Пробки в интернет
или немного об управлении трафиком

Контент бывает разный - зеленый, синий, красный...

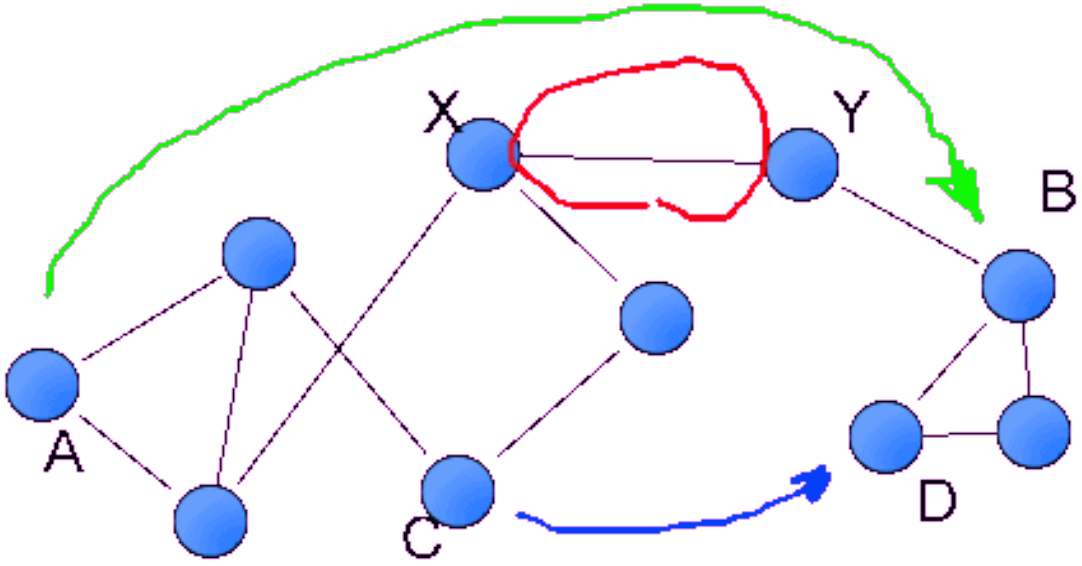
Конечные пользователи платят за возможность получать качественный доступ к любимому контенту через интернет.

Телятников Александр
Netassist
e-mail: alter@alter.org.ua

Что значит "качественный" ?

* чтобы web-странички, в т.ч. с картинками открывались быстро
* интерактивные странички обновлялись за комфортное время и не “задумывались” на несколько минут
* online-аудиопотоки не “квакали”
* online-видео не “залипало” и не “квадратило”
* в игрушках ping был ровный, лежал в допустимых пределах и не терялись пакеты
* чтобы работали Skype и ICQ
* файлы выкачивались на заявленной скорости
* и все это происходило одновременно :)

Первая сложность - отсутствие **информации о состоянии каналов**. Маршрут выбирается просто по "карте" AS-path, без учета фактического времени отклика и уровня потерь пакетов. Пора создавать **Yandex-пробки для интернет** ?



Вторая - отсутствие **политики приоритезации трафика** в случае перегрузки канала и выбора пути в зависимости от вида трафика.

С классификацией данных тоже все не так просто. Удобного алгоритма нет.

Менее очевидные проблемные места

* временные перегрузки и/или нарушение связности .
* потери пакетов в техническом трафике (в 1-ю очередь DNS)
* невозможность/отсутствие кеширования на стороне клиента.
* использование шифрования без необходимости
* наличие большого количества подгружаемых из сети объектов (картинки, скрипты, доп. запросы к серверу), необходимых для отображения одной страницы.
* “медленный” DNS
* технологические ограничения скорости передачи. Да, GPRS и DSL еще живы.

временные перегрузки и/или нарушение связности .

Нужны механизмы мониторинга загруженности и задержек в каналах в целом (по AS-path) и по отдельным сервисам.
Нужны средства ограничения исхоящего трафика в соответствии с результатами мониторинга, дабы не создавать "пробку".
Даже клиентскому ПО полезно обладать информацией о фактической доступности ресурсов.

Из практики, при ограничении потока шейпером с буферизацией уровень потерь пакетов остается приемлемым даже при перегруках. При ограничении на уровне L2 (скоростью порта) без буферизации потери начинаются уже при достижении 70-80% загрузки канала.

 # невозможность кеширования на стороне клиента.

* явная, обусловленна директивами управления кешем в HTTP-заголовках. Ряд контент-провайдеров объявляют контент некешируемым, хотя де-факто, данные всегда отдаются одни и те же.

Сюда же можно отнести вариант короткого времени жизни объекта. В этом случае получаем запросы revalidate, занимающие не столько канал, сколько время.

refresh_pattern vec.*\.maps\.yandex\.net\/tiles\? 14400 90% 20080 ignore-no-cache override-expire override-lastmod ignore-reload **ignore-auth**

refresh_pattern youtube.*videoplay 14400 90% 24400 ignore-no-cache override-expire override-lastmod ignore-reload **ignore-private**

refresh_pattern (mt|kh|pap).*\.google\.com 14400 90% 24400 ignore-no-cache override-expire override-lastmod ignore-reload **ignore-private ignore-auth**

* неявная. При использовании CDN/mirror технологий часто (как правило) при повторной загрузки страницы ссылки на объекты генерируются уже другие. Таким образом один и тот же объект многократно загружается под разными именами.

http://vec04.maps.yandex.net/tiles?
l=map&v=2.39.0&x=1197&y=693&z=11&lang=uk_UA
http://vec01.maps.yandex.net/tiles?
l=map&v=2.39.0&x=1197&y=693&z=11&lang=uk_UA
http://vec01.maps.yandex.net/tiles?
l=map&x=1197&y=693&z=11&lang=uk_UA&v=2.39.0

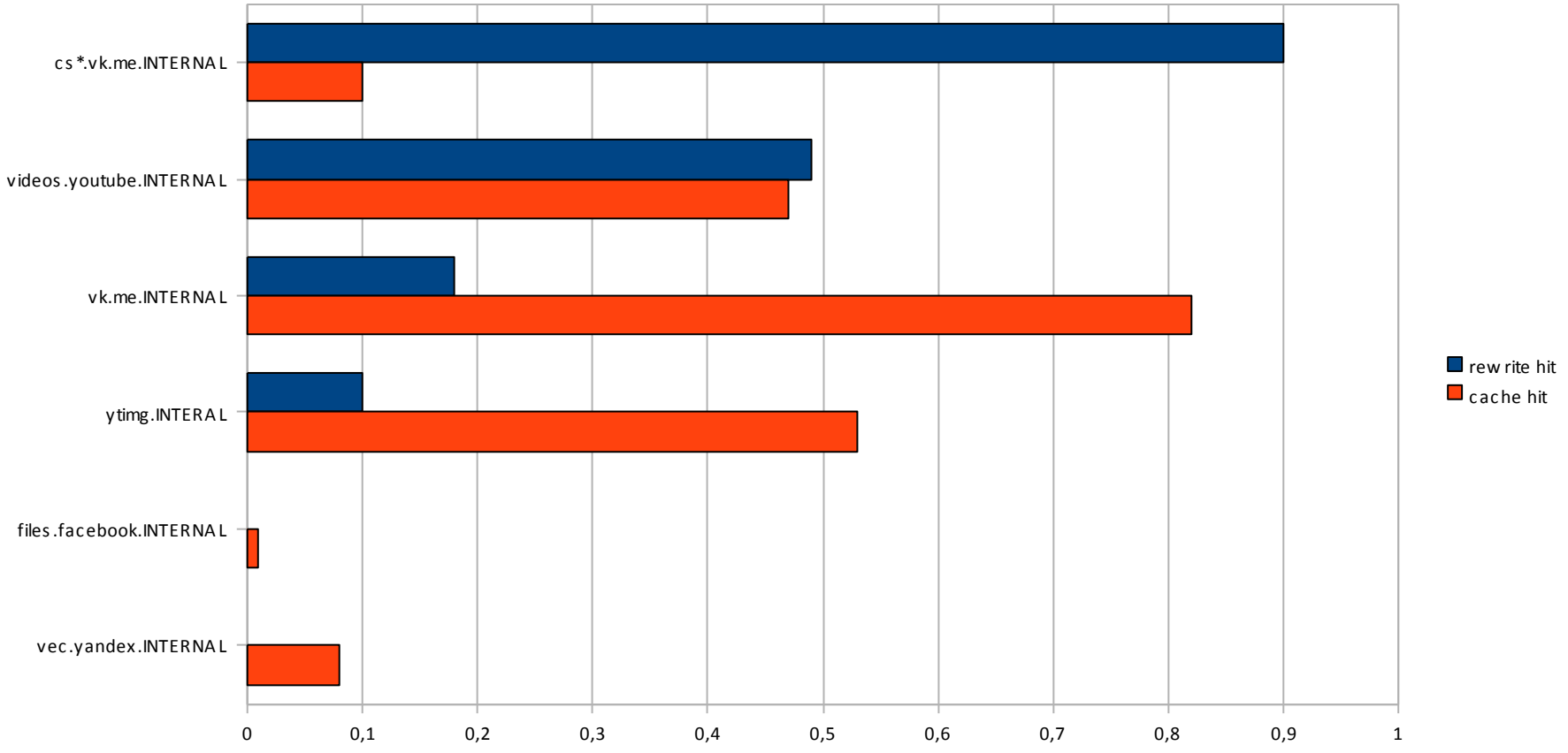
http://r12---sn-5hn7sb7k.c.youtube.com/videoplayback?algorithm=.....
...&**id=d2e4d35a7a16e4c9**&ip=162.25.15.93&....
http://r14---sn-57re8b9k.d.youtube.com/videoplayback?algorithm=.....
...&**id=d2e4d35a7a16e4c9**&ip=205.2.151.19&....

Эффективность "Склеивания" URL'ов

http://alter.org.ua/ru/soft/win/squid_url_rewrite/rewrite.pl

Host	Join-hits	Hits	Total	Efficiency, %
cs*.vk.me.INTERNAL	3357	354	3711	90
videos.youtube.INTERNAL	1818	1735	3681	50
vk.me.INTERNAL	88	401	489	18
ytimg.INTERAL	19	105	200	9
files.facebook.INTERNAL	3	31	2755	0
vec.yandex.INTERNAL	0	180	2239	0

Эффективность "Склеивания" URL'ов (rewrite.pl)



Главные потребители трафика, они же наиболее эффективно кешируемые (**только принудительно!**):

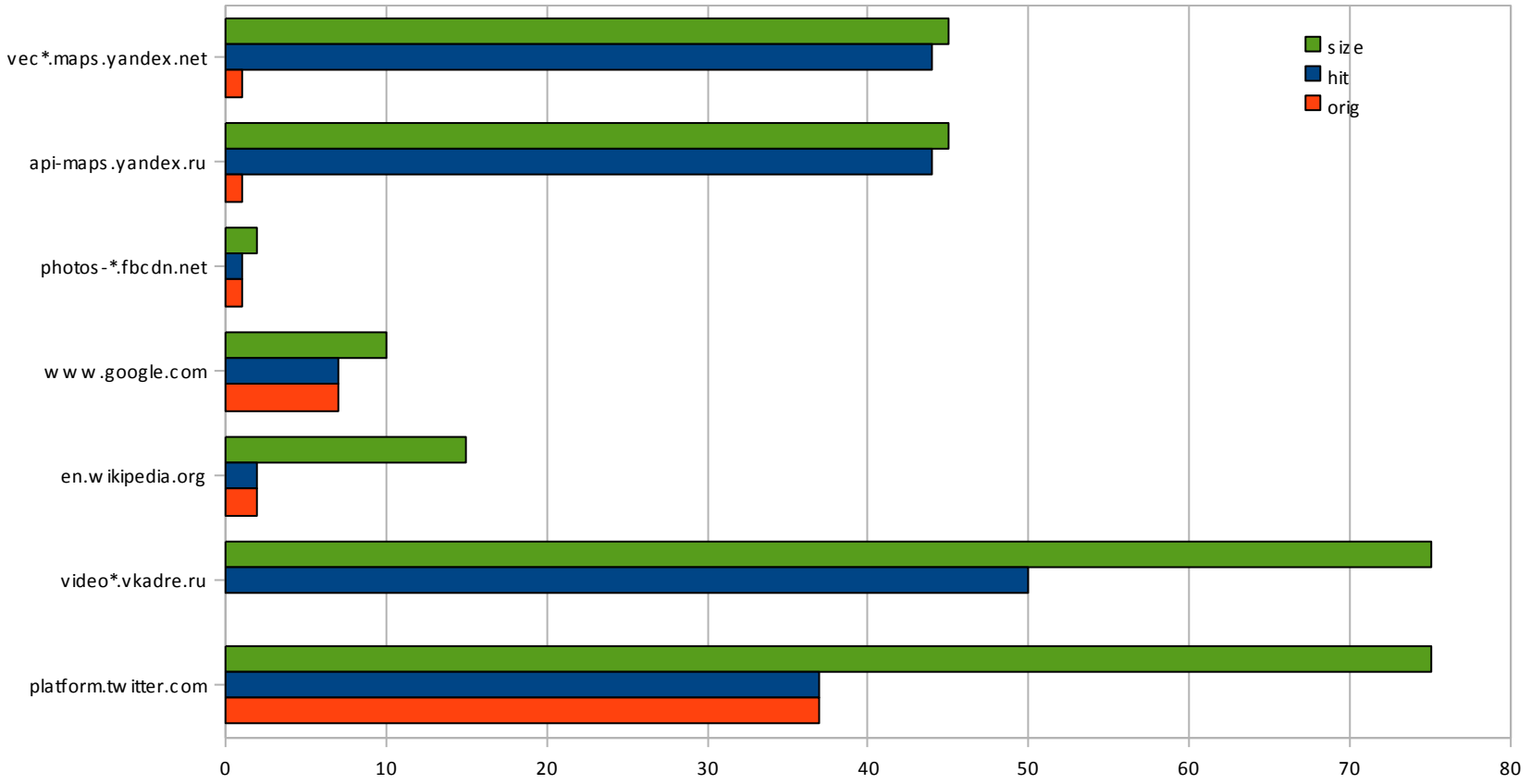
http://alter.org.ua/ru/docs/net/http_caching/

Domain	Traffic Efficiency, %	Hit Efficiency, %	Original cache efficiency	Weight
cs*.vk.me	23	37	0,00%	88,00%
youtube.com	78	94	0,00%	4,00%
st*.vk.me	99	98	0,00%	1,00%

Другие знакомые ресурсы

Domain	Traffic Efficiency, %	Hit Efficiency, %	Original cache efficiency	Weight
vec*.maps.yandex.net	45	44	1,00%	
api-maps.yandex.ru	45	44	1,00%	
photos-*.fbcdn.net	2	1	1,00%	
www.google.com	10	7	7,00%	1,00%
en.wikipedia.org	15	2	2,00%	
video*.vkadre.ru	75	50	0,00%	1,00%
platform.twitter.com	75	37	37,00%	

Другие знакомые ресурсы (графики)



В общем объеме на HTTP приходится около 10-30%. Из них на web-серфинг - тоже ~10%. Т.е. ради экономии 1-2% трафика строить что-то специальное смысла нет.
А для улучшения качества сервиса - смысл есть. В случае перегрузки канала или самого ресурса - тоже.

Отдельный момент - процент запросов, обработанных proxy без обращения к внешним серверам, в т.ч. без REFRESH_HIT. Чем больше время жизни картинки в кеше — тем более счастлив будет клиент.

А для самих контент-провайдеров есть, наверное, повод задуматься. Если принудительный кеш экономит от **30% до 95% полосы** и вычислительных ресурсов, может это повод таки позволить клиентам кешировать данные ?

 # наличие большого количества подгружаемых из сети объектов (картинки, скрипты, доп. запросы к серверу), необходимых для отображения одной страницы. Тут есть сразу несколько моментов:

* время ответа DNS
* время ответа сервера
* (не)способность браузера выполнять параллельную загрузку.
* (не)использование параллельного исполнения внутри самой страницы (jquery) или приложения

Классические решения

* Расширить канал. Можно, дорого и как правило, только свой.
* Пиринг, альтернативные каналы и включение в точки обмена. Вопрос оптимального распределения трафика между каналами остается открытым.
* Кеширование контента. Копия хранится как можно ближе к клиенту (а то и у него самого). Но как показало наше исследование — вопросов много.
* Использование jumbo-фреймов. В теории это позволяет снизить нагрузку по pps за счет увеличения MTU. На практике получается, что большинство клиентов использует MTU <= 1500.
* Кеширующий DNS на уровне ISP

Что вообще нужно сделать ?

* обеспечить прохождение критического трафика с минимальными задержками и без потерь даже при потере части каналов.
* ограничить некритичный исходящий трафик в соответствии с реальной пропускной способностью канала до получателя
* балансировать трафик между каналами
* оптимизировать формат подачи контента для эффективного кеширования
* по возможности замыкать трафик на внутренней сети, в т.ч. путем кеширования
* обеспечить возможность выбора наиболее подходящего источника данных или наиболее подходящего пути к нему
* использовать пакеты как можно большего размера. По возможности аггрегировать данные.

Что можно делать прямо сейчас

* "правильный" **шейпер/QoS** с очередью пакетов на **исходящем канале**. Полезно всем участникам при достижении нагрузки на канал ~70%. К примеру для клиента-домосетки это может выглядеть так:

- 1% под DNS, SSH, ICMP
- 5% под игрушки, Skype,
- еще 5% - web запросы и доступ к БД
- еще 5% - доступ к online video
- Остальное - как придется.

* "правильный" **шейпер/QoS** с очередью пакетов на **входящем канале**, чтобы протоколы, контролирующие скорость передачи, притормозились. Полезно на уровне ISP и транспорта, тоже при достижении ~70% нагрузки. Для клиента-домосетки это может выглядеть так:

- 1% под DNS, SSH, ICMP ответы
- 5% под игрушки, Skype,
- еще 10% - web ответы и доступ к БД
- еще 20% - online video сервера
- Остальное - как придется.

* CDN.
* Сделать возможным эффективное кеширование контента.
* оптимизация формата (объема) контента в соответствии с каналом пользователя (GPRS, DSL, FastEthernet, etc.)
* использование p2p контент-провайдерами.
* локальный torrent ретрекер и torrent peer policy (retracker.local, peerpolicy.local) внутри ISP
* кеширование HTTP, DNS. Возможно, кеширование/ретрекинг torrent
* использовать, где это возможно, протоколы маршрутизации, позволяющие быстро реагировать на изменения маршрутов.

Чего не хватает ?

нет механизма ограничения входящего трафика и способа получения актуальной информации о загруженности промежуточных каналов и канала принимающей стороны. Это мог бы быть спец. **network load discovery** протокол. В этом случае необходимо будет обеспечить прозрачное прохождение через сети, не поддерживающие данный протокол.

нет способа эффективно и автоматизированно отделять критический трафик, нужны access-list'ы и списки ресурсов. Была бы полезна общедоступная база серверов и сервисов (tcp/udp + ip + port range) для построения acl'ов. Либо стандартизация номеров портов (наиболее легко реализуемо).

Чего не хватает ?

нет способа эффективно отделять кешируемый (условно статический) HTTP контент от динамического, часто используемый от редкого. При наличии такой информации ISP и транспортные операторы имели бы возможность сократить время отклика ресурсов и сэкономить каналы. А больше всех выиграют поставщики контента, поскольку работа по оптимизации доставки данных будет распределена по всей цепочке.

не хватает технологии объединения мелких пакетов, следующих в одном направлении в jumbo-фреймы. Это можно делать как по MAC-адресу следующего узла, так и по MPLS route ID.

Чего не хватает ?

не хватает интеграции p2p-протокола(ов).

Это позволило бы частично переложить задачу кеширования контента на локальную сеть, и тем самым значительно ускорить скорость загрузки. Кроме того, p2p существенно менее чувствителен к потерям пакетов в сети.

а может даже отдельного CDN сервиса на уровне крупных сетей передачи данных, ISP и IX

клиентского набора ПО, работающего “из коробки”

Кому это вообще нужно и зачем ?

* провайдерам - чтобы спать спокойно и слушать похвальные отзывы от довольных пользователей

* для транспортных операторов и точек обмена это возможность заработать на доп. сервисах и качестве связи

* для контент-провайдеров - способ более качественно доставить контент конечным пользователям и уменьшить нагрузку на свою сеть и оборудование. Также, знание пропускной способности канала клиента можно эффективно оптимизировать формат предоставляемого контента для комфортного просмотра.

* для всех - более равномерно распределить нагрузку по сети.

* еще одним плюсом будет защита от части DDoS-аттак, т.к. трафик в направлении сети-жертвы будет ограничиваться еще на выходе из сетей-источников.