

Введение в сетевые технологии для пользователей и администраторов ГРИД

Антон Теслюк

31 января 2006, ИФВЭ Протвино

- **Цели этого курса:**

- дать пользователям и администраторам сайтов грид обзор современных сетевых технологий и сервисов, а также тенденций развития сетевых технологий в ближайшем будущем
- детально рассмотреть вопросы производительности сети при передачи больших объемов данных. **Что делать если сеть работает медленно?**
- обзор метрик и средств сетевого мониторинга для грид. **Что мерить и чем мерить?**
- обзор способов решения сетевых проблем в распределенных грид-инфраструктурах. **Что делать если сеть работает не так как положено?**

- **Обзор основных сетевых технологий:**
 - Протоколы IP, IPv6
 - IP and QoS (DiffServ, Premium IP)
 - MPLS, LightPath
 - TCP
- **Вопросы производительности при передаче больших массивов данных – TCP Congestion Control**
- **Обзор метрик и средств сетевого мониторинга для грид (EDG E2EMonit tools, NPM by JRA4)**
- **Решение сетевых проблем в глобальных информационных инфраструктурах – EGEE NOC (ENOC)**

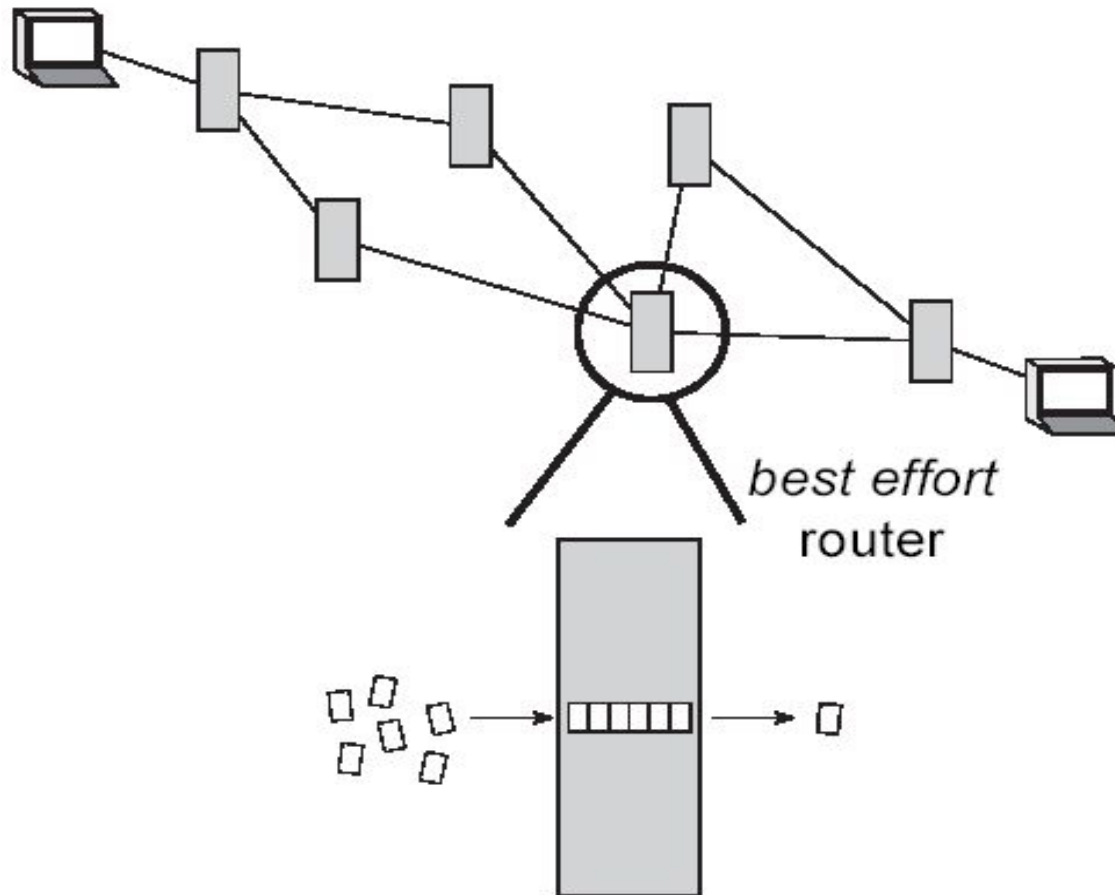
- **IP protocol**

- RFC 791, сентябрь 1981 года
- принцип коммутации пакетов
- нет гарантий доставки пакета, нет гарантий что пакеты придут в правильном порядке правильному адресату (если вообще придут ☺)
- нет встроенных средств для поддержки QoS
- размер заголовка не фиксирован
- мало адресов, распределение адресов по странам очень неравномерно (особенно мало адресов досталось Японии и Китаю)
- адресные блоки плохо агрегированы, большие таблицы маршрутизации, большие требования к производительности маршрутизаторов, дорогая маршрутизация

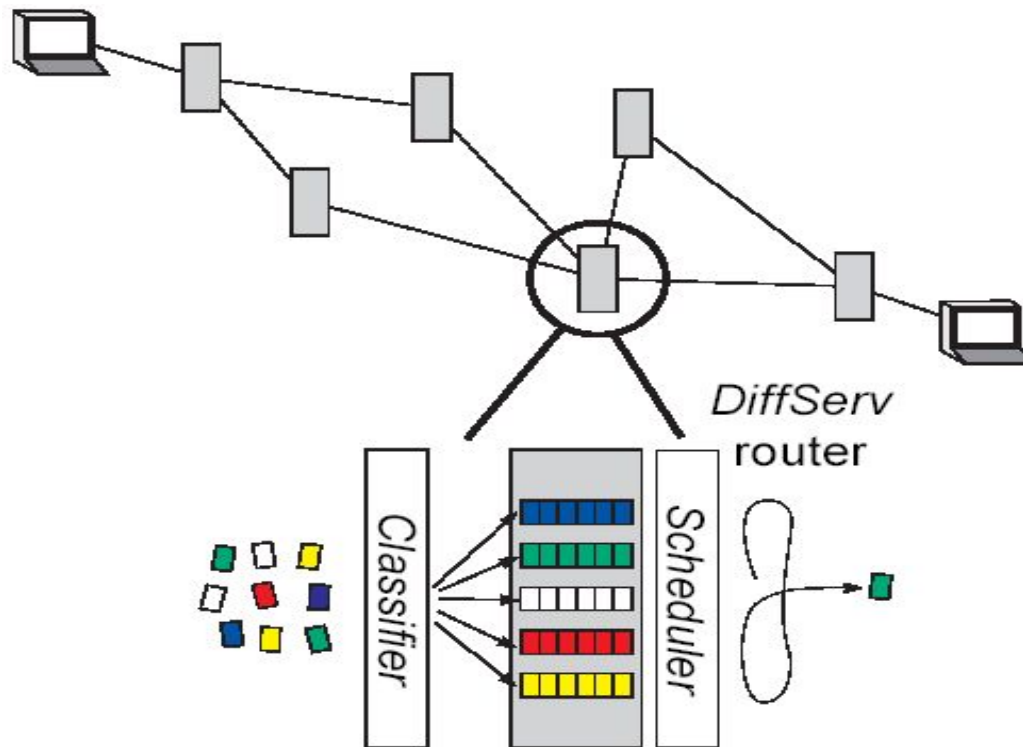
- IPv6 протокол

- RFC 2460, декабрь 1998 года
- 128 бит на адрес. **Очень** много адресов ($\sim 10^{38}$), хватит на каждый компьютер, принтер, холодильник, чайник и.т.д.
- Фиксированный размер заголовка, заголовок максимально упрощен
- Агрегирование блоков адресов (RFC 2374)
- Встроенные механизмы автоконфигурирования (IPv6 stateless autoconfiguration, neighbor discovery), поддержка мобильности (IPv6 multihoming)
- Реализован в большей части ключевого ПО (в т.ч. и в grid middleware)
- Тем не менее, реальное распространение IPv6 и полная замена им IPv4 идет с трудом. В IPv6 пока нет **killing applications**

- Обычно IP трафик – best effort traffic. Все IP пакеты эквивалентны, имеют одинаковый приоритет, обслуживаются в одной очереди маршрутизатора

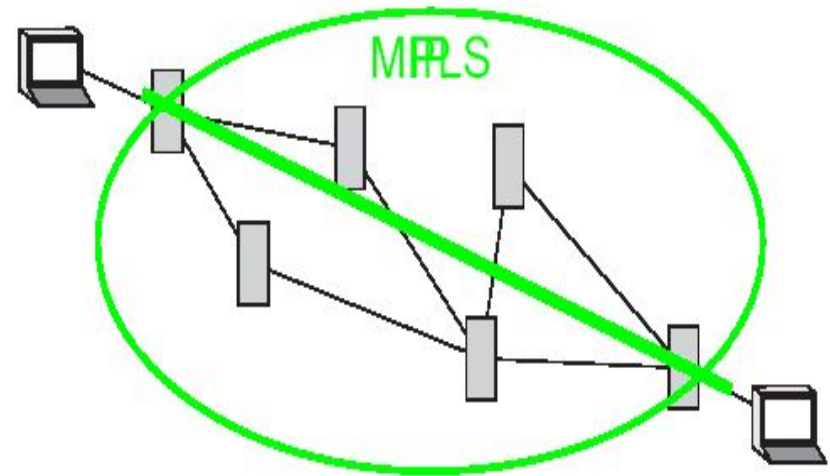


- RFC 2474, декабрь 1998 (заменяет RFC 1455 и 1349, 94 и 92 года соответственно)
- Пакеты обрабатываются маршрутизаторами в зависимости от значения поля DSCP (младшие 6 бит байта TOS в IP заголовке)



- пример реализации DiffServ – Expedited Forwarding (EF), RFC 2598 (1999 год). Premium IP в GEANT
- пакеты с DSCP = 46 обслуживаются приоритетно по сравнению с остальными
- для надежного функционирования сети объем Premium IP трафика не должен превышать 10% от общего трафика в сети
- необходим контроль за трафиком
- в GEANT помимо приоритетного класса Premium IP существует низкоприоритетный класс трафика – Less than Best Effort (LBE).

- QoS на IP уровне – это достаточно сложно и дорого
- Гарантированный сервис можно реализовать и на канальном уровне (с точки зрения IP) – MPLS
- Виртуальный канал с требуемым QoS



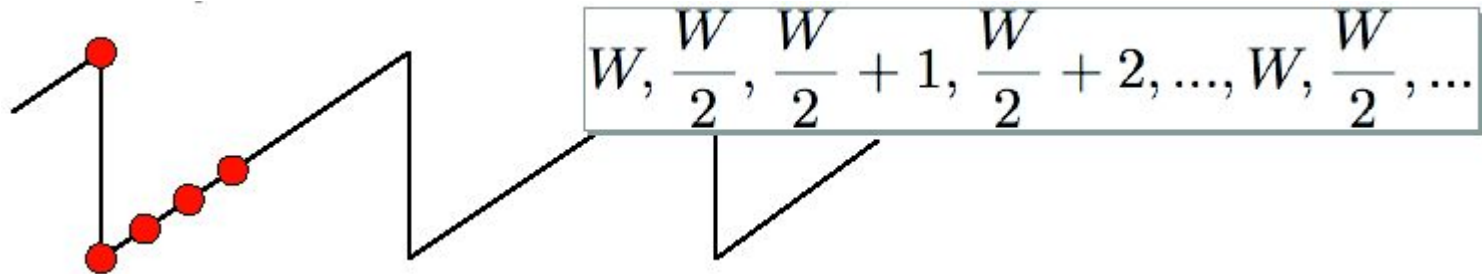
- **LightPath** – гарантированный сервис на L1
 - между двумя узлами выделенный 'light path' (например выделенная длина волны в оптическом кабеле), прямое физическое соединение

- RFC 791, сентябрь 1981
- предоставляет приложениям сервис надежной передачи двунаправленного потока данных
 - поток данных определяется IP-адресами двух узлов и парой TCP портов: ip1:port1 ↔ ip2:port2
 - TCP порты позволяют разделять трафик разных приложений (и разных потоков одного приложения)
- e-mail (smtp, pop, imap), ftp, http, **gridftp**, telnet, ssh и т.д. используют TCP
- каждый пакет имеет свой порядковый номер (sequence number)
- получатель отправляет подтверждения отправителю о получении данных

- **Пакет может прийти искаженный**
 - TCP Checksum (но всего CRC16)
- **Пакет может не дойти**
 - он будет отправлен заново
- **Пакеты могут задержаться в сети**
 - в TCP используются адаптивные таймеры
- **Пакеты могут прийти в разной последовательности**
 - с помощью порядковых номеров пакетов порядок будет восстановлен
 - в большинстве реализаций TCP частое нарушение порядка пакетов будет воспринято как перегрузку сети
 - это может быть проблемой в случае load balancing

- **Отправитель может отправлять данные быстрее чем может получить получатель**
 - в каждом TCP-пакете получатель указывает количество информации, которое он готов принять (поле Window)
- **Отправитель может пытаться отправлять больше данных, чем может передать сеть**
 - в TCP реализован механизм устранения заторов сети (end-to-end congestion control)
- **Механизм устранения заторов**
 - позволяет равномерно разделять пропускную способность сети между различными TCP потоками
 - предотвращает перегрузку сети избыточным трафиком

- **Принцип скользящего окна**
 - в сети не может находиться больше пакетов, чем размер текущего окна (congestion window)
 - принцип линейное увеличение, степенное уменьшение
 - при каждом подтверждении о доставки пакета размер окна увеличивается на один пакет (MSS)
 - если пакет теряется, то размер окна уменьшается в два раза



- **Стационарный режим одного TCP потока**

- W – максимальный размер окна (определяется узким местом в сети) в пакетах, B – размер пакета, R – RTT
- максимальная скорость передачи: $W \cdot B / R$, средняя скорость передачи: $T = \frac{3}{4} W \cdot B / R$
- вероятность потери пакета: $p = 1 / (3/8 W^2)$

- Получаем, что
$$T = \frac{\sqrt{6}B}{2R\sqrt{p}} = \frac{\sqrt{3/2}B}{R\sqrt{p}}$$

- Максимальная пропускная способность $W_{max} \leq CWND_{max}/RTT$
- Максимальный размер окна ограничен размерами TCP буфера отправителя, а также возможностями сети
 - например: $BW=1\text{Гбит/с}$, $RTT = 70\text{ ms}$, $BW*RTT = 8.75\text{ Mb}$ – оптимальный размер буфера.
 - по умолчанию TCP буфер в FreeBSD равен 32кб (в Windows XP – 17кб). $BW \leq 3.74\text{Mbps}$ (для $RTT=70\text{ms}$)
 - в заголовке TCP window size – 16 бит ($W_{max}=64\text{kb}$). RFC 1323 – дополнительная TCP option
 - для больших $BW*RTT$ очень критичны потери пакетов (двух и более). RFC 2018 – Selective Acknowledgement.
 - Path MTU discovery (RFC 1191). Необходимо использовать максимальный размер пакета.
 - 32 бита для TCP sequence number может быть мало. Для 1Gbps $T_{wrap} = 17\text{s}$. RFC 1323 – Timestamp TCP Option.
- Увеличение начального размера окна – RFC 2414

- **Kernel tuning (начиная с 2.4.xx и 2.6.xx):**
 - maximum receive window: `/proc/sys/net/core/rmem_max`
 - maximum send window: `/proc/sys/net/core/wmem_max`
 - TCP send buffers: `/proc/sys/net/ipv4/tcp_wmem`
 - автонастройка буфера при соединении:
`/proc/sys/net/ipv4/tcp_moderate_rcvbuf`
 - RFC 1323 window scaling:
`/proc/sys/net/ipv4/tcp_window_scaling`
 - RFC 2018 selective acks: `/proc/sys/net/ipv4/tcp_sack`
- **Настройки отдельного ПО:**
 - Gridftp: можно настроить socket buffer

- **Что мерить?**

- Bandwith:

- полная полоса пропускания (BW)
 - доступная полоса пропускания (ABW)
 - пропускная способность транспортного уровня (TCP, UDP)

- Круговые характеристики QoS:

- круговая задержка (RTT)
 - круговой процент потерь (Packet loss)
 - вариация круговой задержки (Jitter)

- Односторонние характеристики QoS:

- односторонняя задержка (OWD)
 - односторонний процент потерь (OWPL)
 - вариация задержки (IPDV)

- **Как мерить?**
 - Активный мониторинг
 - ICMP ping – packet loss, delay
 - Iperf – available BW
 - Пассивный мониторинг
 - измерения имеющегося сетевого трафика – ntmf
- **Чем мерить?**
 - пакет e2emonit
 - Pinger
 - Iperf
 - UDPmon
 - Интерфейс NPM
 - Веб-интерфейс для пользователя
 - Интерфейс веб-сервисов для агентов

- **Network provider user support**
- **EGEE – ENOC**
 - Для пользователя интерфейс тикетов на www.ggus.org (Global Grid User Support)
 - Решение сетевых проблем ENOC (EGEE Network Operation Centre)
 - взаимодействие с провайдерами сетевых услуг
 - сбор и анализ оповещений от сетевых провайдеров
 - решение сетевых проблем
 - инсталляция и мониторинг SLAs

THE END