

## On the Naming and Binding of Network Destinations

Именованье и привязки адресатов в сети

### Статус документа

Этот документ содержит информацию для сообщества Internet. Документ не задает какого-либо стандарта Internet и может распространяться без ограничений.

### Аннотация

В этом коротком документе рассматривается вопрос именования адресатов в коммуникационных сетях и предложены две идеи. Во-первых, полезно различать 4 разных типа объектов, которые можно назвать получателями пакетов в сети. Во-вторых, концепции привязки операционных систем являются полезным способом описания взаимоотношений между этими 4 типами объектов. Для иллюстрации полезности такого подхода рассматриваются некоторые более тонкие и противоречивые свойства двух реальных систем именования адресатов.

### Примечание

Этот документ изначально был опубликован в сборнике «Local Computer Networks» под редакцией P. Ravasio и др., изданном North-Holland Publishing Company, Amsterdam в 1982 г. (стр. 311-317). Copyright IFIP, 1982. От IFIP получено согласие на воспроизведение документа в некоммерческих целях. Право бесплатного копирования документа предоставляется при условии нераспространения копий в коммерческих целях, наличия указания на авторские права IFIP, названия и даты публикации и указания, что копирование выполняется с согласия IFIP. Для иных вариантов копирования или повторной публикации требуется специальное разрешение.

Исследование поддерживалось Defense Advanced Research Projects Agency (DARPA) Государственного департамента США и Управлением военно-морских исследований (Office of Naval Research) по контракту N00014-75-C-0661.

## В чем заключается проблема?

Несмотря на полезную работу John Shoch [1] по организации обсуждения имен, адресов и маршрутов к адресатам в компьютерных сетях, эти дискуссии по-прежнему остаются более путанными, чем следует ожидать. Путаница иной раз возникает из-за слишком тесной связи между разными типами сетевых объектов и наиболее распространенными формами их именования. Она также обусловлена попытками обсуждать вопросы без четко определенных концепций. В этом документе предлагается иной подход на основе точки зрения, которая оказалась полезной в соответствующей области операционных систем компьютеров.

В операционных системах может возникать похожая путаница с именами и адресами, поскольку имеются имена файлов, уникальные идентификаторы, адреса в виртуальной и реальной памяти, номера страниц и блоков, адреса каналов ввода-вывода, адреса дисковых дорожек и т. д., почти бесконечно. Но большая часть этих проблем давно решена за счет понимания того, что концепция привязки (binding) обеспечивает системный подход к именванию [2] (Шох указал возможность применения этой концепции операционных систем и она стала основой документа). В операционных системах очень рано пришло понимание наличия слишком большого числа разнотипных идентификаторов, препятствующего пониманию на основе лишь различий между именами и адресами. Вместо этого лучше рассматривать все идентификаторы как примеры одного явления и просто спрашивать, где находится контекст, к которому привязано это имя (идентификатор, адрес и т. п.), и к какому объекту, с каким именем этот идентификатор там привязан. Такой же подход применим и к коммуникационным сетям.

Для начала рассмотрим предложенную Шохом терминологию в самом широком смысле:

- имя указывает то, что вы хотите (получить, найти);
- адрес указывает, где это находится;
- маршрут указывает путь к искомому.

Нет необходимости менять эти определения, но видно, что они оставляют возможности разных интерпретаций. Предложение Шоха предполагает, что эти три абстрактных концепции совместно обеспечивают интеллектуальный охват обсуждаемой темы. В этом документе предполагается, что более механистический подход может привести к упрощению понимания этого набора концепций. Этот механистический подход начинается с перечисления того, что можно найти в коммуникационной сети.

## Типы сетевых адресатов и их привязки

В сети передачи данных при рассмотрении вопроса об описании получателя пакета имеется несколько типов сущностей, присутствующих более чем в одном экземпляре, поэтому с ними связывают имена, чтобы отличить один экземпляр от другого. Из числа таких типов четыре появляются наиболее часто.

1. **Службы и пользователи** - это функции, которые кто-то использует, и клиенты, которые пользуются функциями. Примерами служб (сервиса) являются службы точного времени, службы учета или службы пересылки пакетов. Примером клиента является конкретный настольный компьютер.

2. Узлы - это компьютеры, на которых работают службы или пользовательские программы. Некоторые узлы являются клиентами сети, другие помогают реализовать сеть, обеспечивая услуги пересылки (здесь не требуется различать эти 2 типа узлов).
3. Точки присоединения к сети - это сетевые порты, к которым подключаются узлы. При обсуждении сетей передачи данных часто используется термин адрес в качестве идентификатора точки присоединения.
4. Пути между точками присоединения к сети, проходящие через узлы пересылки и каналы связи.

Можно отметить, что первый сделан шаг - заимствованы перечисление и классификация обсуждаемых объектов из мира абстрактных типов данных. Вторым шагом будет два наблюдения об именовании сетевых объектов - первый о форме, второй о привязках.

Первое наблюдение заключается в том, что можно выбрать любую форму именования, которая представляется полезной, - двоичные идентификаторы, строки печатаемых символов или что-то иное и можно выбирать имена из плоского или иерархического пространства. Для данного типа объектов может применяться более одной формы имен. Например, узел может иметь символьное имя (строка) из плоского пространства имен и уникальный двоичный идентификатор. Есть две семантические ловушки, связанные с формой именования. Во-первых, слово «имя» (name) в мире сетей обычно связывается со строкой печатаемых символов, а «адрес» с интерпретируемыми машиной двоичными строками. В мире систем и языков для первого обычно применяется термин «печатное имя» (print name), а для второго - «имя машины» (machine name) или «адрес», хотя термин имя часто применяется для обоих (здесь принято расширенное толкование термина «имя»). Вторая семантическая ловушка состоит в связывании некой условной формы имени с определенным типом объекта, как свойства этого типа (атрибута). Например, службы могут именоваться символьными строками, узлы уникальными идентификаторами, а точки присоединения к сети - двоичными адресами. Когда один участник дискуссии предполагает привязку определенной формы к конкретному типу объекта, а другой может не делать этого, обсуждение может зайти в тупик и озадачить всех участников.

Второе наблюдение для 4 упомянутых выше типов сетевых объектов состоит в том, что большинство требований к именованию в сети можно просто и кратко описать в терминах привязок и изменения привязок.

1. Данный сервис может работать на одном или множестве узлов и может потребоваться перенос службы с одного узла на другой без потери идентичности службы.
2. Данный узел может быть соединен с одной или несколькими сетевыми точками присоединения и может потребоваться переключение без потери идентичности узла.
3. Данная пара точек присоединения к сети может быть соединена одним или множеством путей и со временем может потребоваться смена пути без потери идентичности точек присоединения.

Это изложение требований к сетевому именованию преднамеренно сделано кратким. Отличный и глубокий обзор этих требований можно найти в работе Sunshine [3].

Каждое из этих трех требований включает идею сохранения идентичности сервиса, узла или точки присоединения. Для сохранения такой идентичности нужно сделать так, чтобы используемое для идентификации имя не менялось при требуемом типе перемещения. Если связи между службами, узлами, точками присоединения и маршрутами поддерживаются как список привязок, эта цель может быть достигнута легко. Обеспечение гибкости решения в конкретной сети является инженерной задачей. Суждение о том, что конкретная привязка может быть задана при проектировании сети и ее не потребуется менять (например, конкретная служба всегда будет работать на определенном узле), не должно запутывать вопрос о том, какие имена и привязки имеются в данный момент. В принципе для отправки пакета данным сервису нужно определить 3 привязки:

1. узел, на котором работает нужная служба;
2. точка подключения, с которой этот узел соединен;
3. путь от точки присоединения узла к точке присоединения сервиса.

В свою очередь, сеть должна предоставлять три концептуально разных службы привязки:

1. распознавание имен служб (service name resolution) для идентификации узлов, обеспечивающих сервис;
2. расположение именованного узла для определения точки подключения к узлу, найденному в п. 1;
3. служба маршрутизации для идентификации пути от запрашивающего к точке подключения, найденной в п. 2.

На каждом уровне привязки возможны варианты выбора узла, точки присоединения и пути. Например, можно выбирать узел лишь после предварительного просмотра разных путей к возможным узлам. В этом случае сетевые таблицы могут давать лишь частичную привязку, которая означает, что запрос дает список ответов вместо одного. Окончательное решение о привязке может быть отложено до последнего момента и записано вне упомянутых выше служб привязки, предоставляемых в сети.

В привязках есть одна важная тонкость, которая часто сбивает разработчиков с пути. Предположим, что в таблице сети указано, что сервис Lockheed DIALOG работает на узле 5. На деле здесь имеется три разных привязки, но в таблицу внесена лишь одна из них и она может быть изменена простой корректировкой таблицы.

1. Имя сервиса Lockheed DIALOG корректно ассоциируется с определенной службой и набором хранящихся в ней файлов. Обычно такое же имя не будет назначаться другому сервису. Связывание имени с сервисом достаточно постоянно (стабильно) и это постоянство обычно не выражается в единой, легко изменяемой таблице.
2. Точно так же, имя 5 назначено определенному узлу на долгосрочной основе и не предполагается его изменение. Поэтому данное назначение обычно тоже не включается в единую, легко изменяемую таблицу.
3. Факт работы сервиса DIALOG на узле 5 является единственной привязкой, включенной в таблицу, поскольку предполагается возможность изменения этой ассоциации. Таблица предназначена для того, чтобы можно

было выражать изменения, например, перенос службы DIALOG на узел 6 и организацию сервиса Pipe-fitting на узле 5.

Ошибкой проектирования будет допущение о том, что эта таблица позволяет дать службе Lockheed DIALOG новое имя путем простого изменения таблицы. Это не входит в функции данной таблицы привязок и такое изменение имени фактически достаточно сложно реализовать, поскольку рассматриваемая ассоциация обычно не выражается привязкой в одной таблице. Чтобы выполнить такую смену имени, нужно изменить не только таблицу, но и пользовательские программы, документацию, и анонсы службы.

## Примеры из реального мира

Хотя изложенные идеи представляются достаточно простыми, в реальном мире легко найти примеры, когда интерпретация этих идей затруднена. Например, в сетях Xerox/DEC/Intel Ethernet<sup>1</sup> [5, 6] концепция точки присоединения к сети исчезает, поскольку она сводится к имени узла. Узел может фактически подключиться к сети Ethernet в любой ее точке и приносит с собой уникальный 48-битовый идентификатор<sup>2</sup>, который его интерфейсы отслеживают в проходящих пакетах. Этот идентификатор вероятно следует считать именем точки подключения к сети, даже если сама точка физического подключения может размещаться где угодно в сети. В то же время, можно принять правило, в соответствии с которым узел будет предоставлять из своей памяти 48-битовый идентификатор, используемый интерфейсом Ethernet, и другим, не менее разумным, допущением (при интерпретации идентификатора в сети) будет считать этот 48-битовый идентификатор именем самого узла. С точки зрения привязки этот способ использования Ethernet привязывает имена узла и точки подключения к одному уникальному 48-битовому идентификатору.

Эта постоянная привязка имени узла к имени точки подключения обеспечивает некоторые преимущества при управлении сетью:

- узел может перемещаться из одного физического места в другое без изменения сетевых записей;
- один уровень привязки таблиц опущен и это особенно заметно при реализации межсетевой маршрутизации;
- узел, подключенный к двум сетям Ethernet, может представлять одну точку присоединения для обеих сетей, что упрощает взаимодействие между маршрутизаторами и поиск дополнительных путей.

Но постоянная привязка вызывает странности, когда одному узлу нужно подключиться к двум точкам присоединения в одной сети Ethernet. Странность обусловлена тем, что для независимой адресации второй точки подключения остается единственный способ - позволить узлу использовать два разных 48-битовых идентификатора, а это означает, что некоторые другие записи сети (интерпретирующие этот идентификатор как имя узла), скорее всего будут обманчивыми, полагая, что в сети два узла, а не один. Для предотвращения путаницы можно использовать один 48-битовый идентификатор для обеих точек подключения, но тогда не будет возможности направить сообщение конкретной точке. В любом случае постоянная привязка имени узла к имени точки подключения осложняет выполнение некоторых функций, хотя суммарное преимущество в этом случае может перевесить потерю функциональности.

В качестве другого примера рассмотрим протокол ARPANET NCP, предоставляющий символьные имена, которые из-за их мнемоники выглядят как имена узлов или служб, но фактически являются именами точек подключения [6]. Таким образом, RADC-Multics является именем точки подключения в ARPANET IMP 18, порт 0 и переподключение узла (компьютер Honeywell 68/80) к другой точке требует от пользователей узнать новое имя для сервиса или поменять таблицы на всех других узлах. Изменение таблиц на первый взгляд кажется сменой всех привязок, но необходимость изменить более одной таблицы является признаком более глубоких изменений. На самом деле меняется постоянное имя точки подключения к сети. Можно увидеть это более четко, отметив, что параллельное подключение этого Honeywell 68/80 ко второму порту ARPANET будет возможно лишь путем назначения второго символьного идентификатора. Это требование подчеркивает, что имя реально относится к точке подключения, а не к узлу. К сожалению, из-за мнемоники имена ARPANET NCP часто воспринимаются как имена служб. Поэтому можно ожидать, что сервис Rome Air Development Center Multics будет работать на узле, доступном по имени RADC-Multics. Это конкретное допущение не вызывает сюрпризов. Однако любой из 4 компьютеров Digital PDP-10 в Bolt, Beranek и Newman может принимать почту для всех остальных, как и группы PDP-10 в USC Information Sciences Institute и Massachusetts Institute of Technology. Если узел, на который они пытаются отправить почту, не работает (down), клиент должен понять, что эта услуга доступна, запрашивая другой узел с тем, что представляется другим именем службы. Необходимость клиента понимать, что он должен указать другое имя для получения тех же услуг, обусловлена тем, что в ARPANET имя не относится к сервису, который привязан к узлу, привязанному к точке подключения, а является просто именем точки подключения.

Кроме того, может возникнуть путаница из-за того, что три концептуально разных службы привязки (распознавание имен служб, размещение поименованного узла и распределение маршрутов) могут не различаться механистически. Обычно предлагается лишь одна идентифицируемая услуга - сервер имен. Этот сервер принимает имя службы и возвращает список точек подключения, которые обеспечивают этот сервис. Таким образом, этот сервер выполняет первую и вторую услугу концептуальной привязки, хотя может оставлять клиенту окончательный выбор используемой точки подключения. Выбор пути может быть реализован распределенным алгоритмом маршрутизации, который незаметно обеспечивает финальную службу привязки.

## Соответствие имен, адресов и маршрутов

С учетом этой модели привязок служб, узлов, точек присоединения и путей, одна из возможных интерпретаций имен, адресов и маршрутов Шоха может иметь описанное ниже представление.

1. Любой из 4 типов объектов (сервис, узел, точка присоединения к сети, путь) может иметь имя, хотя Шох ограничил имена понятными человеку строками символов.
2. Адрес объекта - это имя (в широком смысле без принятого Шохом ограничения), связанное с объектом. Таким образом, адресом сервиса будет имя узла, на котором этот сервис работает. Адрес узла - это имя точки подключения к сети, с которой узел соединен. Адрес точки присоединения (концепция, которую обычно не

<sup>1</sup>Прообраз современных сетей Ethernet IEEE 802. Прим. перев.

<sup>2</sup>Прообраз MAC-адреса. Прим. перев.

рассматривают) можно принять как имя пути, ведущего к этой точке. Эта интерпретация следует Шоку (адрес указывает, где это находится), но не совсем соответствует другому представлению Шоха о том, что адрес является формой идентификации, обрабатываемой машиной, а не человеком. Это по-видимому является основным расхождением во взглядах по части четкости и ясности определений.

3. Маршрут является более сложным понятием. Маршрут к точке присоединения или узлу - это просто путь в принятом здесь толковании термина. Поскольку на одном узле может работать множество служб, маршрут к службе состоит из пути к точке присоединения узла, где работает сервис, и той или иной идентификации действия, запускающего службу на узле (например, идентификатор сокета в протоколах PUP internet [4] или ARPA Internet [7]). Следует отметить, что в реальности маршрут может состоять из последовательности имен, обычно это список имен узлов пересылки или точек присоединения и имен, применяемых узлами пересылки для путей между ними.

Независимо от отношения к данной интерпретации терминов Шоха, кажется очевидным, что используется более 3 концепций, поэтому для их обсуждения нужно больше 3 языков.

## Заключение

В этом документе утверждается, что некоторое понимание в части именования адресатов (получателей) в сети можно получить путем выделения (принятия) 4 типов именуемых объектов в каждой точке назначения или на пути к ней - служб, узлов, точек присоединения и маршрутов - с определением 3 последовательных изменяемых привязок (службы к узлу, узла к точке присоединения, точки присоединения к маршруту). Это представление, промоделированное на основе аналогичных успешных привязок в системах управления хранилищами (файл - область хранения - физическое местоположение) и виртуальной памятью (объект - сегмент - страница - блок памяти), дает систематическое объяснение некоторых проблем проектирования, которые возникают в сетевых системах именования.

## Благодарности

Обсуждения с David D. Clark, J. Noel Chiappa, David P. Reed и Danny Cohen помогли прояснить используемые здесь рассуждения. John F. Shoch вдохновил эту работу и представил подробные замечания, но он не отвечает за результат.

## Литература

1. Shoch, John F., "Inter-Network Naming, Addressing, and Routing," IEEE Proc. COMPCON Fall 1978, pp. 72-79. Also in Thurber, K. (ed.), Tutorial: Distributed Processor Communication Architecture, IEEE Publ. #EHO 152-9, 1979, pp. 280-287.
2. Saltzer, J. H., "[Naming and Binding of Objects](#)", in: Operating Systems, Lecture notes in Computer Science, Vol. 60, Edited by R. Bayer, New York; Springer-Verlag, 1978.
3. Sunshine, Carl A., "[Addressing Problems in Multi-Network Systems](#)", to appear in Proc. IEEE INFOCOM 82, Las Vegas, Nevada, March 30 - April 1, 1982.
4. Boggs, D. R., Shoch, J. F., Taft, E. A., and Metcalfe, R. M., "[PUP: An Internetwork Architecture](#)", IEEE Trans. on Comm. 28, 4 (April, 1980) pp. 612-623.
5. (Anonymous), "[The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications, Version 1.0](#)", published by Xerox Corp., Palo Alto, Calif., Intel Corp., Sunnyvale, Calif., and Digital Equipment Corp., Tewksbury, Mass., September 30, 1980.
6. Dalal, Y. K., and Printis, R. S., "[48-bit Absolute Internet and Ethernet Host Numbers](#)", Proc. Seventh Data Communications Symposium, Mexico City, Mexico, October 1981, pp. 240-245.
7. Feinler, E., and J. Postel, Editors, "ARPANET Protocol Handbook", SRI International, Menlo Park, Calif., January, 1978.

## Вопросы безопасности

В этом документе вопросы безопасности не рассматриваются.

## Адрес автора

Jerome H. Saltzer

M.I.T. Laboratory for Computer Science

545 Technology Square

Cambridge, MA 02139

U.S.A.

Phone: (617) 253-6016

E-Mail: [Saltzer@MIT.EDU](mailto:Saltzer@MIT.EDU)

## Перевод на русский язык

Николай Малых

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)