**SIP** ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Session Initiation Protocol* — протокол установления сеанса) — [протокол передачи данных](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), который описывает способ установления и завершения пользовательского интернет-сеанса, включающего обмен [мультимедийным](http://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0) содержимым ([видео-](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%BA%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F) и [аудиоконференция](http://ru.wikipedia.org/wiki/%D0%90%D1%83%D0%B4%D0%B8%D0%BE%D0%BA%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F), [мгновенные сообщения](http://ru.wikipedia.org/wiki/%D0%9C%D0%B3%D0%BD%D0%BE%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F), [онлайн-игры](http://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%B8%D0%B3%D1%80%D1%8B)).

В [модели взаимодействия открытых систем](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI) SIP является [сетевым протоколом](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C).

Протокол описывает, каким образом клиентское приложение (например, [софтфон](http://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%84%D0%BE%D0%BD)) может запросить начало соединения у другого, возможно, физически удалённого клиента, находящегося в той же сети, используя его уникальное имя. Протокол определяет способ согласования между клиентами об открытии каналов обмена на основе других протоколов, которые могут использоваться для непосредственной передачи информации (например, [RTP](http://ru.wikipedia.org/wiki/RTP)). Допускается добавление или удаление таких каналов в течение установленного сеанса, а также подключение и отключение дополнительных клиентов (то есть допускается участие в обмене более двух сторон — конференц-связь). Протокол также определяет порядок завершения сеанса.

Принципы протокола

Разработкой занималась организация [IETF](http://ru.wikipedia.org/wiki/IETF) MMUSIC Working Group[[1]](http://ru.wikipedia.org/wiki/SIP#cite_note-1). Протокол начал разрабатываться в 1996 году Хенингом Шулзри (Henning Schulzrinne, [Колумбийский университет](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D1%83%D0%BC%D0%B1%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82)) и Марком Хэндли ([Университетский колледж Лондона](http://ru.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BB%D0%BB%D0%B5%D0%B4%D0%B6_%D0%9B%D0%BE%D0%BD%D0%B4%D0%BE%D0%BD%D0%B0)). В ноябре 2000 года SIP был утверждён как [сигнальный протокол](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) проекта [3GPP](http://en.wikipedia.org/wiki/3GPP) и основной протокол архитектуры [IMS](http://ru.wikipedia.org/wiki/IMS_(%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%81%D0%B2%D1%8F%D0%B7%D1%8C))(модификация 3GPP TS.24.229[[2]](http://ru.wikipedia.org/wiki/SIP#cite_note-2))[[3]](http://ru.wikipedia.org/wiki/SIP#cite_note-3). Наряду c другим распространённым протоколом [H.323](http://ru.wikipedia.org/wiki/H.323), SIP — один из протоколов, лежащих в основе [Voice over IP](http://ru.wikipedia.org/wiki/Voice_over_IP).

В основу протокола рабочая группа MMUSIC заложила следующие принципы:

* *Простота*: включает в себя только шесть методов (функций)
* *Независимость* от транспортного уровня, может использовать [UDP](http://ru.wikipedia.org/wiki/UDP), [TCP](http://ru.wikipedia.org/wiki/TCP), [ATM](http://ru.wikipedia.org/wiki/ATM) и т. д.
* *Персональная мобильность пользователей*. Пользователи могут перемещаться в пределах сети без ограничений. Это достигается путем присвоения пользователю уникального идентификатора. При этом набор предоставляемых услуг остается неизменным. О своих перемещениях пользователь сообщает с помощью сообщения **REGISTER**.
* *Масштабируемость сети*. Структура сети на базе протокола SIP позволяет легко ее расширять и увеличивать число элементов.
* *Расширяемость протокола*. Протокол характеризуется возможностью дополнять его новыми функциями при появлении новых услуг.
* *Интеграция в стек существующих протоколов*[*Интернет*](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Протокол SIP является частью глобальной архитектуры мультимедиа, разработанной комитетом [IETF](http://ru.wikipedia.org/wiki/IETF). Кроме SIP, эта архитектура включает в себя протоколы [RSVP](http://ru.wikipedia.org/wiki/RSVP_(%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)), [RTP](http://ru.wikipedia.org/wiki/RTP), [RTSP](http://ru.wikipedia.org/wiki/RTSP), [SDP](http://ru.wikipedia.org/wiki/SDP).
* *Взаимодействие с другими протоколами сигнализации*. Протокол SIP может быть использован совместно с другими протоколами [IP-телефонии](http://ru.wikipedia.org/wiki/VoIP), протоколами [ТфОП](http://ru.wikipedia.org/wiki/%D0%A2%D1%84%D0%9E%D0%9F), и для связи с интеллектуальными сетями.

**Архитектура сети SIP**

В некотором смысле прародителем протокола SIP является протокол переноса гипертекста - HTTP (Hypertext Transfer Protocol, RFC 2068). Протокол SIP унаследовал от него синтаксис и архитектуру <клиент-сервер>, которую иллюстрирует рис. 2.

   
**Рис. 2 Архитектура "клиент-сервер"**

Клиент выдает запросы, в которых указывает, что он желает получить от сервера. Сервер принимает запрос, обрабатывает его и выдает ответ, который может содержать уведомление об успешном выполнении запроса, уведомление об ошибке или информацию, затребованную клиентом.

Управление процессом обслуживания вызова распределено между разными элементами сети SIP. Основным функциональным элементом, реализующим функции управления соединением, является терминал. Остальные элементы сети отвечают за маршрутизацию вызовов, а в некоторых случаях предоставляют дополнительные услуги.

**Терминал**

В случае, когда клиент и сервер взаимодействуют непосредственно с пользователем (т.е. реализованы в оконечном оборудовании пользователя), они называются, соответственно, клиентом агента пользователя - User Agent Client (UAC) - и сервером агента пользователя - User Agent Server (UAS).

Следует особо отметить, что сервер UAS и клиент UAC могут (но не обязаны) непосредственно взаимодействовать с пользователем, а другие клиенты и серверы SIP этого делать не могут. Если в устройстве присутствуют и сервер UAS, и клиент UAC, то оно называется агентом пользователя - User Agent (UA), а по своей сути представляет собой терминальное оборудование SIP.

Кроме терминалов определены два основных типа сетевых элементов SIP: прокси-сервер (proxy server) и сервер переадресации (redirect server).

**Прокси-сервер**

Прокси-сервер (от английского proxy - представитель) представляет интересы пользователя в сети. Он принимает запросы, обрабатывает их и, в зависимости от типа запроса, выполняет определенные действия. Это может быть поиск и вызов пользователя, маршрутизация запроса, предоставление услуг и т.д. Прокси-сервер состоит из клиентской и серверной частей, поэтому может принимать вызовы, инициировать собственные запросы и возвращать ответы. Прокси - сервер может быть физически совмещен с сервером определения местоположения (в этом случае он называется registrar) или существовать отдельно от этого сервера, но иметь возможность взаимодействовать с ним по протоколам LDAP (RFC 1777), rwhois (RFC 2167) и по любым другим протоколам.

Предусмотрено два типа прокси-серверов - с сохранением состояний (stateful) и без сохранения состояний (stateless).

Сервер первого типа хранит в памяти входящий запрос, который явился причиной генерации одного или нескольких исходящих запросов. Эти исходящие запросы сервер также запоминает Все запросы хранятся в памяти сервера только до окончания транзакции, т.е. до получения ответов на запросы.

Сервер первого типа позволяет предоставить большее количество услуг, но работает медленнее, чем сервер второго типа. Он может применяться для обслуживания небольшого количества клиентов, например, в локальной сети. Прокси-сервер должен сохранять информацию о состояниях, если он:

- использует протокол TCP для передачи сигнальной информации;   
- работает в режиме многоадресной рассылки сигнальной информации;   
- размножает запросы.

Последний случай имеет место, когда прокси-сервер ведет поиск вызываемого пользователя сразу в нескольких направлениях, т.е. один запрос, который пришел к прокси-серверу, размножается и передается одновременно по всем этим направлениям.

Сервер без сохранения состояний просто ретранслирует запросы и ответы, которые получает. Он работает быстрее, чем сервер первого типа, так как ресурс процессора не тратится на запоминание состояний, вследствие чего сервер этого типа может обслужить большее количество пользователей. Недостатком такого сервера является то, что на его базе можно реализовать лишь наиболее простые услуги. Впрочем, прокси-сервер может функционировать как сервер с сохранением состояний для одних пользователей и как сервер без сохранения состояний - для других.

Алгоритм работы пользователей с прокси-сервером выглядит следующим образом. Поставщик услуг IP-телефонии сообщает адpec прокси-сервера своим пользователям. Вызывающий пользователь передает к прокси-серверу запрос соединения. Сервер обрабатывает запрос, определяет местоположение вызываемого пользователя и передает запрос этому пользователю, а затем получает от него ответ, подтверждающий успешную обработку запроса, и транслирует этот ответ пользователю, передавшему запрос. Прокси-сервер может модифицировать некоторые заголовки сообщений, которые он транслирует, причем каждый сервер, обработавший запрос в процессе его передачи от источника к приемнику, должен указать это в SIP-запросе для того, чтобы ответ на запрос вернулся по такому же пути.

**Сервер переадресации**

Сервер переадресации предназначен для определения текущего адреса вызываемого пользователя. Вызывающий пользователь передает к серверу сообщение с известным ему адресом вызываемого пользователя, а сервер обеспечивает переадресацию вызова на текущий адрес этого пользователя. Для реализации этой функции сервер переадресации должен взаимодействовать с сервером определения местоположения.

Сервер переадресации не терминирует вызовы как сервер RAS и не инициирует собственные запросы как прокси-сервер. Он только сообщает адрес либо вызываемого пользователя, либо прокси-сервера. По этому адресу инициатор запроса передает новый запрос. Сервер переадресации не содержит клиентскую часть программного обеспечения.

Но пользователю не обязательно связываться с каким-либо SIP-сервером. Он может сам вызвать другого пользователя при условии, что знает его текущий адрес.

**Сервер определения местоположения пользователей**

Пользователь может перемещаться в пределах сети, поэтому необходим механизм определения его местоположения в текущий момент времени. Например, сотрудник предприятия уезжает в командировку, и все вызовы, адресованные ему, должны быть направлены в другой город на его временное место работы. О том, где он находится, пользователь информирует специальный сервер с помощью сообщения REGISTER. Возможны два режима регистрации: пользователь может сообщить свой новый адрес один раз, а может регистрироваться периодически через определенные промежутки времени. Первый способ подходит для случая, когда терминал, доступный пользователю, включен постоянно, и его не перемещают по сети, а второй - если терминал часто перемещается или выключается.

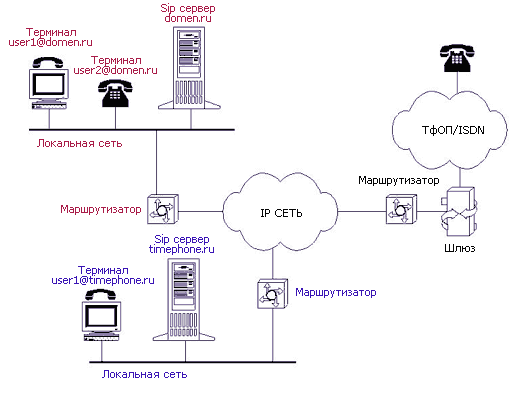
Для хранения текущего адреса пользователя служит сервер определения местоположения пользователей, представляющий собой базу данных адресной информации. Кроме постоянного адреса пользователя, в этой базе данных может храниться один или несколько текущих адресов.

Этот сервер может быть совмещен с прокси-сервером (в таком случае он называется registrar) или быть реализован отдельно от прокси-сервера, но иметь возможность связываться с ним.

В RFC 2543 сервер определения местоположения представлен как отдельный сетевой элемент, но принципы его работы в этом документе не регламентированы. Стоит обратить внимание на то, что вызывающий пользователь, которому нужен текущий адрес вызываемого пользователя, не связывается с сервером определения местоположения напрямую. Эту функцию выполняют SIP-серверы при помощи протоколов LDAP (RFC 1777), rwhois (RFC 2167), или других протоколов.

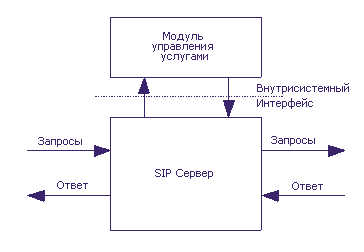
**Пример SIP- сети**

Резюмируя все сказанное выше, отметим, что сети SIP строятся из элементов трех основных типов: терминалов, прокси-серверов и серверов переадресации. На рис. 3 приведен пример возможного построения сети SIP.

   
**Рис. 3 Пример построения сети SIP**

Стоит обратить внимание на то что, что SIP-серверы, представленные на рис. 3, являются отдельными функциональными сетевыми элементами. Физически они могут быть реализованы на базе серверов локальной сети, которые, помимо выполнения своих основных функций, будут также обрабатывать SIP-сообщения. Терминалы же могут быть двух типов: персональный компьютер со звуковой платой и программным обеспечением SIP-клиента (UA) или SIP-телефон, подключающийся не посредственно к ЛВС Ethernet (SIР-телефоны, производимые компанией Cisco Systems, недавно появились на российском рынке). Таким образом, пользователь локальной вычислительной сети передает все запросы к своему SlP-серверу, а тот обрабатывает их и обеспечивает установление соединений. Путем программирования сервер можно застроить на разные алгоритмы работы: он может обслуживать часть пользователей (например, руководство предприятия или особо важных лиц) по одним правилам, а другую часть - по иным. Возможно также, что сервер будет учитывать категорию и срочность вызовов, а также вести начисление платы за разговоры.

Структурная схема организации услуг SIP-сервера представлена на рисунке 4.

   
**Рис. 4 Структурная схема организации услуг SIP-сервера**

Модуль управления услугами отвечает за предоставление услуг и за общее управление сервером. Принятые сервером запросы и ответы поступают в модуль управления услугами и обрабатываются им, на основании чего определяется реакция на полученные сообщения. Интерфейс человек-машина позволяет гибко менять настройки сервера и вести мониторинг сети.

### Термина

Когда клиент и сервер реализованы в оконечном оборудовании и взаимодействуют непосредственно с пользователем, они называются *пользовательским агентским клиентом* — User Agent Client (UAC) — и *пользовательским агентским сервером* — User Agent Server (UAS). Если в устройстве присутствуют и UAC, и UAS, то оно называется пользовательским агентом — User Agent (UA), а по своей сути представляет собой терминальное оборудование SIP.

Сервер UAS и клиент UAC имеют возможность непосредственно взаимодействовать с пользователем. Другие клиенты и серверы SIP этого делать не могут.

### [Прокси-сервер](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BA%D1%81%D0%B8-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)

Прокси-сервер (от [англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *proxy* — «представитель») представляет интересы пользователя в сети. Он принимает запросы, обрабатывает их и выполняет соответствующие действия. Прокси-сервер состоит из клиентской и серверной частей, поэтому может принимать вызовы, инициировать запросы и возвращать ответы. Прокси-сервер может не изменять структуру и содержимое передаваемых сообщений, лишь добавляя свою адресную информацию в специальное поле Via.

Предусмотрено два типа прокси-серверов

* с сохранением состояний (stateful). Такой сервер хранит в своей памяти все полученные запросы и связанные с ним новые сформированные запросы до окончания [транзакции](http://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D0%B7%D0%B0%D0%BA%D1%86%D0%B8%D1%8F_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)).
* без сохранения состояний(stateless). Такой сервер просто обрабатывает получаемые запросы. Но на его базе нельзя реализовать сложные, интеллектуальные услуги.

### Сервер переадресации

Сервер переадресации используется для определения текущего местоположения пользователя. Сервер переадресации не терминирует вызовы и не инициирует собственные запросы, а только сообщает адрес необходимого терминала или прокси-сервера. Для этих целей он взаимодействует с сервером определения местоположения.

Однако, для осуществления соединения пользователь может не использовать сервер переадресации, если он сам знает текущий адрес требуемого пользователя.

### Сервер определения местоположения пользователей

Пользователь может перемещаться в пределах сети SIP, поэтому существует механизм определения его местоположения в текущий момент времени. Сервер определения местоположения пользователей служит для хранения текущего адреса пользователя и представляет собой базу данных адресной информации. В качестве источника информации о местоположении пользователя, может выступать даже устройство или сервер, не работающее по протоколу SIP (например [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94), [MS Exchange](http://ru.wikipedia.org/wiki/Microsoft_Exchange_Server), [RADIUS](http://ru.wikipedia.org/wiki/RADIUS)-сервер и т.п.).

Пользователь, которому нужна адресная информация не связывается с сервером определения местоположения напрямую. Эту функцию выполняют другие SIP-серверы при помощи протоколов [LDAP](http://ru.wikipedia.org/wiki/LDAP), R[WHOIS](http://ru.wikipedia.org/wiki/WHOIS), или других протоколов.

Сервер B2BUA

B2BUA — (англ. *back-to-back user agent*, буквально: *пользовательский агент спина-к-спине*) — вариант серверного логического элемента в приложениях, работающих с протоколом SIP. По идеологии работы, B2BUA похож на [прокси-сервер SIP](http://ru.wikipedia.org/wiki/SIP#.D0.9F.D1.80.D0.BE.D0.BA.D1.81.D0.B8-.D1.81.D0.B5.D1.80.D0.B2.D0.B5.D1.80), однако есть принципиальные различия. Сервер B2BUA, работает одновременно с несколькими (как правило, двумя) оконечными устройствами — терминалами, разделяя вызов или сеанс на разные плечи-участки. С каждым участком B2BUA работает индивидуально, как UAS по отношению к инициатору и как UAC по отношению к терминалу, принимающему вызов. При этом сигнальные сообщения передаются в рамках сеанса в обе стороны синхронно. Таким образом, каждый из участников соединения (сеанса связи), на уровне сигнализации взаимодействует с B2BUA, как с оконечным устройством, хотя в действительности, сервер является посредником. Это отражается в адресных полях (таких как From, To и Contact) сообщений, отправляемых сервером B2BUA. Таким образом, ключевое отличие B2BUA - полностью независимая сигнализация всех участков вызова. Это означает, в частности, что для взаимодействия с каждым отдельным пользователем в рамках сеанса связи используются уникальные идентификаторы, а содержимое одних и тех же сообщений для разных участков будет различным.

Сервер B2BUA может предоставлять следующие функции:

* Управление звонками ([биллинг](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BB%D0%BB%D0%B8%D0%BD%D0%B3), перевод звонка, автоматическое разъединение и т. д.)
* Сопряжение разных сетей (в частности, для адаптации разных диалектов протокола, зависимых от производителей)
* Сокрытие структуры сети (частные адреса, сетевая топология и т. п.)

Довольно часто B2BUA является частью [медиа-шлюза](http://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D0%B0%D1%88%D0%BB%D1%8E%D0%B7) для того, чтобы полностью контролировать медиа-потоки в рамках сессии. Сигнальный шлюз, являющийся частью[пограничного контроллера соединений/сеансов](http://ru.wikipedia.org/wiki/Session_Border_Controller) — наглядный пример применения B2BUA.

## Сообщения протокола SIP

[Сообщения протокола](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D0%B0) SIP (запросы и ответы), представляют собой последовательности текстовых строк, закодированных в соответствии с документом [RFC 2279](http://tools.ietf.org/html/rfc2279). Структура и синтаксис сообщений SIP идентичны используемым в протоколе [HTTP](http://ru.wikipedia.org/wiki/HTTP). Структура сообщений протокола SIP:

|  |
| --- |
| **Стартовая строка** |
| **Заголовки** |
| Пустая строка |
| **Тело сообщения** |

* Стартовая строка — начальная строка любого SIP-сообщения. Если сообщение является запросом, в ней указывается тип запроса, адресат и номер версии протокола. Если сообщение является ответом на запрос, в ней указывается номер версии протокола, тип ответа и его короткая расшифровка.
* Заголовки сообщений содержат информацию, необходимую для обработки сообщения (информация об отправителе, адресате, пути следования и пр.)
* Тело сообщения содержит описание сеансов связи. Не все запросы содержат тело сообщения (например запрос **BYE**). Все ответы могут содержать тело сообщения, но содержимое тела в них бывает разным.

## Алгоритмы установления соединения

Протокол SIP является управляющим протоколом для установления, модификации и разрыва соединения, ориентированного на передачу потоковых данных. Параметры передачи медиа-потоков описываются в протоколе SIP посредством [SDP](http://ru.wikipedia.org/wiki/SDP) (протокол описания сессии). Потоковые медиа-данные могут передаваться различными средствами, среди которых наиболее популярны транспортные протоколы [RTP](http://ru.wikipedia.org/wiki/RTP) и [RTCP](http://ru.wikipedia.org/wiki/RTCP).

Протокол SIP определяет 3 основных сценария установления соединения: с участием прокси-сервера, с участием сервера переадресации и непосредственно между пользователями. Сценарии отличаются по тому, как осуществляется поиск и приглашение вызываемого пользователя. Основные алгоритмы установления соединения описаны в [RFC 3665](http://tools.ietf.org/html/rfc3665).

## SIP-T и SIP-I

## Для взаимодействия с [традиционными телефонными сетями](http://ru.wikipedia.org/wiki/%D0%A2%D1%84%D0%9E%D0%9F), использующими сигнализацию [ОКС-7](http://ru.wikipedia.org/wiki/%D0%9E%D0%9A%D0%A1-7), были разработаны модификации протокола SIP для телефонии: Session Initiation Protocol for Telephones (SIP-T) и Session Initiation Protocol Internetworking (SIP-I). Разность версий ввиду того, что SIP-I был разработан [ITU-T](http://ru.wikipedia.org/wiki/ITU-T), а SIP-T — [IETF](http://ru.wikipedia.org/wiki/IETF) и описан в [RFC](http://ru.wikipedia.org/wiki/RFC) 3372. Основная задача данных модификаций протокола SIP заключается в прозрачной передаче сообщений [ISUP](http://ru.wikipedia.org/wiki/ISUP) по IP-сети. Данная задача осуществляется путём [инкапсуляции](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8))сигнальных единиц ОКС в сообщения SIP. Все требуемые задачи по взаимодействию между протоколами были решены на базе протокола SIP:

|  |  |
| --- | --- |
| **Требование по взаимодействию** | **Функция SIP-T** |
| Прозрачность сигнализации ISUP | Инкапсуляция ISUP в тело сообщения SIP |
| Возможность маршрутизировать сообщения SIP в зависимости от ISUP | Трансляция параметров ISUP в заголовке сообщения SIP |
| Трансляция адресной информации при установленном соединении | Использование метода INFO |

## Сравнение с H.323

SIP пригоден для чтения человеком и структурирован в отношении запросов и откликов. Сторонники SIP также заявляют о нём как о более простом, по сравнению с [H.323](http://ru.wikipedia.org/wiki/H.323)[[4]](http://ru.wikipedia.org/wiki/SIP#cite_note-4). Однако некоторые[[*кто?*](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%98%D0%B7%D0%B1%D0%B5%D0%B3%D0%B0%D0%B9%D1%82%D0%B5_%D0%BD%D0%B5%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D1%85_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9)] склонны считать, что, в то время как первоначально целью SIP была простота, в своём сегодняшнем виде он стал так же сложен, как и H.323. Другие[[*кто?*](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%98%D0%B7%D0%B1%D0%B5%D0%B3%D0%B0%D0%B9%D1%82%D0%B5_%D0%BD%D0%B5%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D1%85_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9)] считают, что SIP — протокол без состояний, который тем самым даёт возможность легко реализовать восстановление при отказе и другие возможности, которые затруднены в протоколах с состояниями, таких как H.323. SIP и H.323 не ограничены голосовой связью, они могут обслуживать любой сеанс связи, от голосового до видеосеанса или приложений будущего.

|  |  |  |
| --- | --- | --- |
| **Параметр сравнения** | **SIP** | [**H.323**](http://ru.wikipedia.org/wiki/H.323) |
| **Дополнительные услуги** | Набор услуг, поддерживаемых обоими протоколами примерно одинаков | |
| **Персональная мобильность пользователей** | Имеется хороший набор средств поддержки мобильности | Персональная мобильность поддерживается, но менее гибко |
| **Расширяемость протокола** | Удобная расширяемость, простая совместимость с предыдущими версиями | Расширяемость поддерживается, но существует ряд сложностей |
| **Масштабируемость сети** | Оба протокола обеспечивают хорошую масштабируемость сети | |
| **Время установления соединения** | Достаточно одной транзакции | Требуется несколько транзакций. |
| **Сложность протокола** | Простой, мало запросов, текстовый формат сообщений | Сложный, много запросов и протоколов, двоичное представление сообщений |
| **Совместимость оборудования** | Практически никакой. Каждый производитель SIP устройств соблюдает только тот набор рекомендаций (RFC) который ему нравится, ибо набор этих рекомендаций очень велик. Совместим фактически только базовый вызов | Практически полная. Стандарты устоявшиеся и имеют чёткий набор спецификаций |