

Добрый день и добро пожаловать в блог «Работа с сетью в Golang». Меня зовут Всеволод и сегодня мы рассмотрим. Для чего нам работать с сетью, для чего нам нужен этот блог? Как устроен Интернет, на каких уровнях мы будем работать? И как, собственно говоря, с сетью этой самой работать?

Погнали! Итак, что же мы, как специалисты DevOps-а либо системные администраторы, либо разработчики, делаем с сетью? С учетом того, что сейчас бизнесы в основном, построены в Интернете, то есть люди покупают товары в Интернете, люди в игры играют в Интернете, люди пользуются какими-то сервисами, услугами в Интернете, мы это все поддерживаем. Наша задача в 90% случаев – это поддержка серверов и сетевой инфраструктуры. И мы как специалисты DevOps-а, зачастую уже взаимодействуем с какими-то оболочками над этими серверами. То есть как правило, у нас есть железки, есть какие-то облачные провайдеры, есть какие-то сторонние провайдеры, у которых приложение крутится на наших машинах, и мы с ними работаем по сети.

Мы пишем... клиенты, если клиентов нет – мы пользуемся клиентами для сборки каких-то метрик либо каких-то данных с систем. Например, у нас там есть профайлер, с которого мы пишем нагрузки на процессор, или есть какой-то сетевой анализатор сетевых интерфейсов, если у вас машина занимается routing-ом. На нем надо тоже смотреть активность горячей точки.

Мы все это можем получить с помощью так называемой API – протокола обмена данными между серверами, одного из протоколов, и Go прекрасно подходит для написания клиентов для этого.

Мы можем анализировать сетевой трафик, особенно если мы работаем в каких-то частных сетях, в своих закрытых сетях и либо мы занимаемся какими-то сетевыми шейпингами, ставим сетевое оборудование. У нас какая-то машина есть,

и она собирает трафик, и нам этот трафик нужно как-то анализировать. Возможно, у вас есть какие-то специфические профили этого трафика – тоже необходимо с ними работать.

И конечно, писать свои сервера: сервера метрик, сервера мониторинга, сервера каких-то лайф проб; сервера, которые каким-то образом взаимодействуют, например, с Kubernetes-ом или сервера deploy-я приложений. В общем, применений очень много.

Итак, давайте вкратце рассмотрим, как работает Интернет. Я очень сомневаюсь, что вам нужно очень детальное погружение в модель, так называемую модель OSI сетевых уровней, но, тем не менее, не повредит ее вспомнить. Мне нравится думать об интернетовском трафике, как о луковице, то есть каждый уровень — это такая кожура. Посередине у нас самый высокий уровень, потом у нас над ним уровень чуть пониже, чуть пониже и так далее.

И соответственно, самый толстый и самая верхняя часть кожуры – это физический уровень – кодирование-декодирование сигнала, производящаяся уже сетевой картой или вашим Wi-Fi-адаптером согласно внутренним протоколам. То есть кодирование сигнала в радиоимпульс или в электрический сигнал по кабелю – нам это не очень интересно. Это, как правило, ответственность производителей сетевого оборудования: писать драйвера, делать (HP3 03:22) схемы и так далее – это нас не очень интересует.

Соответственно, после этого у нас есть физическая адресация, то есть адресация на конкретное какое-то устройство, имеющее MAC-адрес. MAC-адрес у нас уникальный для каждого устройства сетевого. И там есть свои протоколы, и нас они тоже особо не интересуют. Это протоколы, заложенные в driver-а switch-ей и

router-ов, то есть в основном, switch-ей. На самом деле, router-ы отвечают за несколько другой уровень и нам это не особо интересно.

А вот уровень, за который отвечают router-ы, то есть у router-а тоже есть MAC-адреса и всё остальное, но ответственность router-а – это уже найти другой router в сети Интернет. То есть вы делаете запрос по какому-то определенному IP-адресу, который точно идентифицирует компьютер или группу компьютеров внутри Интернета, где-то в Интернете. И соответственно, router отвечает за то, чтобы найти нужный port у себя выходной, найти нужный router и туда отправить этот пакет. И router-ов таких очень много, они образуют собой сеть Интернет. В общем, по IP-шнику вы можете найти другой компьютер в сети либо вычислить кого-то и соответственно, набить ему лицо.

4-й уровень, который уже нас интересует – это транспортный уровень, то есть это тот уровень, на котором у нас происходит уже запись данных. Соответственно это такие протоколы как TCP и UDP, есть еще протокол QUIC – Quick и куча еще, но, по сути, их назначение – это упаковать этот массив байтов и его передать по сети, то есть чтобы у вас другой компьютер получил этот массив байтов.

А вот какие это байты будут, и как он будет это расшифровывать – это уже ответственность следующих 3-х уровней: сессионного, презентационного и прикладного. Это всё обычно объединяется – эти все 3 верхних уровня в 1, потому что может такое быть, что у вас есть только сессионный уровень, или у вас есть только прикладной уровень, который всех все объединяет, либо у вас есть транспортный уровень поверх прикладного – такое тоже бывает. Полное нарушение модели OSI. Но смысл в том, что транспортный уровень вы пропустить не можете, а 1 из этих 3-х уровней вы точно можете пропустить.

Тем не менее, сессионный уровень – это уровень, на котором устанавливаются сессии между 2-мя клиентами для какого-то продолжительного обмена сообщениями. Дальше идет презентационный уровень, то есть у вас есть сессия, у вас внутри этой сессии есть какой-то набор пакетов через транспортный уровень, и эти пакеты что-то собой представляют. И эти пакеты могут иметь разные представления, то есть на прикладном уровне у вас одно и то же: какие-то данные, а кодируются они по-разному. Например, через безопасный протокол они шифруются, через какие-то другие протоколы они могут компрессироваться вместе, но на выходе у вас получится одно и то же: дешифрованное сообщение, декомпрессированное сообщение будут означать одно и то же для протокола более высокого уровня. Поэтому существует такая прослойка, абстракция – презентационный протокол, который отвечает за конвертацию этих сообщений.

И последний уровень – самый известный, самый часто применяемый. Тот, с которым будем, скорее всего, больше всего работать – это прикладной уровень. Это всеми нам известный протокол HTTP, Telnet, SMTP – это когда вы работаете с конкретными уже конечными клиентами. Как правило, это будет именно протокол прикладного уровня.

Мы разобрались более-менее, как работает Интернет, в общих чертах. Я рекомендую, если не очень понятно – прочитать об этом в Wikipedia-и, но скорее всего, вы уже про это знаете, поэтому много времени я здесь посвящать этому не буду. Поехали дальше.