

# UDP

---

МАГОМЕДОВ АЛИ

# UDP - ЭТО

---

один из ключевых элементов набора **сетевых протоколов** для **Интернета**. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые **датаграммами**) другим **хостам** по **IP-сети** без необходимости предварительного сообщения для установки специальных **каналов передачи** или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году

UDP использует простую модель передачи, без явных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа, но гарантируется, что если они придут, то в целостном состоянии.

# Более глубоко про UDP

---

UDP-приложения используют [датаграммные сокеты](#) для установки соединения между хостами. Приложение связывает сокет с его конечной точкой передачи данных, которая является комбинацией [IP-адреса](#) и порта службы. Порт — это программная структура, определяемая номером порта — 16-[битным](#) целочисленным значением (то есть от 0 до 65535)

UDP — минимальный ориентированный на обработку сообщений протокол [транспортного уровня](#).

# Структура пакета

---

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

Заголовок UDP состоит из четырёх полей, каждое по 2 байта (16 бит). Два из них необязательны к использованию в IPv4 (розовые ячейки в таблице), в то время как в IPv6 необязателен только порт отправителя.

# Сравнение UDP и TCP

TCP — ориентированный на соединение протокол, что означает необходимость «рукопожатия» для установки соединения между двумя хостами. Как только соединение установлено, пользователи могут отправлять данные в обоих направлениях.

UDP — более простой, основанный на сообщениях протокол без установления соединения.

Протоколы такого типа не устанавливают выделенного соединения между двумя хостами.

Связь достигается путём передачи информации в одном направлении от источника к получателю без проверки готовности или состояния получателя.

# TCP

---

- *Надёжность* — TCP управляет подтверждением, повторной передачей и тайм-аутом сообщений. Производятся многочисленные попытки доставить сообщение. Если оно потеряется на пути, сервер вновь запросит потерянную часть. В TCP нет ни пропавших данных, ни (в случае многочисленных тайм-аутов) разорванных соединений.
- *Упорядоченность* — если два сообщения последовательно отправлены, первое сообщение достигнет приложения-получателя первым. Если участки данных прибывают в неверном порядке, TCP отправляет неупорядоченные данные в буфер до тех пор, пока все данные не могут быть упорядочены и переданы приложению.
- *Тяжеловесность* — TCP необходимо три пакета для установки сокет-соединения перед тем, как отправить данные. TCP следит за надёжностью и перегрузками.
- *Потоковость* — данные читаются как поток [байтов](#), не передается никаких особых обозначений для границ сообщения или сегментов.

# UDP

- *Ненадёжный* — когда сообщение посылается, неизвестно, достигнет ли оно своего назначения — оно может потеряться по пути. Нет таких понятий, как подтверждение, повторная передача, тайм-аут.
- *Неупорядоченность* — если два сообщения отправлены одному получателю, то порядок их достижения цели не может быть предугадан.
- *Легковесность* — никакого упорядочивания сообщений, никакого отслеживания соединений и т. д. Это небольшой транспортный уровень, разработанный на IP.
- *Датаграммы* — пакеты посылаются по отдельности и проверяются на целостность только если они прибыли. Пакеты имеют определенные границы, которые соблюдаются после получения, то есть операция чтения на сокете-получателе выдаст сообщение целиком, каким оно было изначально послано.
- *Нет контроля перегрузок* — UDP сам по себе не избегает перегрузок. Для приложений с большой пропускной способностью возможно вызвать коллапс перегрузок, если только они не реализуют меры контроля на прикладном уровне.

# Приложения использующие UDP

---

Приложения, допускающие потери некоторого количества данных, которые будут едва заметны или незаметны вовсе.

Онлайн игры, Scype, Discord и т. д.

# Приложения использующие TSP

---

Приложения, для которых важен порядок и целостность данных.

Мессенджеры, Электронная почта и т. д.