Лекция 5. Клиент-серверная архитектура ПО. Протоколы передачи данных между клиентом и сервером

NetCracker®

Операционные системы и основы проектирования информационных систем

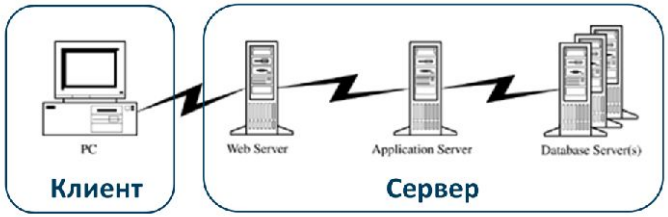
План лекции

- Клиент-серверная архитектура.
- Базы данных и СУБД.
- Модель обмена информацией OSI.
- Протоколы передачи данных.
- Основные протоколы транспортного уровня: TCP, UDP.
- Основные протоколы прикладного уровня: POP3/IMAP, SMTP, HTTP, HTTPS, FTP, SSH, SFTP, RDP, SCP.

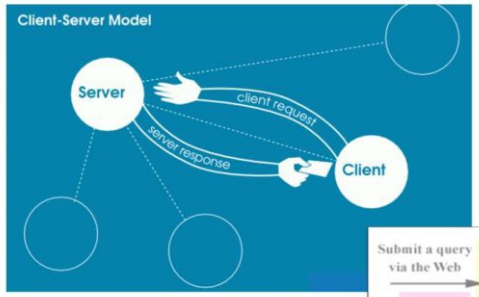
Клиент-серверная архитектура

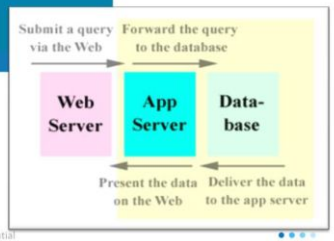
- **Клиент-сервер (Client-server)** — сетевая архитектура, в которой устройства являются либо клиентами, либо серверами.
- **Клиентом (front end)** является запрашивающая машина (обычно ПК), **сервером (back end)** — машина, которая отвечает на запрос.

Оба термина (**клиент и сервер**) могут применяться **как к физическим устройствам, так и к программному обеспечению.**



Клиент-серверная архитектура

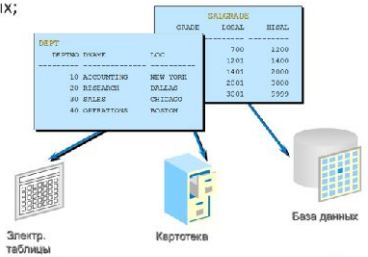




Базы данных и СУБД

<u>База данных (Database)</u> – упорядоченный набор данных, который обладает следующими свойствами:

- хранится в вычислительной системе;
- находится под управлением такой системы (СУБД), которая обеспечивает:
  - отсутствие дублирования данных;
  - ссылочную целостность;
  - согласованность данных.



По модели данных:

- **Иерархическая модель данных** — представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами (в программировании применительно к структуре данных дерево устоялось название братья).

- **Объектно-ориентированная база данных (ООБД)** — база данных, в которой данные моделируются в виде объектов, их атрибутов, методов и классов.

- **Объектно-реляционная СУБД (ОРСУБД) —** реляционная СУБД (РСУБД), поддерживающая некоторые технологии, реализующие объектно-ориентированный подход: объекты, классы и наследование реализованы в структуре баз данных и языке запросов.

Объектно-реляционными СУБД являются, к примеру, широко известные **Oracle Database, Informix, DB2, PostgreSQL, FirstSQL/J.**

- **Реляционная модель данных (РМД)** — логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики как теории множеств и логика первого порядка.

- **Сетевая модель данных** — логическая модель данных, являющаяся расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных.

Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков.

Классификация по степени распределённости:

- Централизованная, или сосредоточенная (англ. centralized database): БД, полностью поддерживаемая на одном компьютере.
- Распределённая (англ. distributed database): БД, составные части которой размещаются в различных узлах компьютерной сети в соответствии с каким-либо критерием.
  - Неоднородная (англ. heterogeneous distributed database): фрагменты распределённой БД в разных узлах сети поддерживаются средствами более одной СУБД
  - Однородная (англ. homogeneous distributed database): фрагменты распределённой БД в разных узлах сети поддерживаются средствами одной и той же СУБД.
  - Фрагментированная, или секционированная (англ. partitioned database): методом распределения данных является фрагментирование (партиционирование, секционирование), вертикальное или горизонтальное.
  - Тиражированная (англ. replicated database): методом распределения данных является тиражирование (репликация).

По учёту времени в данных

- Пространственная (англ. spatial database): БД, в которой поддерживаются пространственные свойства сущностей предметной области. Такие БД широко используются в геоинформационных системах.
- Временная, или темпоральная (англ. temporal database): БД, в которой поддерживается какой-либо аспект времени, не считая времени, определяемого пользователем.
- Пространственно-временная (англ. spatial-temporal database) БД: БД, в которой одновременно поддерживается одно или более измерений в аспектах как пространства, так и времени.

Базы данных и СУБД

Система управления базами данных, СУБД (Database management system, DBMS) – **программная система, обеспечивающая управление базами данных**, а именно:

- обеспечение свойств БД (отсутствие дублирования данных, целостность и согласованность данных);
- взаимодействие с внешней и оперативной памятью компьютера;
- журнализация изменений в БД;
- резервное копирование БД;
- восстановление БД;
- поддержка языков запросов к БД.

Базы данных и СУБД

Клиент СУБД – программная система, предоставляющий интерфейс доступ к функциям СУБД и результатам работы СУБД.

Oracle SQLPlus

NetCracker db150.jsp

Модель OSI– это семиуровневая логическая модель работы сети. Модель OSI реализуется группой протоколов и правил связи, организованных в несколько уровней.

7. **Прикладной уровень** разрешает приложениям пользователя иметь доступ к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления.

6. **Представительский уровень** - отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально. На представительском уровне передаваемая по сети информация не меняет содержания.

5. **Сеансовый уровень** — отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень

управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия. Сеансы передачи составляются из запросов и ответов, которые осуществляются между приложениями. Службы сеансового уровня обычно используются в средах приложений, в которых требуется использование удалённого вызова процедур.

4. **Транспортный уровень** — предназначен для доставки данных. При этом не важно, какие данные передаются, откуда и куда, то есть, он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, а длинные разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка.

3. **Сетевой уровень** - предназначается для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

2. **Канальный уровень** — предназначенный для передачи данных узлам, находящимся в том же сегменте локальной сети. Также может использоваться для обнаружения и, возможно, исправления ошибок, возникших на физическом уровне. Канальный уровень отвечает за доставку кадров между устройствами, подключенными к одному сетевому сегменту. Кадры канального уровня не пересекают границ сетевого сегмента. Функции межсетевой маршрутизации и глобальной адресации осуществляются на более высоких уровнях модели OSI, что позволяет протоколам канального уровня сосредоточиться на локальной доставке и адресации.

1. **Физический уровень** — физическая и электрическая среда для передачи данных. Обычно физический уровень описывает: передачи на примерах топологий, сравнивает аналоговое и цифровое кодирование, синхронизацию бит, сравнивает узкополосную и широкополосную передачу, многоканальные системы связи, последовательную передачу данных.


Модель обмена информацией OSI
Подготовка сообщения
Форматирование, перевод, зашифровка, сжатие сообщения
Предоставление адреса и запаковка сообщения
Перевозка на почту
Принятие и сортировка
Упаковка
Загрузка
7. Прикладной (application)
6. Представительский (presentation)
5. Сеансовый (session)
4. Транспортный (transport)
3. Сетевой (network)
2. Канальный (data link)
1. Физический (physical)
Средство передачи сообщения
Чтение сообщения
Форматирование, перевод, расшифровка, сжатие сообщения
Проверка адреса и открытие сообщения
Получение на почте
Сортировка и распределение
Распаковка
Разгрузка
NetCracker
© 2013 NetCracker Technology Corporation Confidential
10

Протоколы передачи данных

Протокол – набор правил и соглашений, позволяющих провести обмен информацией между разнородными системами.

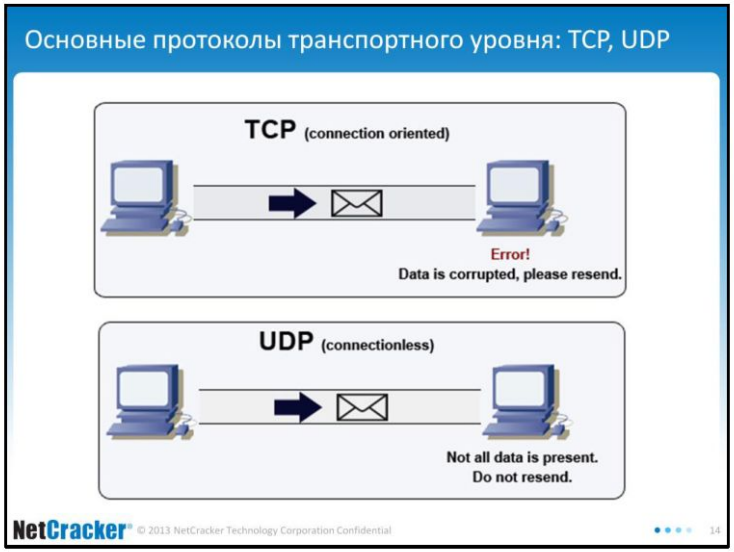
Стандартизированный протокол передачи данных также **позволяет разрабатывать интерфейсы** (уже на физическом уровне), **не привязанные к конкретной аппаратной платформе и производителю.**

TCP — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в достоверности получаемых данных, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета. В отличие от UDP гарантирует целостность передаваемых данных и уведомление отправителя о результатах передачи.

Основные протоколы транспортного уровня: TCP, UDP

UDP (User Datagram Protocol — протокол пользовательских датаграмм) — один из ключевых элементов TCP/IP, 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.

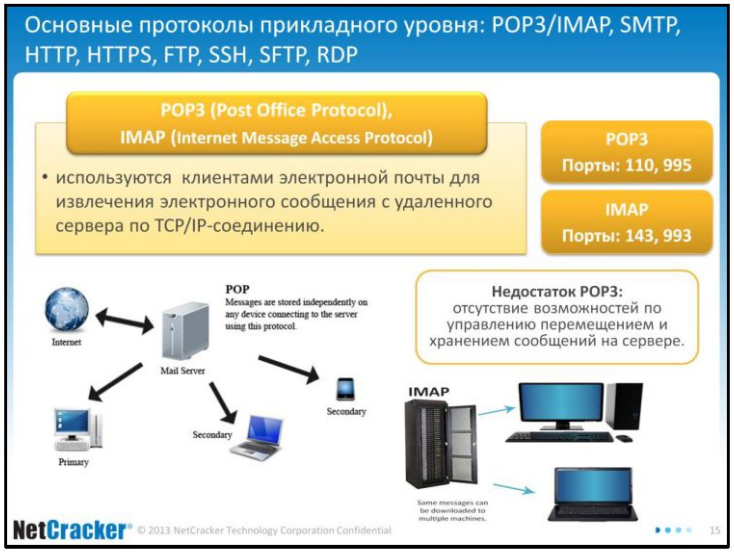
- управление передачей данных на уровне приложений;
- нет проверки факта доставки и порядка доставки данных;
- не подразумевает проверку соединения;
- широко используется в системах реального времени.



UDP — один из ключевых элементов Internet Protocol Suite (более известного как TCP/IP), 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.

UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени. При необходимости исправления ошибок на сетевом уровне интерфейса приложение может задействовать TCP или SCTP, разработанные для этой цели.

Природа UDP как протокола без сохранения состояния также полезна для серверов, отвечающих на небольшие запросы от огромного числа клиентов, например DNS и потоковые мультимедийные приложения вроде IPTV, Voice over IP, протоколы туннелирования IP и многие онлайн-игры.

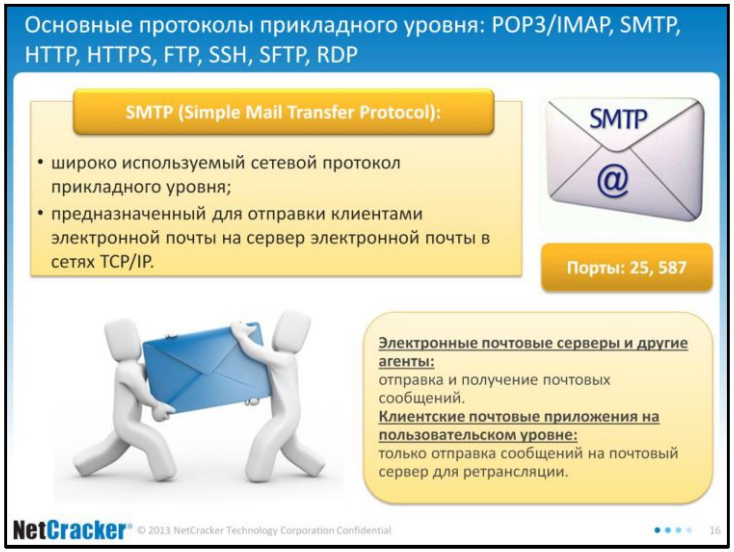


POP поддерживает простые требования «загрузи-и-удали» для доступа к удаленным почтовым ящикам. Хотя большая часть POP-клиентов предоставляют возможность оставить почту на сервере после загрузки, использующие POP клиенты обычно соединяются, извлекают все письма, сохраняют их на пользовательском компьютере как новые сообщения, удаляют их с сервера, после чего разъединяются.

Другие протоколы, в частности IMAP, предоставляют более полный и комплексный удаленный доступ к типичным операциям с почтовым ящиком. Многие клиенты электронной почты поддерживают как POP, так и IMAP; однако, гораздо меньше интернет-провайдеров поддерживают IMAP.

Протокол IMAP представляет собой альтернативу POP3.

POP3 имеет ряд недостатков, и наиболее серьёзный из них — отсутствие возможностей по управлению перемещением и хранением сообщений на сервере. Сообщения, как правило, загружаются с почтового сервера все сразу, после чего они с сервера удаляются, то есть отсутствует возможность выбирать сообщения для получения.



Электронные почтовые серверы и другие агенты пересылки сообщений используют SMTP для отправки и получения почтовых сообщений, а работающие на пользовательском уровне клиентские почтовые приложения обычно используют SMTP только для отправки сообщений на почтовый сервер для ретрансляции. Для получения сообщений клиентские приложения обычно используют либо POP (англ. Post Office Protocol — протокол почтового отделения), либо IMAP (англ. Internet Message Access Protocol), либо патентованные системы (такие как Microsoft Exchange и Lotus Notes/Domino) для доступа к учетной записи своего почтового ящика на сервере.

Основные протоколы прикладного уровня: POP3/IMAP, SMTP, HTTP, HTTPS, FTP, SSH, SFTP, RDP

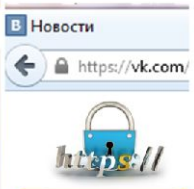
HTTP (Hypertext Transfer Protocol)

- протокол **прикладного уровня**;
- **обмен контентом** между web-сервером и web-клиентом.
- обмен сообщениями по схеме «**запрос-ответ**».
- для идентификации ресурсов HTTP использует **глобальные URI**.
- работает **поверх TCP/IP**.

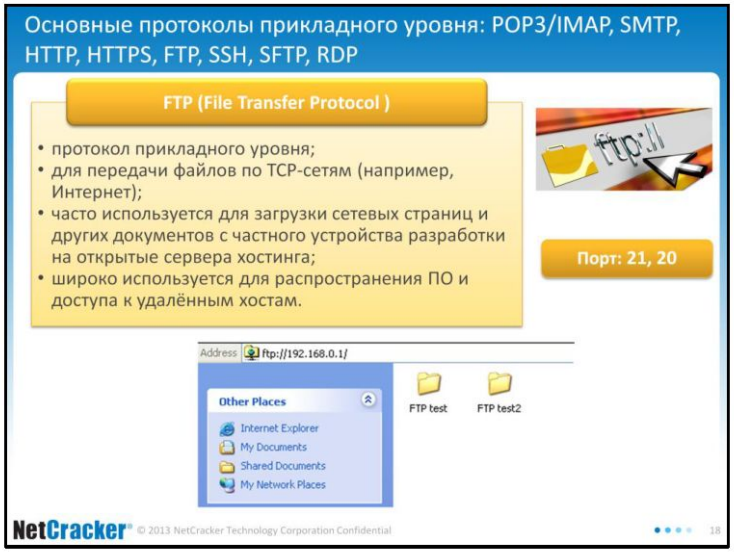
Порт: 80, 8080

HTTPS (Secure HTTP)

- **расширение протокола HTTP**, поддерживающее **шифрование**;
- **безопасный обмен контентом**;
- **защита от атак;**
- используется для приложений, в которых **важна безопасность соединения**.



Порт: 443



FTP не разрабатывался как защищённый (особенно по нынешним меркам) протокол и имеет многочисленные уязвимости в защите. В мае 1999 авторы RFC 2577 свели уязвимости в следующий список проблем:

- Скрытые атаки (bounce attacks)
- Спуф-атаки (spoof attacks)
- Атаки методом грубой силы (brute force attacks)
- Перехват пакетов, сниффинг (packet capture, sniffing)
- Защита имени пользователя
- Захват портов (port stealing)

FTP не может зашифровать свой трафик, все передачи - открытый текст, поэтому имена пользователей, пароли, команды и данные могут быть прочитаны кем угодно, способным перехватить пакет по сети. Эта проблема характерна для многих спецификаций Интернет-протокола (в их числе SMTP, Telnet, POP, IMAP), разработанных до создания таких механизмов шифрования, как TLS и SSL. Обычное решение этой проблемы - использовать "безопасные", TLS-защищенные версии уязвимых протоколов (FTPS для FTP, TelnetS для Telnet и т.д.) или же другой, более защищённый протокол, вроде SFTP/SCP, предоставляемого с большинством реализаций протокола Secure Shell.

Основные протоколы прикладного уровня: POP3/IMAP, SMTP, HTTP, HTTPS, FTP, SSH, SFTP, RDP

SSH (Secure Shell)

- сетевой протокол прикладного уровня;
- позволяет производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов);
- допускает выбор различных алгоритмов шифрования.
- позволяет безопасно передавать в незащищённой среде сетевые протоколы.
- позволяет не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры).
- может использовать сжатие передаваемых данных для последующего их шифрования.

Порт: 22

Основные протоколы прикладного уровня: POP3/IMAP, SMTP, HTTP, HTTPS, FTP, SSH, SFTP, RDP

SFTP (SSH File Transfer Protocol)

- протокол прикладного уровня, предназначенный для копирования и выполнения других операций с файлами поверх надёжного и безопасного соединения;
- использует SSH для передачи файлов;
- не связан с FTP, за исключением того, что он тоже передаёт файлы и имеет аналогичный набор команд для пользователей;
- шифрует и команды, и данные, предохраняя пароли и конфиденциальную информацию от открытой передачи через сеть;
- по функциональности похож на FTP, но так как он использует другой протокол, клиенты стандартного FTP не могут связаться с SFTP-сервером и наоборот.

Основные протоколы прикладного уровня: POP3/IMAP, SMTP, HTTP, HTTPS, FTP, SSH, SFTP, RDP

RDP (Remote Desktop Protocol)

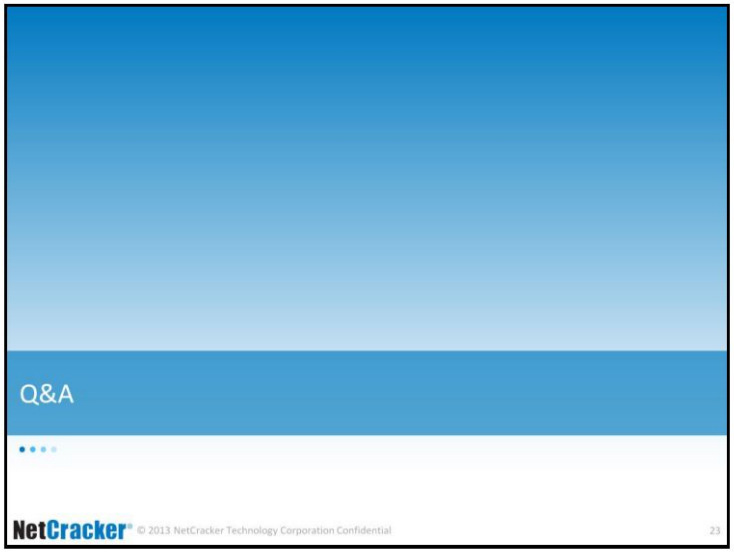
Порт: 3389

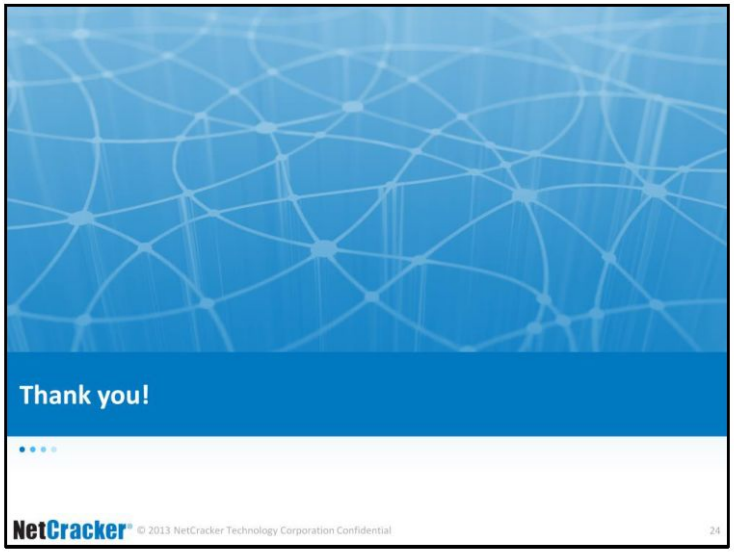
- проприетарный протокол прикладного уровня;
- протокол удалённого рабочего стола;
- купленный Microsoft у Citrix;
- используется для обеспечения удалённой работы пользователя с сервером, на котором запущен сервис терминальных подключений.

mstsc.exe – клиент в Windows 2k/XP/2003/Vista/2008/7/8

Дополнительные материалы

- http://ru.wikipedia.org/wiki/Протокол_передачи_данных
- http://fas.aics.ru/student/lectures/aiit/cli-se.pdf
- http://www.mstu.edu.ru/study/materials/zelenkov/ch_7_1.html
- http:// www.google.com


Q&A
NetCracker® © 2013 NetCracker Technology Corporation Confidential
23


Thank you!
NetCracker®
© 2013 NetCracker Technology Corporation Confidential
24