

# Прикладной уровень

(Компьютерные сети т.2 стр.182-216)

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.

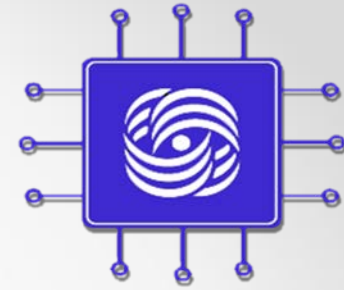
Кафедра АСВК

ф-т ВМК МГУ



# Прикладной уровень

- NAT - Network Address Translation
- DNS - Domain Name Service
- HTTP - Hyper Text Transfer Protocol
- SNMP - Simple Network Management Protocol
- SMTP - Simple Mail Transfer Protocol
- FTP - File Transfer Protocol

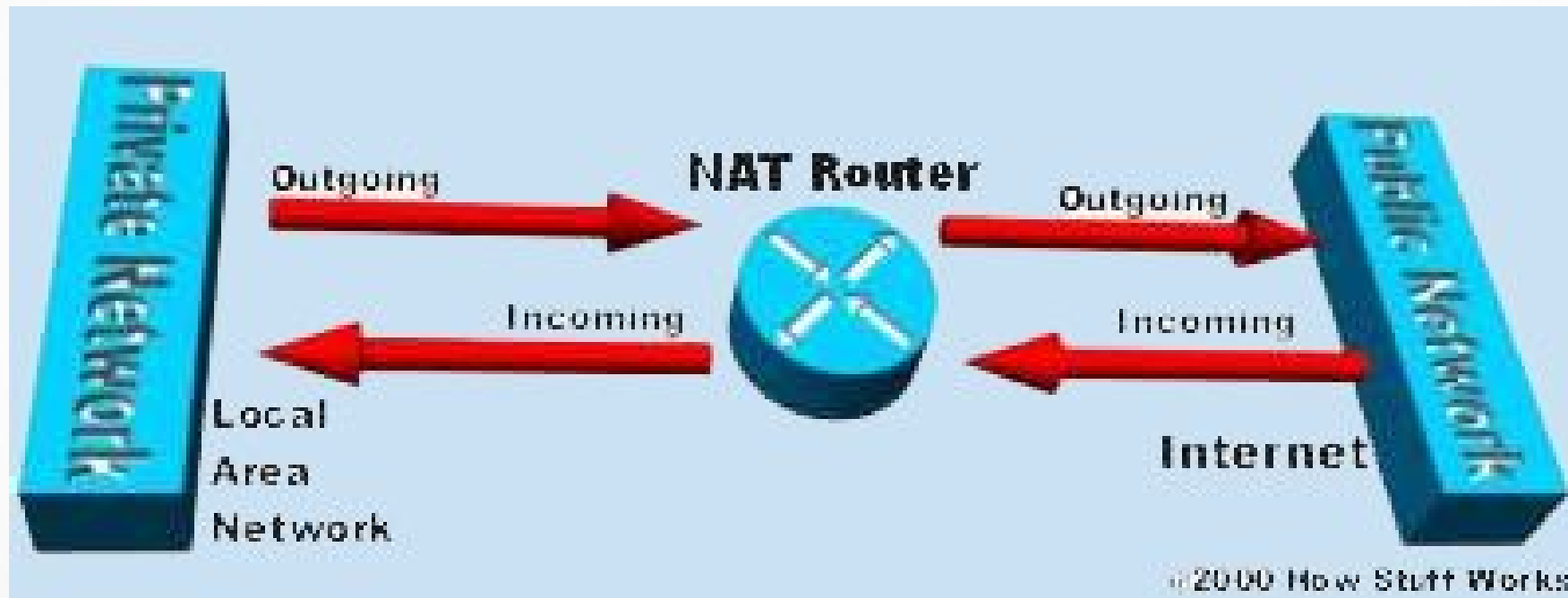


# NAT – Network Address Translation

(NAT описан в RFC 1631, RFC 3022)



# NAT OPERATION





# Виды отображения, реализуемые NAT

- **Статический NAT**
- **Cone NAT, Full Cone NAT**
- **Address-Restricted cone NAT (Restricted cone NAT)**
- **Port-Restricted cone NAT**
- **Симметричный NAT (Symmetric NAT)**
- Терминология и типизация NAT по RFC3489



# Статический NAT

10.0.0.101



TCP port 5000



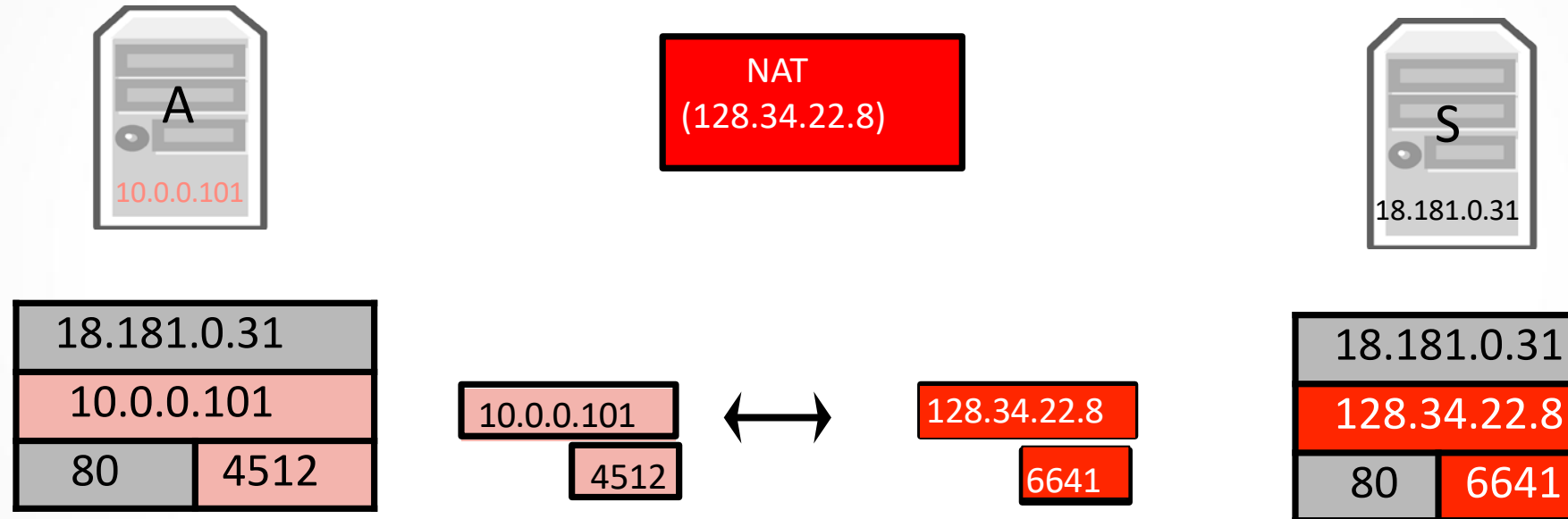
18.181.0.31



TCP port 22

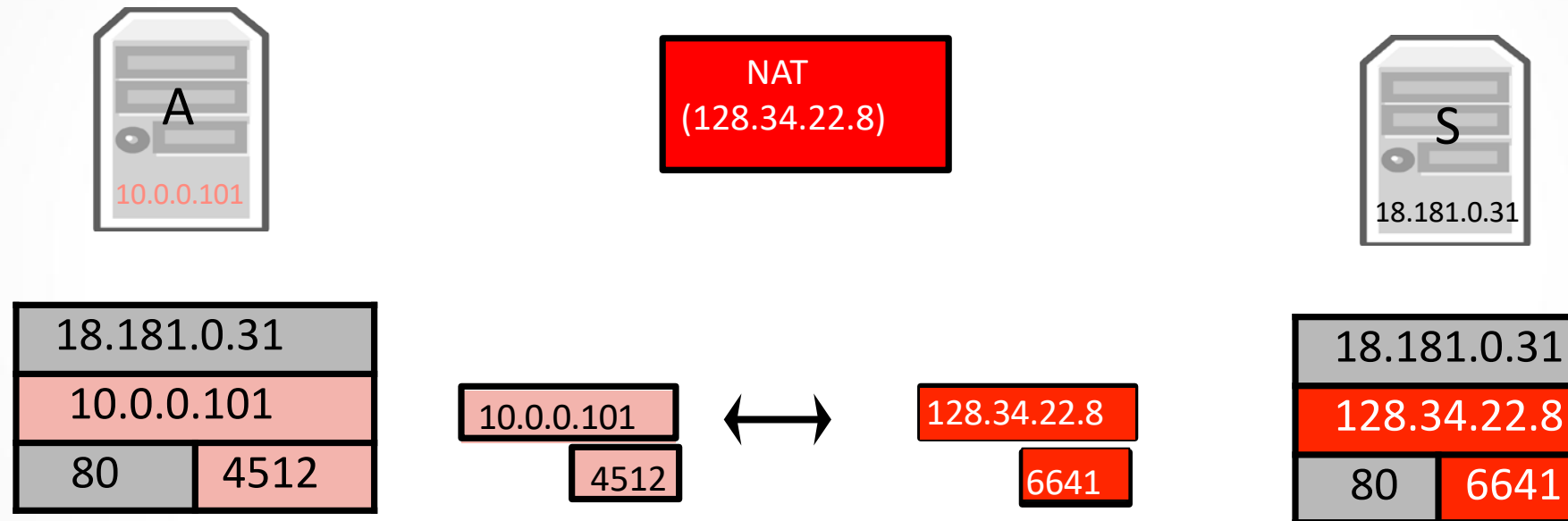


# Full Cone (FC) NAT





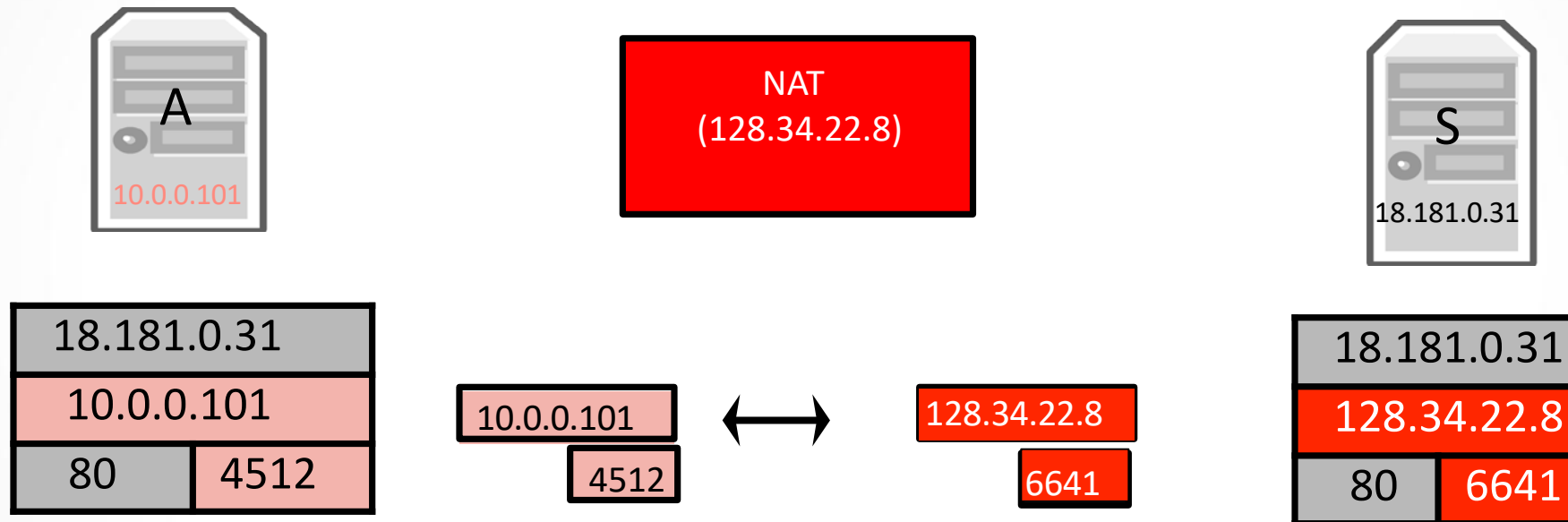
# Restricted Cone (RC) NAT





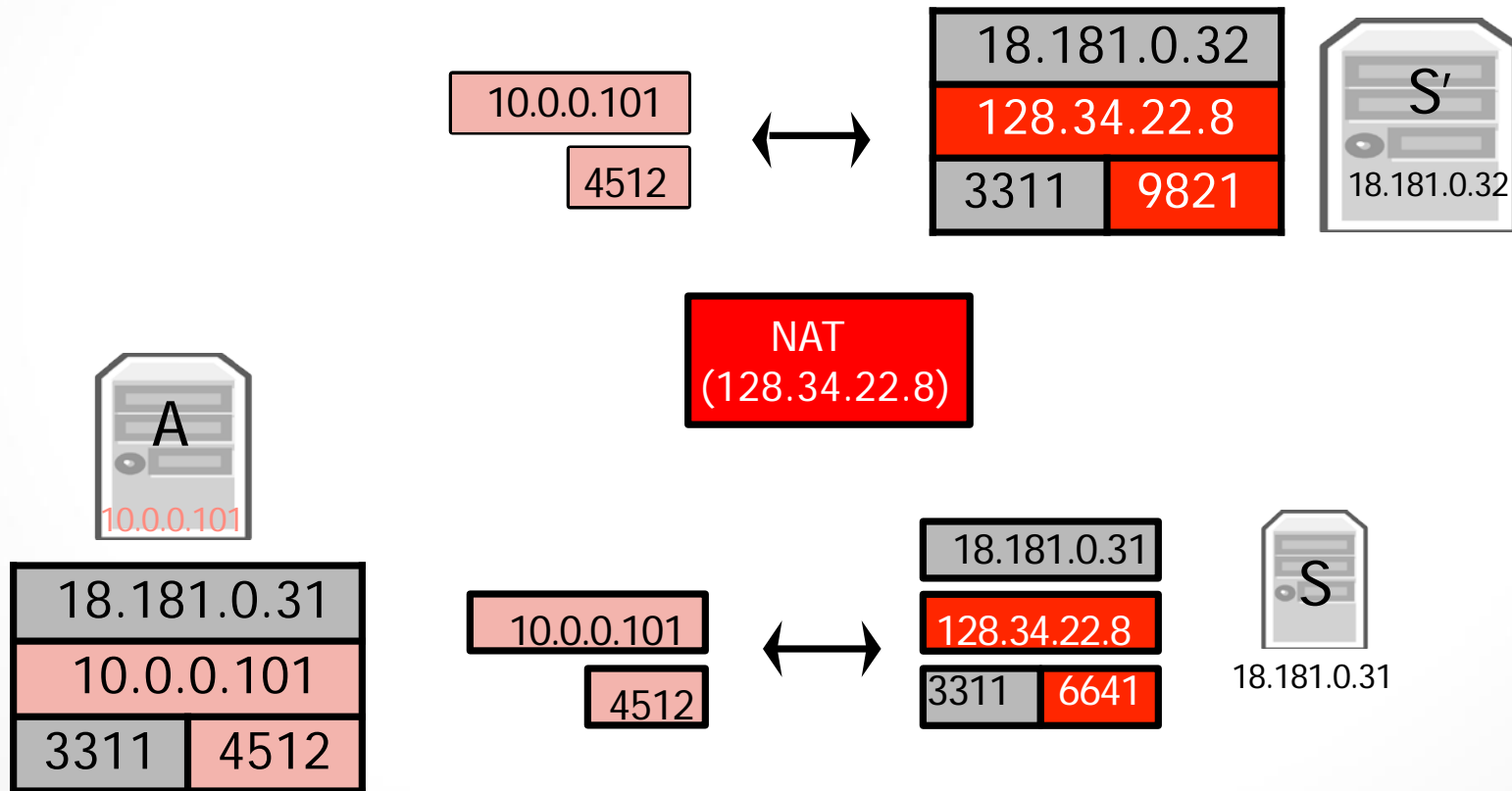


# Port Restricted (PR) NAT



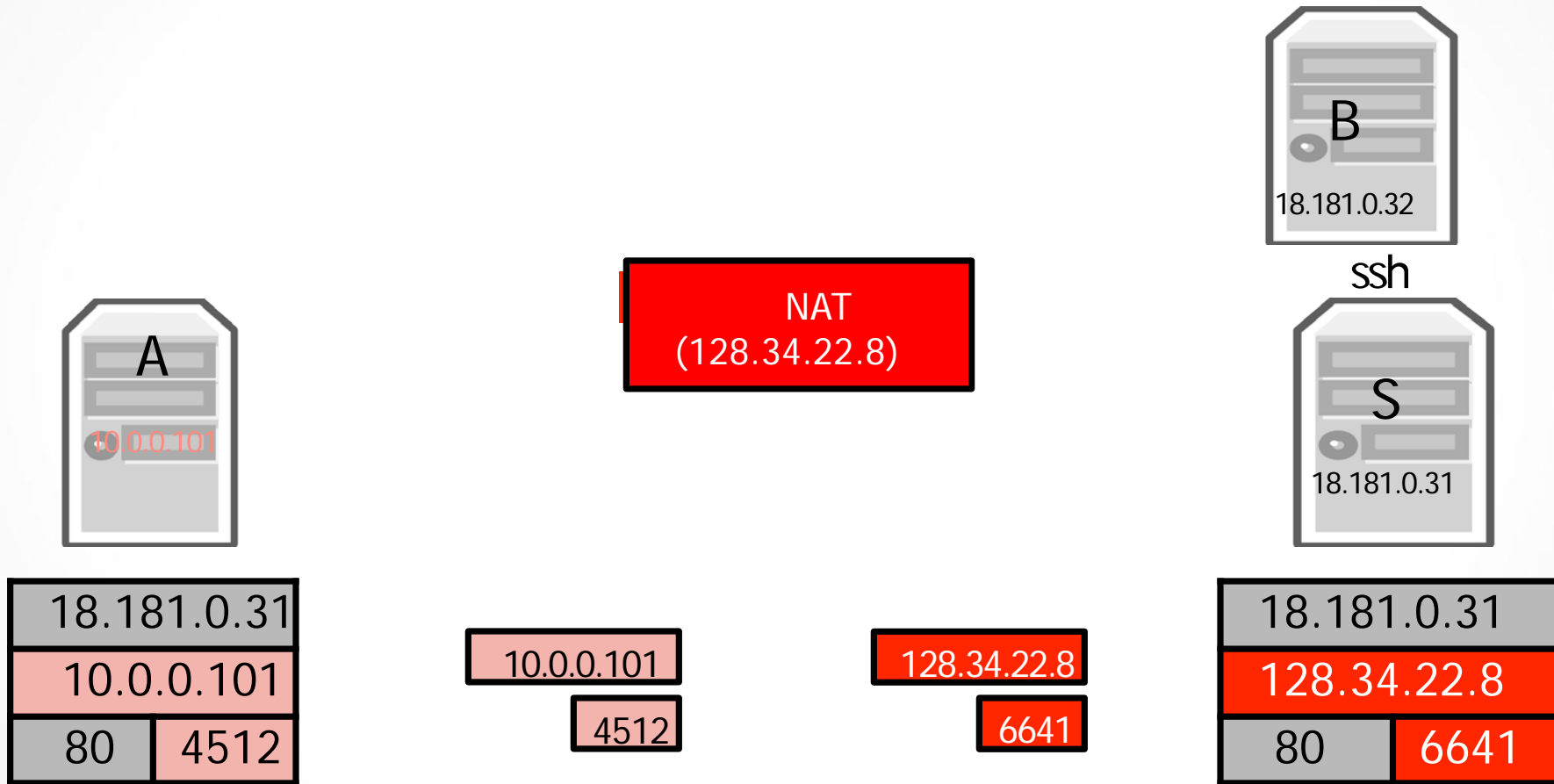


# Symmetric NAT





# Влияние NAT на приложения: ВХОДЯЩИЕ СОЕДИНЕНИЯ





# Приложение: NAT перфоратор (hole-punching)

Server  
(18.181.0.31)

NAT  
(128.34.22.8)

NAT  
(76.18.117.20)

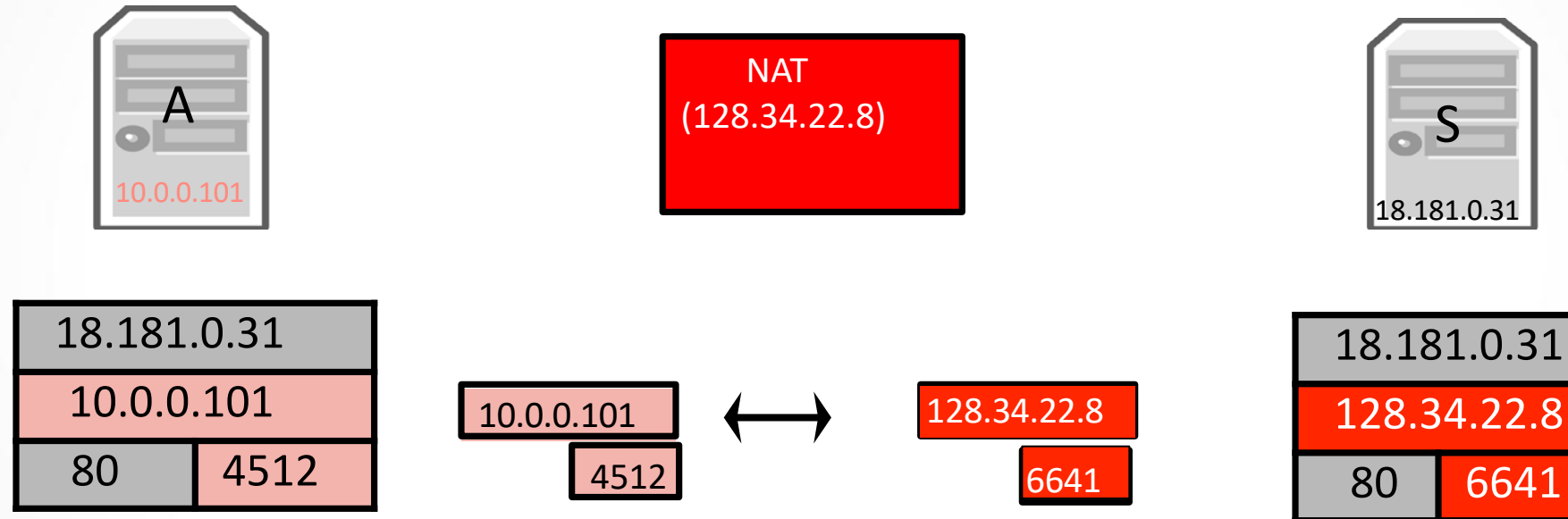


Client A  
(10.0.0.101)

Client B  
(10.1.1.9)



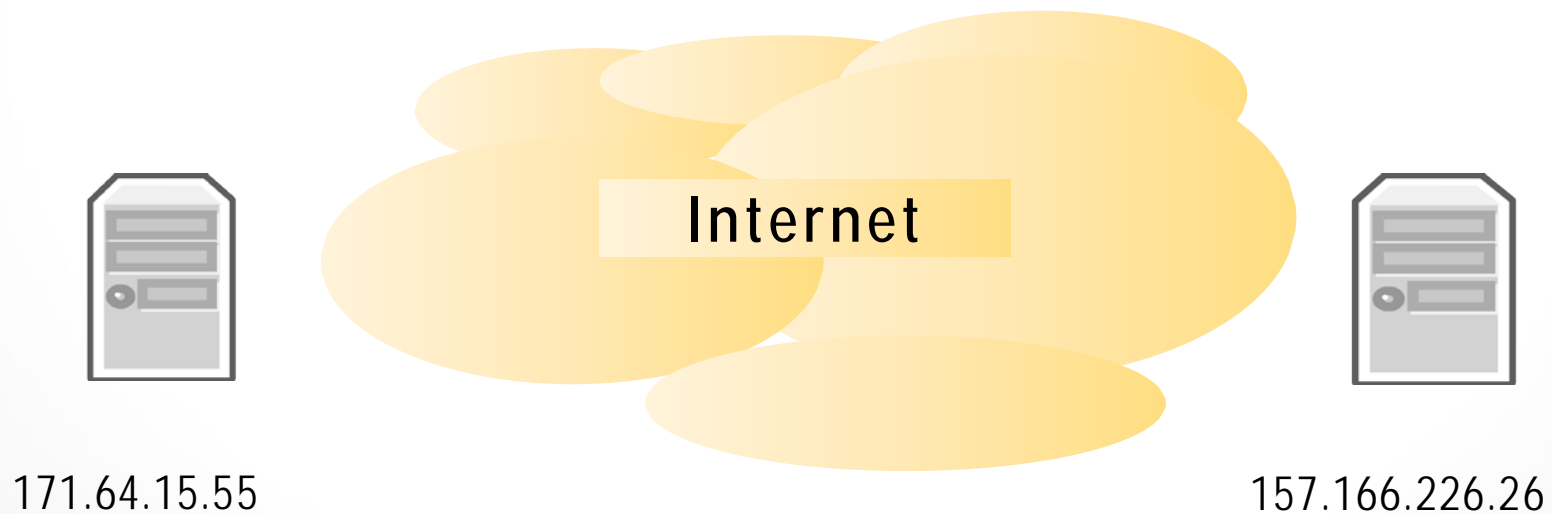
# NAT = Нет новому Транспорту!





# Strong End-to-End

"The network's job is to transmit datagrams as efficiently and flexibly as possible. Everything else should be done at the fringes."





# Новая талия Интернета





# Обсуждение NAT

- *Очень полезное новшество*
  - *повторное использование адресов*
  - *безопасность (не явное открытие соединения – это может быть полезно)*
- *Очень осложняет жизнь*
  - *очень усложняет разработку приложений*
  - *без NAT VoIP было бы проще*
- *Дебатировать NAT можно, но бесполезно – он есть!*





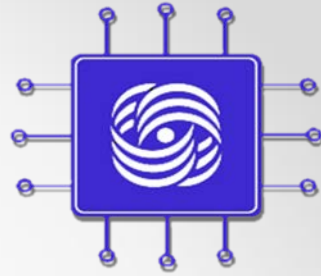
# NAT выполняет три важных функции

- Позволяет сэкономить IP-адреса, транслируя несколько внутренних IP-адресов в один внешний публичный IP-адрес (или в несколько, но меньшим количеством, чем внутренних).
- Позволяет предотвратить или ограничить обращение снаружи к внутренним хостам, оставляя возможность обращения из внутренней сети во внешнюю. Если для пакетов, поступающих из внешней сети, соответствующей трансляции не существует (а она может быть созданной при инициации соединения или статической), они не пропускаются.
- Позволяет скрыть определённые внутренние сервисы внутренних хостов/серверов. Тем самым, снаружи, на внешнем IP-адресе после трансляции адресов на сайт (или форум) для осведомлённых посетителей можно будет попасть по адресу <http://dlink.ru:54055>, но на внутреннем сервере, находящимся за NAT, он будет работать на обычном 80-м порту.



## Однако следует упомянуть и о недостатках данной технологии:

- *Не все протоколы могут "преодолеть" NAT.*
- *Из-за трансляции адресов "много в один" появляются дополнительные сложности с идентификацией пользователей и необходимость хранить полные логи трансляций.*
- *Атака DoS со стороны узла, осуществляющего NAT – если NAT используется для подключения многих пользователей к одному и тому же сервису, это может вызвать иллюзию DoS-атаки на сервис (множество успешных и неуспешных попыток).*



# DNS – Domain Name Service



# DNS

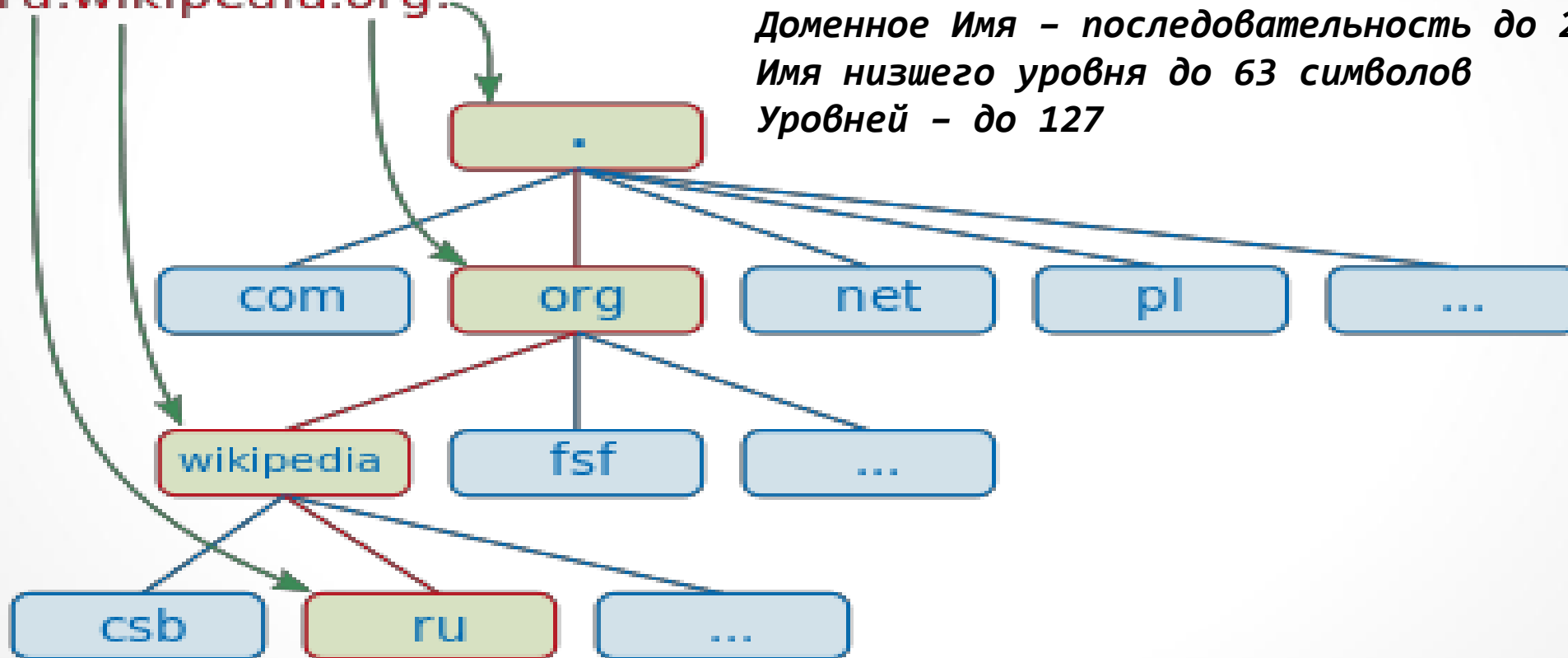
- *Задача:*
  - сопоставление символического имени IP адресу
  - автоматическая резолуция адресов
  - DNS (Domain Name System)
  - распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Internet
  - механизм взаимодействия машин в сети Internet с этой базой данных



# DNS: структура имен

- Иерархическая древовидная структура имен

ru.wikipedia.org.



*Доменное Имя - последовательность до 254 символов  
Имя низшего уровня до 63 символов  
Уровней - до 127*

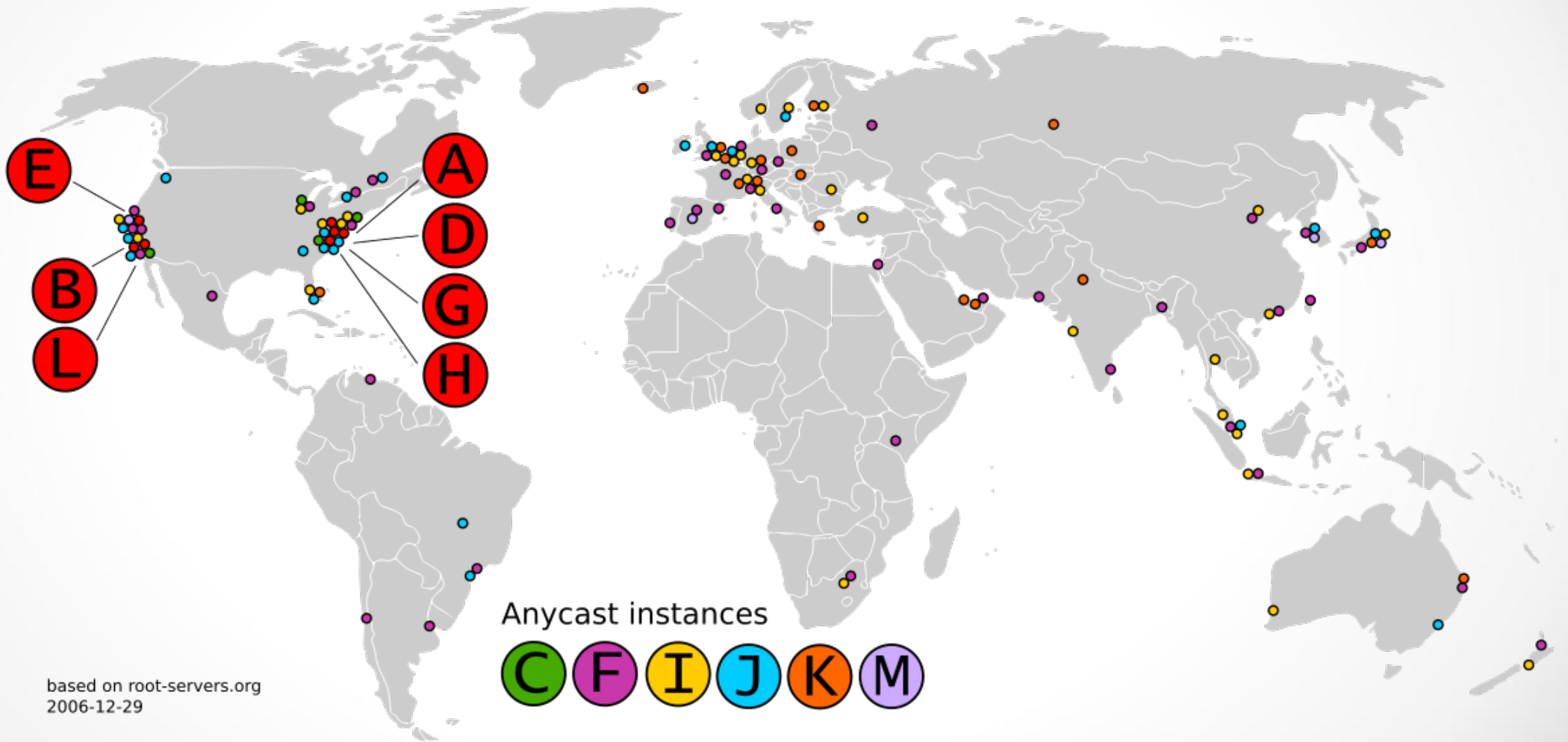


## Стандартизированные суффиксы имен

Поле адреса	Тип сети
.aero	Фирма или организация, относящаяся к сфере авиации;
.arts	Культура и досуг;
.biz	Организация, относящаяся к сфере бизнеса;
.com	Коммерческая организация;
.coop	Кооперативная организация;
.firm	Коммерческое предприятие;
.gov	Государственное учреждение (США);
.info	Открытая TLD-структура (регистрация имен доменов)
.org	Бесприбыльная организация;
.edu	Учебное заведение;
.jobs	Работодатели;
.mil	Военное предприятие или организация (США);
.mobi	Сайты и сервисы, ориентированные на работу с мобильными телефонами и беспроводными устройствами
.museum	Имя домена музея
.name	Имя домена частного лица
.net	Большая сеть;
.pro	Профессионал, достойный доверия. Управляется RegistryPro ( <a href="http://www.nic.pro/">http://www.nic.pro/</a> );
.int	Международная организация;
.rec	Развлечения;
.tel	Хранение и управление персональными и корпоративными контактными данными;
.travel	Турагентства;
.tv	Телевидение. Хотя существует домен bbc.tv, а регистрация в этой зоне в РФ процветает (см. <a href="#">Ru center</a> , официальный статус в качестве TLD этот домен не получил. В базе данных IANA (см. <a href="#">Национальные коды доменов в Интернет</a> ) этот домен записан попрежнему за TUVALU
.arpa	Специальный домен, используемый для преобразования IP-



# Географическое расположение корневых серверов DNS





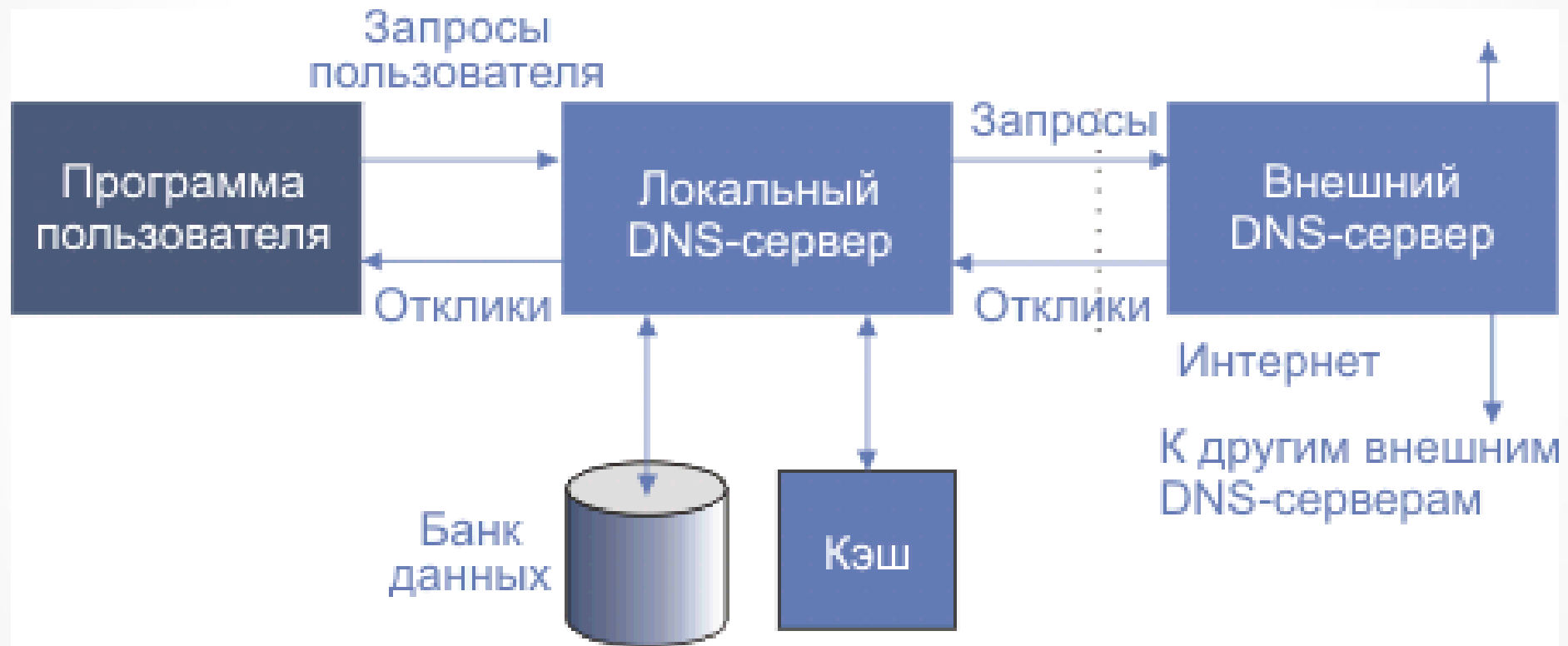
# DNS: ОСНОВНЫЕ ПОНЯТИЯ

- *Домен – узел и связанное поддерево в иерархической структуре имен*
- *Ресурсная запись – единица хранения информации*
- *DNS-сервер и DNS-клиент*
- *DNS-запрос*
- *Зона – часть поддерева домена вместе с ресурсными записями*
- *Делегирование – передача ответственности за зону отдельному серверу*





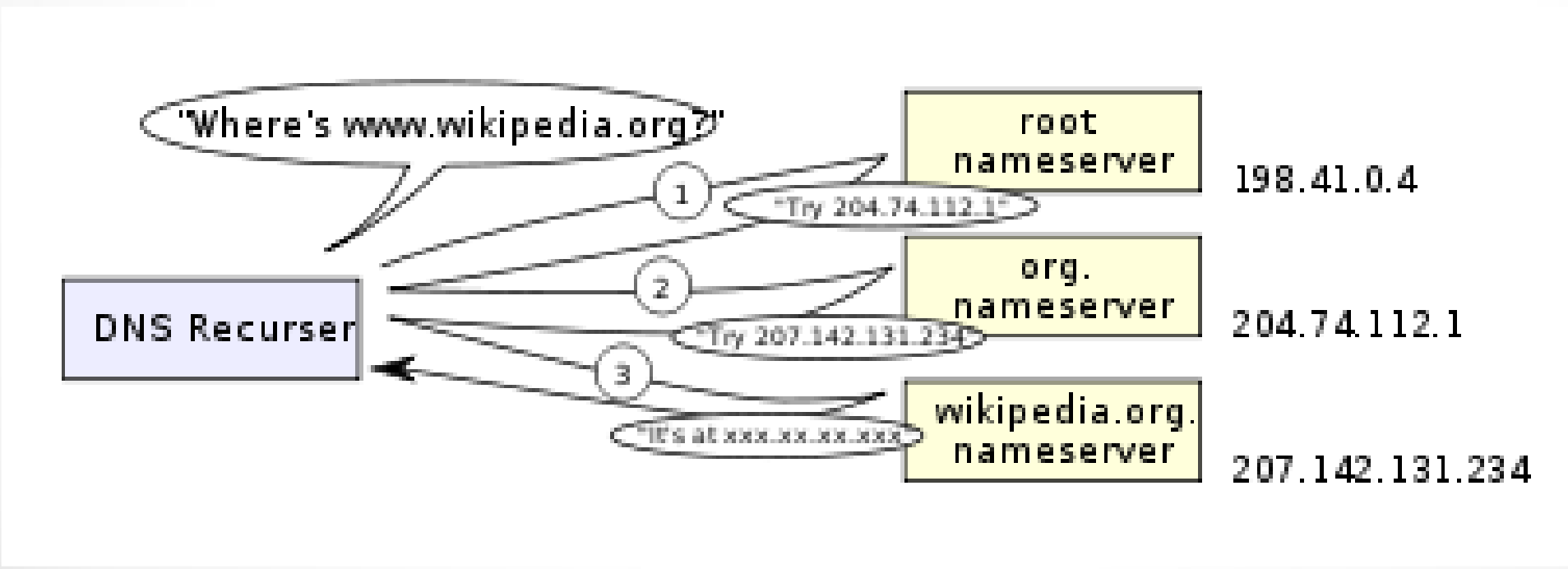
# Структура организации работы DNS приложения





# DNS: пример рекурсивной резолуции адреса

## Рекурсивное или итеративное поведение





# DNS: ресурсная запись

- *имя (NAME) – доменное имя, к которому привязана данная ресурсная запись*
- *TTL (Time To Live) – допустимое время хранения данной ресурсной записи в кэше неответственного **DNS-сервера***
- *тип (TYPE) ресурсной записи – определяет формат и назначение данной ресурсной записи*
- *поле данных (RDATA), формат и содержание которого зависит от типа записи.*



# DNS: типы ресурсных записей

- *Запись A (address record) или запись адреса связывает имя хоста с адресом IP*
- *Запись AAAA (IPv6 address record) связывает имя хоста с адресом протокола IPv6*
- *Запись CNAME (canonical name record) или каноническая запись имени (псевдоним) используется для перенаправления на другое имя*
- *Запись MX (mail exchange) или почтовый обменник указывает сервер(ы) обмена почтой для данного домена*
- *Запись NS (name server) указывает на DNS-сервер для данного домена*
- *Запись PTR (pointer) или запись указателя связывает IP хоста с его каноническим именем*
- *Запись SOA (Start of Authority) или начальная запись зоны указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону*
- *Запись TXT содержит не интерпретируемую текстовую информацию*



# Пример записи в БД DNS

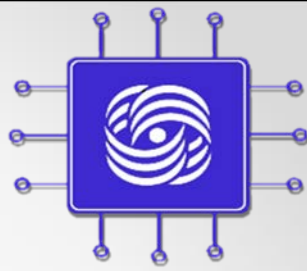
Authoritative data for cs.msu.ru

Name	TTL	Class	Type	
cs.msu.ru.	86400	IN	SOA	ns.cs.msu.ru. root.cs.msu.ru. (2001082400, 10800, 1800, 3600000, 259200)
cs.msu.ru.	86400	IN	TXT	“CS Dept of the Moscow State University”
cs.msu.ru.	86400	IN	NS	ns.cs.msu.ru.
cs.msu.ru.	86400	IN	NS	ns1.cs.msu.ru.
cs.msu.ru.	86400	IN	NS	ns1.barrnet.net.
cs.msu.ru.	86400	IN	NS	ipsun.ac.msk.su.
cs.msu.ru.	86400	IN	NS	ns.radio-msu.net.
cs.msu.ru.	86400	IN	MX	10 mailhost.cs.msu.ru.
mailhost.cs.msu.ru.	86400	IN	HINFO	Sun Enterprise 450, Solaris 10



## DNS: заключительные замечания

- Доменная система именованя указывает на то, кто ответственен за поддержку имени, но не где эта машина находится (несмотря на коды стран)
- Понятия доменного имени и адрес сети вообще говоря не связаны: две машины одного домена имен могут не принадлежать к одной сети
- У машины может быть много имен. В частности, это верно для машин, предоставляющих какие-либо услуги, которые в будущем могут быть помещены под опеку другой машины



# HTTP

## Hyper Text Transfer Protocol



# Hyper Text

## History [ [edit source](#) | [edit beta](#) ]

The term **HyperText** was coined by **Ted Nelson** who in turn was inspired by **Vannevar Bush**'s microfilm-based "memex". **Tim Berners-Lee** first proposed the "WorldWideWeb" project — now known as the **World Wide Web**. Berners-Lee and his team are credited with inventing the original HTTP along with HTML and the associated technology for a web server and a text-based web browser. The first version of the protocol had only one **method**, namely GET, which would request a page from a server.<sup>[3]</sup> The response from the server was always an HTML page.<sup>[4]</sup>

The first documented version of HTTP was **HTTP V0.9** <sup>[5]</sup> (1991). **Dave Raggett** led the **HTTP Working Group** (HTTP WG) in 1995 and wanted to expand the protocol with extended operations, extended negotiation, richer meta-information, tied with a security protocol which became more efficient by adding additional methods and **header fields**.<sup>[5][6]</sup> **RFC 1945** <sup>[6]</sup> officially introduced and recognized HTTP V1.0 in 1996.

The HTTP WG planned to publish new standards in December 1995<sup>[7]</sup> and the support for pre-standard HTTP/1.1 based on the then developing **RFC 2068** <sup>[7]</sup> (called HTTP-NG) was rapidly adopted by the major browser developers in early 1996. By March 1996, pre-standard HTTP/1.1 was supported in **Arena**,<sup>[8]</sup> **Netscape 2.0**,<sup>[8]</sup> Netscape Navigator Gold 2.01,<sup>[8]</sup> **Mosaic 2.7**,<sup>[citation needed]</sup> **Lynx 2.5**<sup>[citation needed]</sup>, and in **Internet Explorer 2.0**<sup>[citation needed]</sup>. End-user adoption of the new browsers was rapid. In March 1996, one web hosting company reported that over 40% of browsers in use on the Internet were HTTP 1.1 compliant.<sup>[citation needed]</sup> That same web hosting company reported that by June 1996, 65% of all browsers accessing their servers were HTTP/1.1 compliant.<sup>[9]</sup> The HTTP/1.1 standard as defined in **RFC 2068** <sup>[8]</sup> was officially released in January 1997. Improvements and updates to the HTTP/1.1 standard were released under **RFC 2616** <sup>[9]</sup> in June 1999.

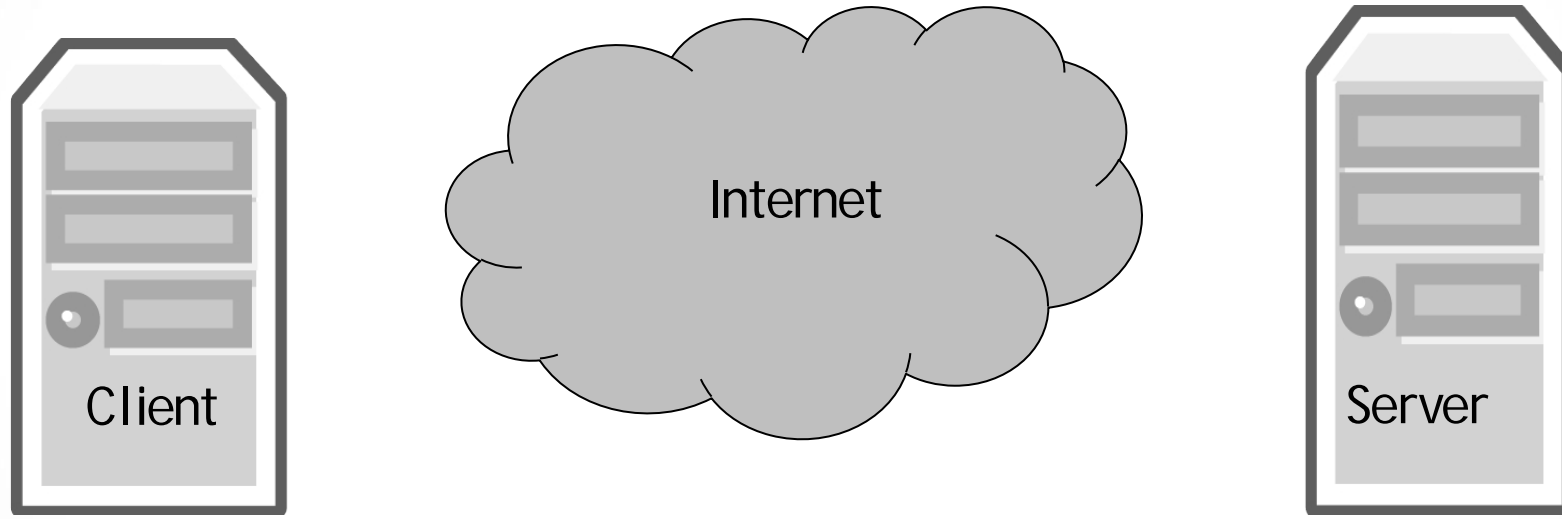


```
218 <h2><span class="mw-headline" id="History">History</span></h2></div><div class="thumb tright">
219 <div class="thumbinner" style="width:192px;"><a href="/wiki/File:Tim Berners-Lee CP 2.jpg" class="image">
221 <div class="magnify"><a href="/wiki/File:Tim Berners-Lee CP 2.jpg" class="internal" title="Enlarge">Tim Berners-Lee</a></div>
223 </div>
224 </div>
225 </div>
226 <p>The term <a href="/wiki/HyperText" title="HyperText" class="mw-redirect">HyperText</a> was coined by <a href="/wiki/Ted Nelson" title="Ted Nelson">
227 <p>The first documented version of HTTP was <b><a rel="nofollow" class="external text" href="http://www.w3.org/pub/WWW/Protocols/HTTP/AsImplemented.ht
228 <p>The HTTP WG planned to publish new standards in December 1995<sup id="cite_ref-7" class="reference"><a href="#cite_note-7"><span></span></span></sup></p></div>
```



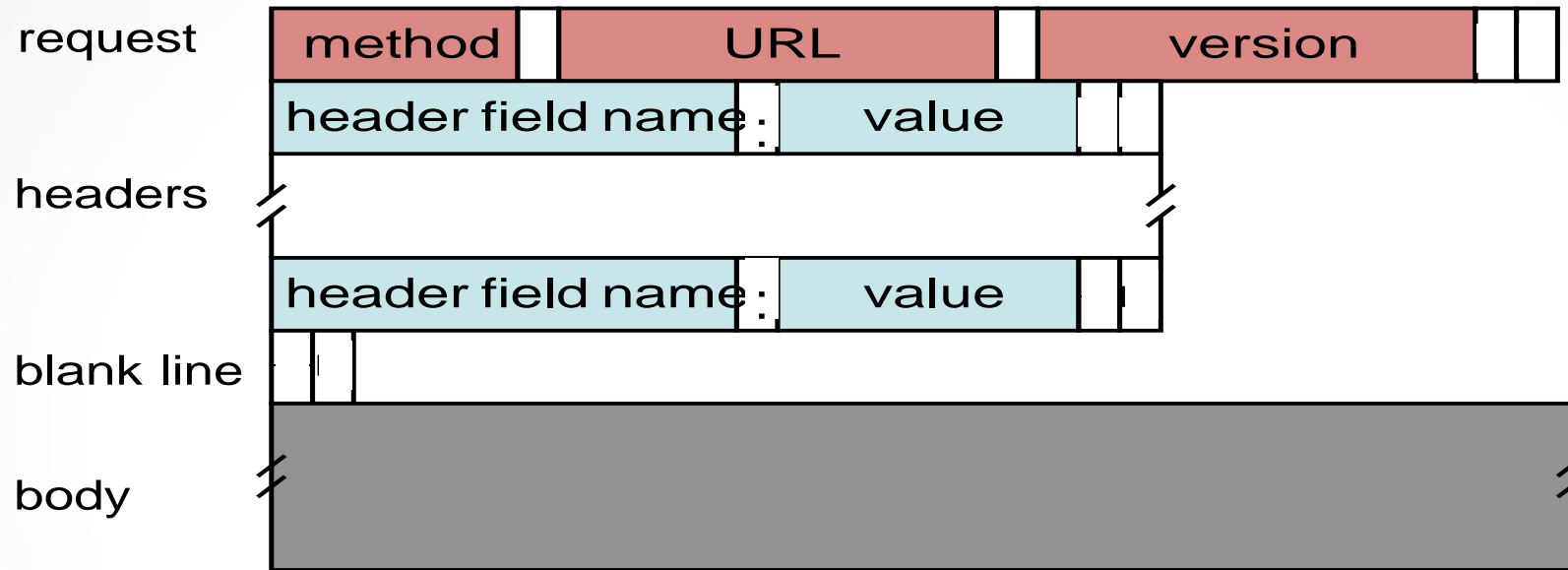


# World Wide Web (HTTP)





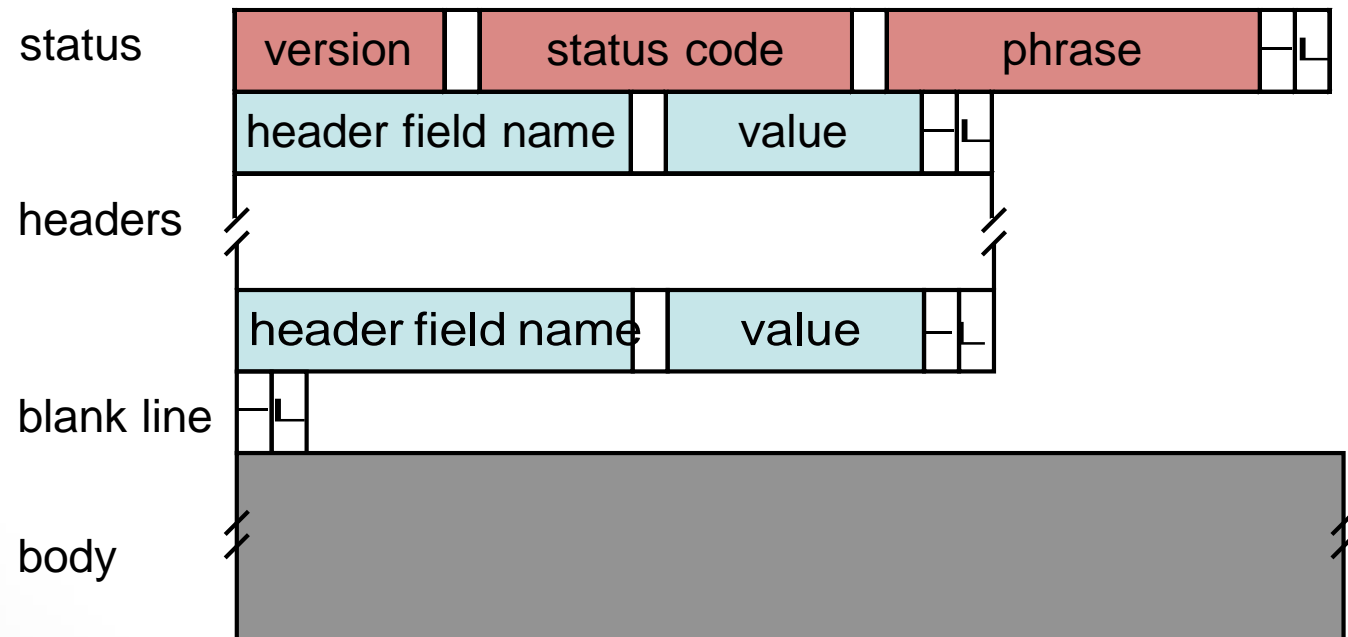
# HTTP Request Format



```
Server: Apache/2.2.11 (Win32) PHP/5.3.0;  
Last-Modified: Sat, 16 Jan 2010 21:16:42 GMT;  
Content-Type: text/plain;  
charset : windows-1251;  
Content-Language: ru;  
X-Powered-By или X-Cache
```



# HTTP Response





## Пример диалога HTTP

Запрос клиента:

GET /wiki/страница HTTP/1.1

Host: ru.wikipedia.org

User-Agent: Mozilla/5.0 (X11; U; Linux  
i686; ru; rv:1.9b5) Gecko/2008050509

Firefox/3.0b5

Accept: text/html

Connection: close

(пустая строка)

Ответ сервера:

HTTP/1.1 200 OK

Date: Wed, 11 Feb 2009 11:20:59 GMT

Server: Apache X-Powered-By: PHP/5.2.4-  
2ubuntu5wm1 Last-Modified: Wed, 11 Feb  
2009 11:20:59 GMT

Content-Language: ru

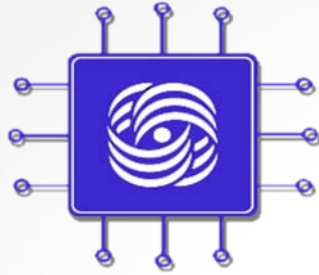
Content-Type: text/html; charset=utf-8

Content-Length: 1234

Connection: close

(пустая строка)

(далее следует запрошенная страница в  
HTML)



# SMTP – Simple Mail Transfer Protocol



## E-mail

- *Задача:*
  - *Унифицированный способ обмена “почтовыми” сообщениями через сеть Интернет*
- *E-mail*
  - *Создание, редактирование, чтение сообщений*
  - *Отправка сообщений получателю*
  - *Получение сообщений*

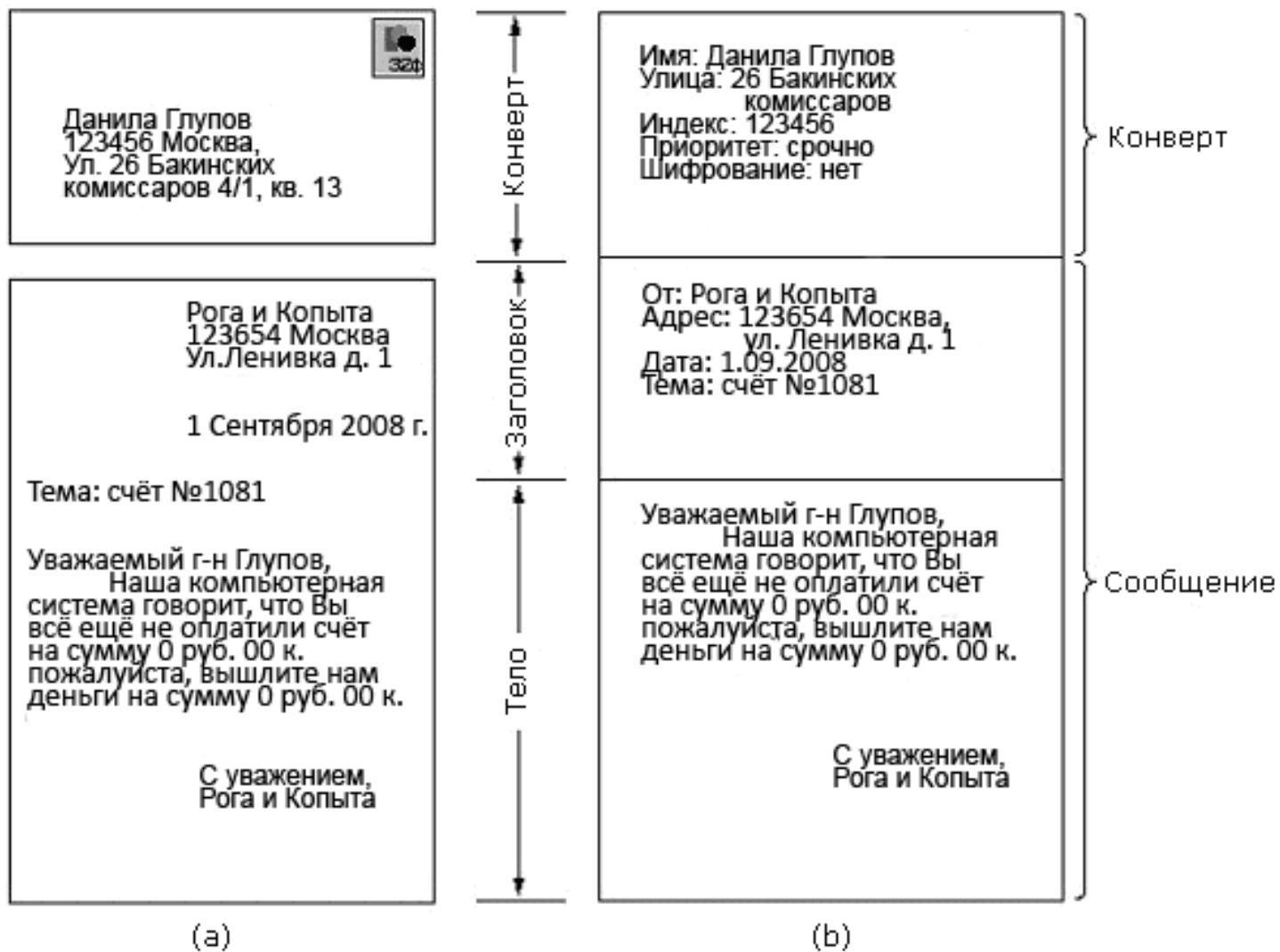


# Архитектура и сервис систем E-mail

- *Объединение агента пользователя (чтения/формирования сообщения) и агента передачи сообщений*
- *Основные функции почтовой службы*
  - *Композиция* – обеспечивает создание сообщений и ответов
  - *Передача* – обеспечивает передачу сообщения от отправителя к получателю без вмешательства пользователей
  - *Отчет перед отправителем о доставке*
  - *Отображение* сообщения, включая вопросы форматирования и кодировки
  - *Размещение* – вопросы хранения сообщений, поиска среди них, повторной отправки или переадресации и т.п.



# Конверты и сообщения







# E-mail: элементы заголовка сообщения

- **Return-Path** – обратный адрес
- **Received** – строчка журналирования прохождения письма. Каждый почтовый сервер (MTA) помечает процесс обработки этим сообщением
- **MIME-Version** – версия MIME, с которым это сообщение создано
- **From:** – Имя и адрес отправителя.
- **Sender:** – Отправитель письма. Добавлено для возможности указать, что письмо от чье-то имени (*from*) отправлено другой персоной (например, секретаршей от имени начальника)
- **To:** – Имя и адрес получателя. Может содержаться несколько раз (если письмо адресовано нескольким получателям). Может не совпадать с полем SMTP RCPT TO
- **CC:** – (от *carbon copy*). Содержит имена и адреса вторичных получателей письма, к которым направляется копия
- **bcc:** – (от *blind carbon copy*). Содержит имена и адреса получателей письма, чьи адреса не следует показывать другим получателям. Это поле обычно обрабатывается почтовым сервером (и приводит к появлению нескольких разных сообщений, у которых *bcc* содержит только того получателя, кому фактически адресовано письмо).



## E-mail: элементы заголовка сообщения

- **Reply-To:** – имя и адрес, куда следует адресовать ответы на это письмо. Если, например, письмо рассылается ботом, то в качестве Reply-To будет указан адрес персоны, готовой принять ответ на письмо
- **Message-ID:** – уникальный идентификатор сообщения. Состоит из адреса узла-отправителя и номера (уникального в пределах узла). Выглядит примерно так: `AAB77AA2175ADD4BACECE2A49988705C0C93BB7B4A@example.com`. Вместе с другими идентификаторами используется для поиска прохождения конкретного сообщения по журналам почтовой системы и для указания на письмо из других писем
- **In-Reply-To:** – указывает на Message-ID, для которого это письмо является ответом (с помощью этого почтовые клиенты могут легко выстраивать цепочку переписки)
- **Subject:** – тема письма
- **Date:** – дата написания письма
- **Content-Type:** – тип содержимого письма. С помощью этого поля указывается тип (HTML, RTF, Plain text) содержимого письма и кодировка, в которой создано письмо



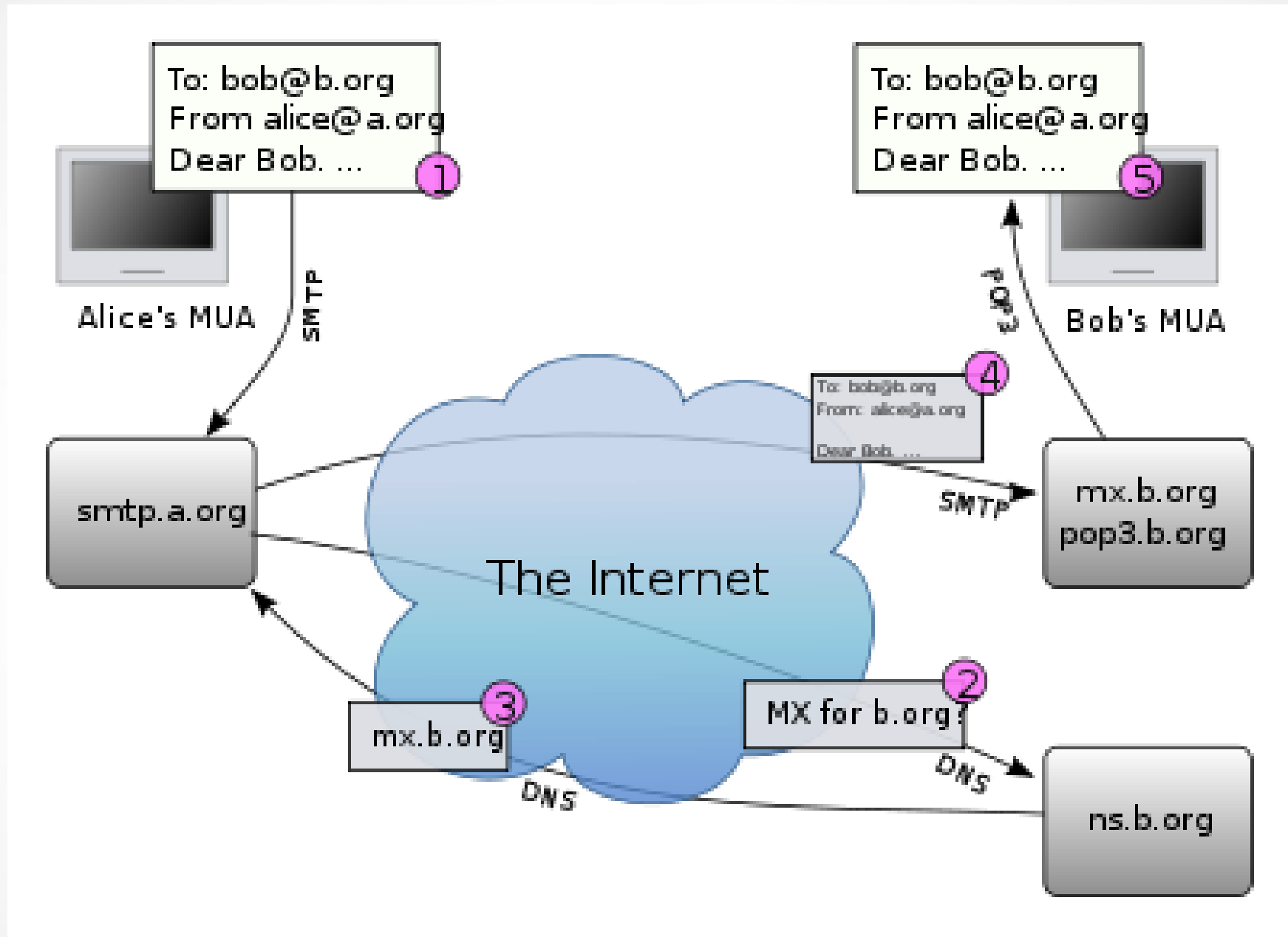
# MIME – Multipurpose Internet Mail Extension

- *Поддержка*
  - *Различных алфавитов, включая нелатинские*
  - *Передачи нетекстовых данных*



# E-mail: передача почтовых сообщений

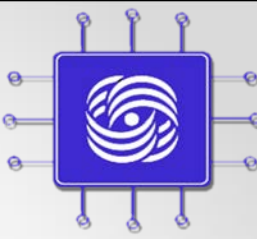
- *SMTP (Simple Mail Transfer Protocol) – передача письма между клиентом и сервером, между почтовыми серверами*
- *POP3 (Post Office Protocol) – простой протокол для изъятия почты из удаленного почтового ящика . Он позволяет забирать почту с сервера и хранить ее на машине пользователя*
- *IMAP (Interactive Mail Access Protocol) – позволяет одному и тому же пользователю заходить с разных машин на сервер, чтобы прочесть, отправить почту*
- *Почтовый ящик vs. Почтовый терминал*
- *Сервера передачи сообщений vs. Сервера хранения сообщений*





# E-mail: конфиденциальность почты

- PGP и PEM - распространенные безопасные почтовые системы



# SNMP - Simple Network Management Protocol



# SNMP

- **Задача:**
  - удаленное унифицированное управление сетевыми устройствами
- **SNMP (Simple Network Management Protocol)**
  - Сбор информации о параметрах конфигурации сетевых устройств
  - Изменение некоторых параметров конфигурации



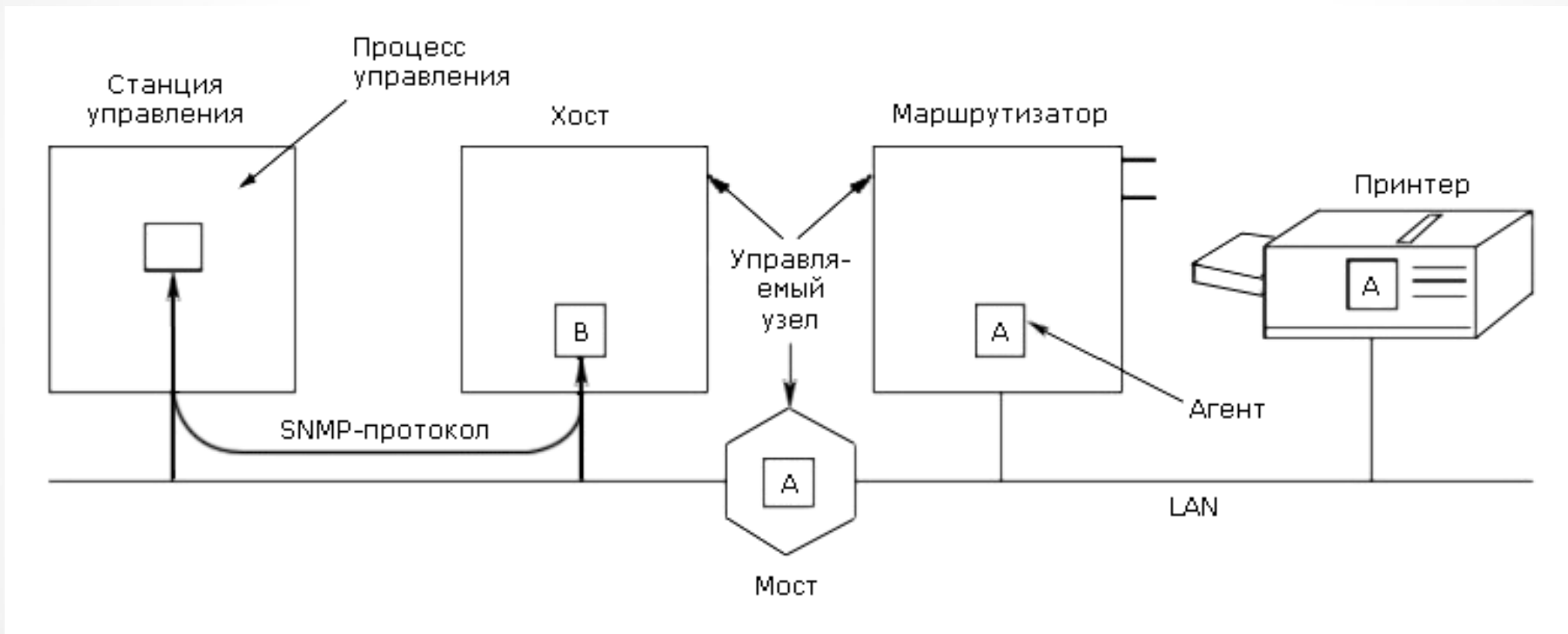


# SNMP: модель управления

- Управляемое устройство
- Агент – программное обеспечение, запускаемое на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства
- Система сетевого управления ( Network Management System , NMS) – приложение, отслеживающее и контролирующее управляемые устройства



# Модель управления в SNMP



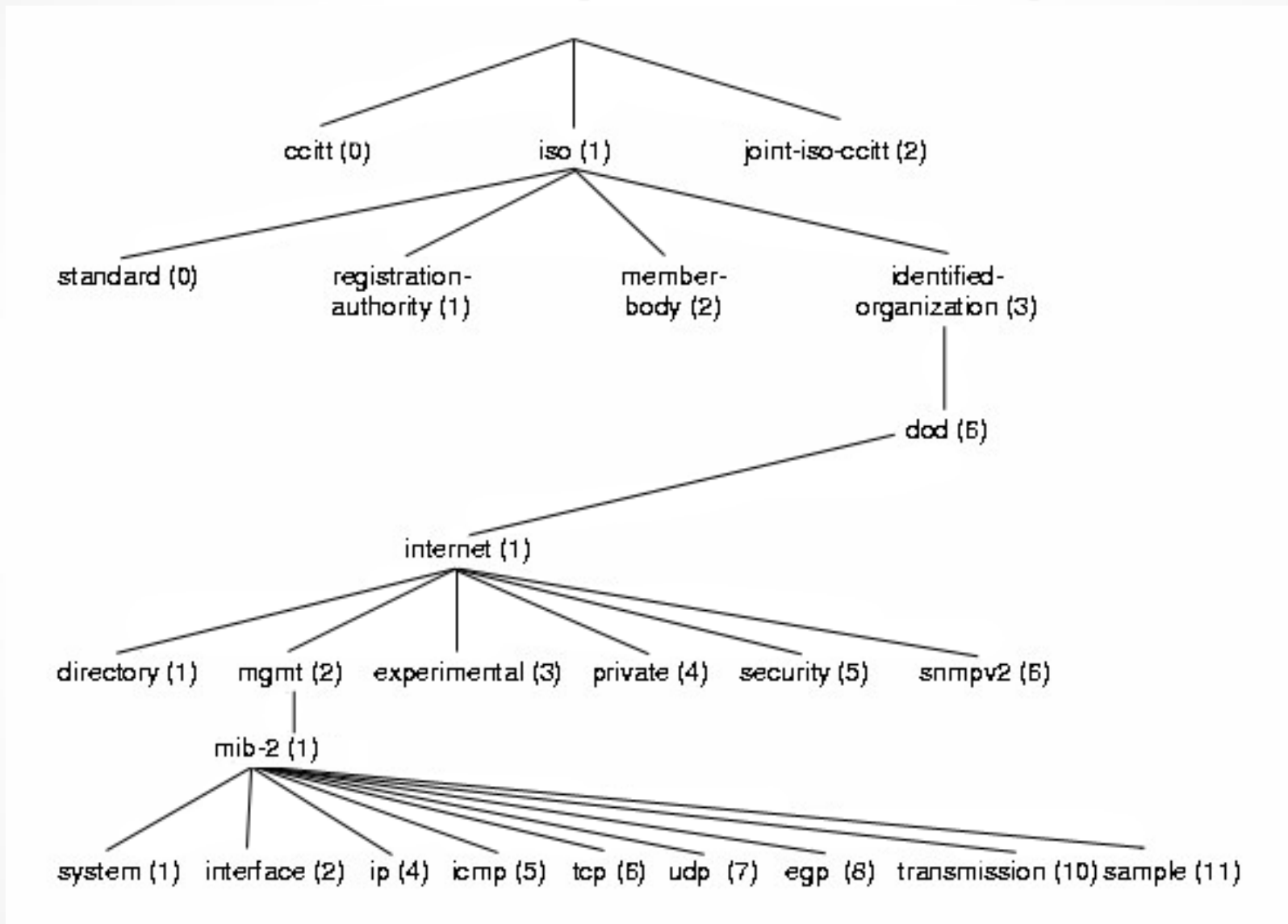


# SNMP: Управляющая информация

- *SNMP не специфицирует, какая именно информация должна предоставляться управляемым устройством*
- *Используется расширяемая иерархическая система представления информации в ASN.1 нотации*
- *Агент обеспечивает:*
  - *Удаленное взаимодействие управляемого устройства с NMS сервером*
  - *Трансляцию и представление управляющей информации*

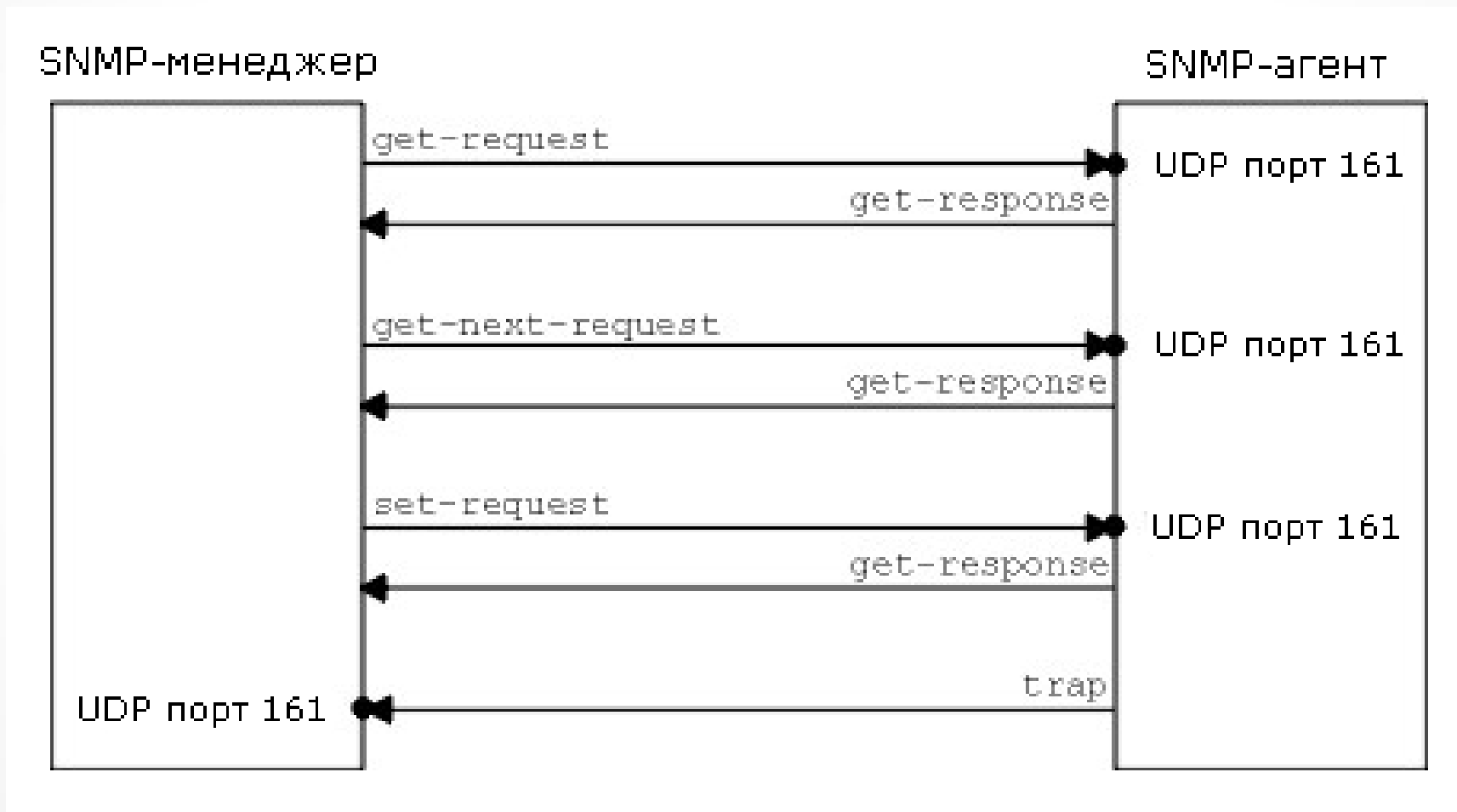


# Подмножество дерева стандартов в ASN.1





# SNMP: протокол взаимодействия и его КОМАНДЫ





# FTP – File Transfer Protocol



# FTP

- **Задача:**
  - доступ к файлам на удаленных машинах
  - надежная передача файлов
  - независимость клиента от файловой системы удаленной машины
- **FTP (File Transfer Protocol)**
  - протокол передачи файлов по сети



# FTP: Протокол передачи файлов

Алгоритм работы протокола FTP:

- Сервер FTP использует в качестве управляющего TCP соединение на порт 21, который всегда находится в состоянии ожидания соединения
- Устанавливают управляющее соединение по порту 21 между модулем "User-PI" и модулем сервера - "Server-PI"
- Клиент начинает отправлять на сервер команды согласования параметров канала передачи данных.
  - FTP-команды определяют параметры соединения передачи данных: роль участников соединения (активный или пассивный), порт соединения (как для "User-DTP", так и для "Server-DTP"), тип передачи, тип передаваемых данных, структуру данных и управляющие директивы, обозначающие действия, которые пользователь хочет совершить, например, сохранить, считать, добавить или удалить данные или файл
- Пассивный участник соединения (например, клиентский модуль "User-DTP") устанавливает режим ожидания открытия соединения на заданный для передачи данных порт.
- Активный модуль (например, "Server-DTP") открывает соединение и начинает передачу данных
- Окончание передачи данных:
  - соединение между "Server-DTP" и "User-DTP" закрывается, но управляющее соединение "Server-PI"-"User-PI" остается открытым.
  - Пользователь, не закрывая сессии FTP, может еще раз открыть канал передачи данных, передать необходимую информацию и т.д.



