). Дополнительно необходимо рассмотреть пакет trace (golang.org/x/net/trace) позволяющий выполнить трассировку до указанного хоста.

**Пример 1:** Посылаем хосту три ICMP пакета используя go-ping:

pinger, err := ping.NewPinger("[www.google.com](http://www.google.com/)")

if err != nil {

panic(err)

}

pinger.Count = 3

pinger.Run() // blocks until finished

stats := pinger.Statistics() // get send/receive/rtt stats

**Пример 2:** Моделирование Unix ping команды на основе go-ping:

pinger, err := ping.NewPinger("www.google.com")

if err != nil {

fmt.Printf("ERROR: %s\n", err.Error())

return

}

pinger.OnRecv = func(pkt \*ping.Packet) {

fmt.Printf("%d bytes from %s: icmp\_seq=%d time=%v\n",

pkt.Nbytes, pkt.IPAddr, pkt.Seq, pkt.Rtt)

}

pinger.OnFinish = func(stats \*ping.Statistics) {

fmt.Printf("\n--- %s ping statistics ---\n", stats.Addr)

fmt.Printf("%d packets transmitted, %d packets received, %v%% packet loss\n",

stats.PacketsSent, stats.PacketsRecv, stats.PacketLoss)

fmt.Printf("round-trip min/avg/max/stddev = %v/%v/%v/%v\n",

stats.MinRtt, stats.AvgRtt, stats.MaxRtt, stats.StdDevRtt)

}

fmt.Printf("PING %s (%s):\n", pinger.Addr(), pinger.IPAddr())

pinger.Run()

**Пример 3.** Моделирование Unix ping команды на основе go-fastping:

p := fastping.NewPinger()

ra, err := net.ResolveIPAddr("ip4:icmp", os.Args[1])

if err != nil {

fmt.Println(err)

os.Exit(1)

}

p.AddIPAddr(ra)

p.OnRecv = func(addr \*net.IPAddr, rtt time.Duration) {

fmt.Printf("IP Addr: %s receive, RTT: %v\n", addr.String(), rtt)

}

p.OnIdle = func() {

fmt.Println("finish")

}

err = p.Run()

if err != nil {

fmt.Println(err)

}

**Задача 1:** Реализовать приложение посылающее ICMP пакеты к заданному хосту и выводящее результаты ответа.

**Задача 2:** Реализовать вид DDoS-атаки типа ICMP-флуд используя горутины <http://golang-book.ru/chapter-10-concurrency.html>

**Задача 3.** Реализовать трассировку до заданного хоста.

**Замечание 1:** Для корректной работы GO на серверах 185.20.227.83, 185.20.226.174 необходимо задать переменную окружения *export GOPATH=~/go*.

**Источники информации**

1. <https://godoc.org/golang.org/x/net/trace>

2. <https://godoc.org/github.com/sparrc/go-ping>

3. <https://github.com/tatsushid/go-fastping>