

# ARP

*Contributors to Wikimedia projects*

У этого термина существуют и другие значения, см. [ARP \(значения\)](#).



Возможно, эта статья содержит [оригинальное исследование](#).

Проверьте соответствие информации приведённым источникам и удалите или исправьте информацию, являющуюся оригинальным исследованием. В случае необходимости подтвердите информацию авторитетными [источниками](#). В противном случае статья может быть выставлена на удаление. (9 февраля 2022)

ARP	
Название	Address Resolution Protocol
Уровень (по <a href="#">модели OSI</a> )	Канальный
Семейство	<a href="#">TCP/IP</a>
Создан в	<a href="#">1982</a>
Порт/ID	0x0806/ <a href="#">Ethernet</a>
Назначение протокола	Преобразование сетевых адресов в канальные
Спецификация	<a href="#">RFC 826</a>
Основные реализации (клиенты)	реализации стека TCP/IP в <a href="#">Microsoft Windows</a> , <a href="#">Linux</a> и <a href="#">BSD</a>
Основные реализации (серверы)	реализации стека TCP/IP в <a href="#">Windows</a> , <a href="#">Linux</a> и <a href="#">BSD</a>

 [Медиафайлы на Викискладе](#)

**ARP** ([англ.](#) Address Resolution Protocol — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения [MAC-адреса](#) другого компьютера по известному [IP-адресу](#).

Описание протокола было опубликовано в ноябре 1982 года в [RFC 826](#). ARP был спроектирован для передачи [IP-пакетов](#) через пакеты (кадры) Ethernet. Принцип выяснения аппаратного адреса целевого хоста, использованный в ARP, был затем использован в сетях других типов.

Наибольшее распространение ARP получил благодаря повсеместности сетей [IP](#), построенных поверх Ethernet, поскольку в них практически всегда используется ARP. В семействе протоколов [IPv6](#) ARP не существует, его функции возложены на [ICMPv6](#).

Рассмотрим суть функционирования *ARP* на простом примере. Компьютер А (IP-адрес 10.0.0.1) и компьютер Б (IP-адрес 10.22.22.2) соединены сетью [Ethernet](#). Компьютер А желает переслать пакет данных на компьютер Б, IP-адрес компьютера Б ему известен. Однако сеть Ethernet, которой они соединены, не работает с IP-адресами. Поэтому компьютеру А для осуществления передачи через Ethernet требуется узнать адрес

компьютера Б в сети Ethernet (*MAC-адрес* в терминах Ethernet). Для этой задачи и используется протокол ARP. По этому протоколу компьютер А отправляет широковещательный запрос, адресованный всем компьютерам в одном с ним [широковещательном домене](#). Суть запроса: «компьютер с IP-адресом 10.22.22.2, сообщите свой *MAC-адрес* компьютеру с *MAC-адресом* (напр. а0:ea:d1:11:f1:01)». Сеть Ethernet доставляет этот запрос всем устройствам в том же сегменте Ethernet, в том числе и компьютеру Б. Компьютер Б отвечает компьютеру А на запрос и сообщает свой *MAC-адрес* (напр. 00:ea:d1:11:f1:11). Теперь, получив *MAC-адрес* компьютера Б, компьютер А может передавать ему любые данные через сеть Ethernet.

Существуют следующие типы сообщений ARP: запрос ARP (*ARP request*) и ответ ARP (*ARP reply*). Система-отправитель при помощи запроса ARP запрашивает аппаратный адрес системы-получателя, который приходит внутри ответа ARP.

Перед тем, как передать пакет сетевого уровня через сегмент Ethernet, [сетевой стек](#) проверяет кэш ARP, чтобы выяснить, не зарегистрирована ли уже в его таблице нужная информация об узле-получателе. Если такой записи в кэше ARP нет, то выполняется широковещательный запрос ARP. Этот запрос для устройств в сети имеет следующий смысл: «Кто-нибудь знает физический адрес устройства, обладающего таким-то адресом IP?» Когда хост с этим адресом IP примет такой пакет запроса, он должен ответить: «Да, это мой адрес IP и мой аппаратный адрес такой-то». После этого отправитель запроса сохранит аппаратный адрес получателя в свой кэш ARP и сможет адресно передать информацию получателю.

Ниже приведён пример запроса и ответа ARP. <см. внизу страницы>

Записи в кэше ARP могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кэша. Можно также создавать статические записи в таблице. В большинстве операционных систем это можно сделать при помощи команды:

```
arp -s <IP-адрес> <MAC-адрес>
```

В Windows Server 2003 записи в таблице ARP, созданные динамически, остаются в кэше в течение 2 минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кэше продлевается ещё на 2 минуты. Эта процедура может повторяться несколько раз, но максимум запись в кэше просуществует до 10 минут. После этого запись будет удалена из кэша, и при необходимости будет отправлен повторный запрос ARP<sup>[1]</sup>.

В более новых операционных системах время хранения записей в таблице ARP и метод хранения выбираются программно и при желании их можно изменить.

ARP изначально был разработан не только для IP, этот протокол также можно использовать для выяснения адресов MAC в различных протоколах 3 уровня ([англ.](#) Layer 3 protocols addresses). ARP был адаптирован также для получения других (аппаратных) адресов 2 уровня модели OSI (Layer 2 addresses).

Протоколы InARP и ATM ARP используются в разных вариантах инкапсуляции IP over ATM, описанных в [RFC 1577](#) (Classical IP and ARP over ATM)<sup>[2]</sup>.

В настоящее время ARP в основном используется для сопоставления адресов IP и MAC в сетях ethernet.

***Inverse Address Resolution Protocol, Inverse ARP*** или ***InARP*** — протокол для

получения адресов сетевого уровня (например [адресов IP](#)) других рабочих станций по их адресам канального уровня (например, [DLCI](#) в сетях [Frame Relay](#)). *InARP* обычно используется в сетях [Frame Relay](#) и [ATM](#).

*ARP* переводит адреса сетевого уровня в адреса канального уровня, в то же время *InARP* можно рассматривать как его инверсию. *InARP* реализовано как расширение *ARP*. Форматы пакетов этих протоколов одни и те же, различаются лишь коды операций и заполняемые поля.

[Reverse Address Resolution Protocol](#), *Reverse ARP* или *RARP*, как и *InARP*, переводит адреса канального уровня в адреса сетевого уровня. Но *RARP* используется для получения логических адресов самих станций отправителей, в то время как в *InARP*-протоколе отправитель знает свои адреса и запрашивает логический адрес другой станции. От *RARP* отказались в пользу [BOOTP](#), который был в свою очередь заменён [DHCP](#).

1. Узел, которому нужно выполнить отображение адреса IP на аппаратный адрес (Ethernet hardware address, [MAC-адрес](#)), формирует запрос *ARP* с адресом IP получателя, вкладывает его в кадр протокола канального уровня и рассылает его широковещательно.
2. Все узлы сегмента локальной сети получают запрос *ARP* и сравнивают указанный там адрес IP с собственным.
3. В случае совпадения собственного адреса IP с полученным в запросе *ARP*, узел формирует ответ *ARP*, в котором указывает и свой адрес IP, и свой аппаратный адрес, и отправляет его уже адресно на аппаратный адрес отправителя запроса *ARP*.

Преобразование адресов выполняется путём поиска в таблице соответствия адресов IP и MAC. Эта таблица, называемая таблицей *ARP*, хранится в памяти операционной системы и содержит записи для каждого известного ей узла сети. В двух столбцах содержатся адреса IP и Ethernet (MAC). Если требуется преобразовать адрес IP в MAC, то в таблице *ARP* ищется запись с соответствующим адресом IP.

Упрощённый пример  
таблицы *ARP*

```
223.1.2.1 08:00:39:00:2F:C3
223.1.2.3 08:00:5A:21:A7:22
223.1.2.4 08:00:10:99:AC:54
```

Ниже проиллюстрирована структура пакета, используемого в запросах и ответах *ARP*. В сетях [Ethernet](#) в этих пакетах используется [EtherType](#) 0x0806, и запросы рассылаются на широковещательный [MAC-адрес](#) — FF:FF:FF:FF:FF:FF. Отметим, что в структуре пакета, показанной ниже, в качестве SHA, SPA, THA и TPA условно используются 32-[битные](#) слова — реальная длина определяется физическим устройством и протоколом.

+	Bits 0 — 7	8 — 15	16 — 31
0	Hardware type (HTYPE)		Protocol type (PTYPE)
32	Hardware length (HLEN)	Protocol length (PLEN)	Operation (OPER)
64	Sender hardware address (SHA)		
?	Sender protocol address (SPA)		
?	Target hardware address (THA)		
?	Target protocol address (TPA)		

Hardware type (HTYPE)

Каждый канальный протокол передачи данных имеет свой номер, который хранится в этом поле. Например, [Ethernet](#) имеет номер 0x0001.

Protocol type (PTYPE)

Код сетевого протокола. Например, для [IPv4](#) будет записано 0x0800.

Hardware length (HLEN)

Длина физического адреса в байтах. Адреса Ethernet имеют длину 6 байт (0x06).

Protocol length (PLEN)

Длина логического адреса в байтах. IPv4 адреса имеют длину 4 байта (0x04).

Operation

Код операции отправителя: 0x0001 в случае запроса и 0x0002 в случае ответа.

Sender hardware address (SHA)

Физический адрес отправителя.

Sender protocol address (SPA)

Логический адрес отправителя.

Target hardware address (THA)

Физический адрес получателя. Не требуется при запросе.

Target protocol address (TPA)

Логический адрес получателя.

14 15.032668 HewlettP\_86:59:e5 Broadcast ARP Who has 192.168.168.7? Tell

▾ Frame 14 (42 bytes on wire, 42 bytes captured)

Arrival Time: Sep 29, 2006 11:11:38.229643000  
 [Time delta from previous packet: 0.323763000 seconds]  
 [Time since reference or first frame: 15.032668000 seconds]  
 Frame Number: 14  
 Packet Length: 42 bytes  
 Capture Length: 42 bytes  
 [Protocols in frame: eth:arp]  
 [Coloring Rule Name: ARP]  
 [Coloring Rule String: arp]

▾ Ethernet II, Src: HewlettP\_86:59:e5 (00:0d:9d:86:59:e5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

▾ Destination: Broadcast (ff:ff:ff:ff:ff:ff)  
     Address: Broadcast (ff:ff:ff:ff:ff:ff)  
     .... 1. .... = Multicast: This is a MULTICAST frame  
     .... 1. .... = Locally Administrated Address: This is NOT a f.

▾ Source: HewlettP\_86:59:e5 (00:0d:9d:86:59:e5)  
     Address: HewlettP\_86:59:e5 (00:0d:9d:86:59:e5)  
     .... 0. .... = Multicast: This is a UNICAST frame  
     .... 0. .... = Locally Administrated Address: This is a FACTO  
     Type: ARP (0x0806)

▾ Address Resolution Protocol (request)

Hardware type: Ethernet (0x0001)  
 Protocol type: IP (0x0800)  
 Hardware size: 6  
 Protocol size: 4  
 Opcode: request (0x0001)  
 Sender MAC address: HewlettP\_86:59:e5 (00:0d:9d:86:59:e5)  
 Sender IP address: 192.168.168.101 (192.168.168.101)  
 Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
 Target IP address: 192.168.168.7 (192.168.168.7)



Если хост с IPv4 адресом 10.10.10.123 и MAC адресом 00:0D:9D:86:59:E2 хочет послать пакет другому хосту с адресом 10.10.10.140, но не знает его MAC адрес, то он должен послать ARP запрос для разрешения адреса.

Пакет, изображённый ниже, изображает широковещательный запрос. Если хост с IP 10.10.10.140 присутствует в сети и доступен, то он получает этот запрос ARP и возвращает ответ.

Bits 0 — 7	8 — 15	16 — 31
0 Hardware type = 0x0001		Protocol type = 0x0800
32 Hardware length = 0x06	Protocol length = 0x04	Operation = 0x0001
64 SHA (first 32 bits) = 0x000D9D86		
96 SHA (last 16 bits) = 0x59E2		SPA (first 16 bits) = 0x0A0A

**128** SPA (last 16 bits) = 0x0A7B

THA (first 16 bits) = 0x0000

**160** THA (last 32 bits) = 0x00000000

**192** TPA = 0x0A0A0A8C

В ситуации, описанной выше, если узел с адресом 10.10.10.140 имеет адрес MAC 00:09:58:D8:33:AA, то он отправит в ответ пакет, проиллюстрированный ниже. Заметим, что блоки адресов отправителя и получателя теперь поменяли значения (отправитель ответа теперь получатель запроса; получатель ответа — отправитель запроса). Кроме того, в ответе есть MAC-адрес узла 10.10.10.140 в поле физического адреса отправителя (SHA), а поле THA не пустое (одноадресный ответ).

Любой узел в той же сети, что и отправитель с получателем, тоже получит запрос (так как он широковещательный) и таким образом добавит в свой кэш информацию об отправителе. Ответ ARP направлен только источнику запроса ARP, поэтому ответ ARP не доступен другим узлам в сети.

+	Bits 0 — 7	8 — 15	16 — 31
<b>0</b>	Hardware type = 0x0001		Protocol type = 0x0800
<b>32</b>	Hardware length = 0x06	Protocol length = 0x04	Operation = 0x0002
<b>64</b>	SHA (first 32 bits) = 0x000958D8		
<b>96</b>	SHA (last 16 bits) = 0x33AA		SPA (first 16 bits) = 0x0A0A
<b>128</b>	SPA (last 16 bits) = 0x0A8C		THA (first 16 bits) = 0x000D
<b>160</b>	THA (last 32 bits) = 0x9D8659E2		
<b>192</b>	TPA = 0x0A0A0A7B		

- **Замечание.** Длина полей SHA, SPA, THA, TPA зависит от параметров Hardware length и Protocol length соответственно.

Эффективность функционирования ARP во многом зависит от кэша ARP (*ARP cache*), который имеется на каждом хосте. В кэше содержится составленная операционной системой таблица соответствия адресов MAC и IP.

Время жизни записи в кэше оставлено на усмотрение разработчика. По умолчанию может составлять от десятков секунд (например, 20 секунд) до четырёх часов ([Cisco IOS](#)).<sup>[3]</sup>

ARP может использоваться для обнаружения конфликтов IP-адресов в локальной сети. [RFC 5227](#) определяет формат запроса *ARP Probe* с полем SPA, состоящим из всех нулей (адрес IP 0.0.0.0). Перед назначением адреса IP интерфейсу хост может проверить, что этот адрес не используется другим хостом сегмента локальной сети.

ARP-оповещение (ARP Announcement) — это пакет (обычно ARP-запрос<sup>[4]</sup>), содержащий корректную SHA и SPA хоста-отправителя, с TPA, равной SPA. Это не разрешающий запрос, а запрос на обновление кэша ARP других хостов, получающих пакет.

Большинство операционных систем посылает такой пакет при включении хоста в сеть, что позволяет предотвратить ряд проблем. Например, при смене сетевой карты (когда необходимо обновить связь между адресами IP и MAC), такой запрос исправит записи в кэше ARP других хостов в сети.

ARP-оповещения также используются для «защиты» адресов IP в протоколе [Zeroconf](#), описанном в [RFC 3927](#).



Специальный случай запроса ARP — запрос собственного адреса IP, он получил название «*Gratuitous ARP*» (добровольный запрос ARP)<sup>[5]</sup>.

В таком запросе адреса IP отправителя и получателя совпадают.

*Gratuitous ARP* используется в двух целях<sup>[5]</sup>:

1. оповещение соседних устройств о том, что в сегменте сети появился новый адрес IP;
2. проверка свободности адреса IP (используется ли он другим устройством).

- [MAC-адрес](#)
- [IP-адрес](#)
- [DHCP](#)
- [RARP](#)
- [BOOTP](#)
- [ARP-spoofing](#)
- [Proxy ARP](#)

1. ↑ [View the Address Resolution Protocol \(ARP\) cache](#) : [англ.] : [apx. 25 февраля 2021] // MSDN. — 2009. — 8 October.
2. ↑ [TCP/IP over ATM](#) : [англ.] : [apx. 9 февраля 2022]. — IBM.
3. ↑ [Embedded System Testing Blog: ARP Timeout Value for Linux, Windows, Cisco 2960 and DELL Switch](#). Дата обращения: 8 ноября 2013. [Архивировано](#) 21 сентября 2013 года.
4. ↑ [Re: \[dhcwg\] Gratuitous ARP in DHCP vs. IPv4 ACD Draft](#) [Архивировано](#) 12 октября 2007 года.
5. ↑ [Перейти обратно: <sup>1</sup> <sup>2</sup> ZvonDozvon](#).

- (англ.) [RFC 826](#) : Address Resolution Protocol
- (англ.) [RFC 1577](#) : Classical IP and ARP over ATM
- (англ.) [RFC 2390](#) : Inverse Address Resolution Protocol
- (англ.) [RFC 5227](#) : IPv4 Address Conflict Detection
- (англ.) [Gratuitous ARP](#)
- (англ.) [ARP Sequence Diagram](#) // EventStudio 1.0 documentation. — [Архивная копия](#) от 10 ноября 2006 на [Wayback Machine](#)
- (фр.) [Free ARP tools with source code](#)
- (англ.) [ARP-SK ARP traffic generation tools](#)
- [ARP-спуфинг](#)
- (англ.) [How To Clear ARP Cache On Linux or Unix](#)
- [Протокол ARP — протокол разрешения адресов, принцип работы](#) // ZvonDozvon.ru.
- [Правила чтения таблиц, описывающих структуру заголовков сетевых протоколов](#)