Телеком-эволюция: Интернет третьего поколения

Первым поколением было коммутация каналов, когда для соединения Алисе приходилось изыскивать нужный провод и просить подключиться к нему Боба. Второе поколение - коммутация пакетов, когда Боб, отвечая Алисе, разбивает сообщение на тысячи мелких кусочков, а потом Алиса собирает их обратно в единое целое...

Разумеется, невозможно жить в одном поколении. Коммутация каналов в осмысленной форме появилась с изобретением первой АТС Элмоном Строуджером, еще в XIX веке, а до уровня почти повсеместного использования концепция (в автоматическом воплощении!) шла немногим меньше 50 лет - только после Второй мировой войны автоматические станции стали привычными и понятными везде, а не только в продвинутых крупных городах.

> *Забавная история вышла с изобретением АТС: Строуджер был владельцем похоронного бюро в городе Канзас-сити и терпел убытки при получении заказов по телефону, так как телефонисткой на станции работала жена его прямого конкурента, — владельца другой похоронной компании. Телефонистка направляла все звонки абонентов, вызывавших похоронное бюро, своему мужу. Элмон Строуджер поклялся навсегда избавить общество от телефонисток и изобрёл автоматический телефонный коммутатор декадно-шагового типа ёмкостью до 99 абонентов на базе декадно-шагового искателя. Так инженер победил коррупцию.*

Коммутация пакетов в первом приближении появилась в 1960-х годах. К 1970-м появилось понимание значимости изобретения, и только в 1980-х появились массовые реализации.

Не смотря на заверения Президента, Интернет все же родился не в недрах ЦРУ, но на задворках Пентагона: интернет был предназначен для сверхнадежной передачи данных в условиях термоядерной войны. А поскольку самым тонким местом для поражения всей сети коммутации каналов была и является собственно АТС, то коммутация пакетов была призвана, прежде всего, эти уязвимые узлы, если не исключить совсем, то хотя бы зарезервировать максимально возможное количество раз.

В 2000-х коммутация пакетов окончательно победила в телекоммуникационной сфере коммутацию каналов. Хотя последние останки некогда могущественной технологии еще теплятся и барахтаются (в иных случаях даже пытаются законодательно продлить свое существование), но в 2010-х уже понятно, что маятник в обратную сторону не качнется, а скорее последует спираль, приводящая в мир новую концепцию.

И это случилось. Разумеется, все пока находится в фазе прототипирования, исследований и обкатки математических моделей. Но проблематика у существующих телекоммуникационных моделей следующая:

Для модели коммутации каналов недостатки очевидны - при росте сетей и количества узлов сложность коммутационной подсистемы растет экспоненциально, а усложнение алгоритмов коммутации невозможно без фундаментальной перестройки узлов коммутации.

Коммутация пакетов позволяет более эффективно использовать физическую инфраструктуру и более высокую доступность узлов в сети. При перегрузке вы теперь не услышите "сеть занята" - просто канал станет несколько медленнее, но данные будут передаваться, пусть и со значительными задержками.

Но, тем не менее, проблема с перегрузками сети все же имеется. Причина в том, что одни и те же данные в модели хост-сервер-хост передаются многократно, а узким бутылочным горлышком IP-сетей являются локализованные средства хранения.

Представим себе для целей иллюстрации, что некий гиперпопулярный контент хранится на серверах в дата-центре штата Иллинойс (ну, или города Тамбов, если хотите). Количество желающих приобщиться к этому контенту растет с каждой секундой, сервер, отдающий запрашиваемые пакеты информации в какой-то момент начинает переставать их отдавать, сваливается и отдает код Timeout.

Владельцы высоконагруженных ресурсов научились худо-бедно бороться с этой проблемой всевозможными техническими методами: увеличением пропускной способности каналов, повышением вычислительной мощности и эффективности программного кода, отдающего контент. Специалисты-математики добились впечатляющих результатов в области сжатия данных.

Впрочем, есть еще один путь - передать функции передачи, хранения и поиска информации на сторону потребителя. Так появились сервисы Peer-to-peer (P2P) - "равный-к-равному".

Сеть в этом случае функционирует следующим образом: присутствует некоторое количество машин,

при этом каждая может связаться с любой из других. Каждая из этих машин может посылать запросы другим машинам на предоставление каких-либо ресурсов в пределах этой сети и, таким образом, выступать в роли клиента. Будучи сервером, каждая машина должна быть способной обрабатывать запросы от других машин в сети, отсылать то, что было запрошено.

Каждая машина также должна выполнять некоторые вспомогательные и административные функции (например, хранить список других известных машин-"соседей" и поддерживать его актуальность).

Разумеется, и у такого способа организации сети есть недостатки, и существенные: например, любой член данной сети не гарантирует свое присутствие на постоянной основе. Он может появляться и исчезать в любой момент времени. Но при достижении определённого критического размера сети наступает такой момент, что в сети одновременно существует множество серверов с одинаковыми функциями. Впрочем, и величина сети не гарантирует потери информации, причем особенно уязвим контент не сильно востребованный в данный момент

Имеются и глубокие проблемы с поисковыми механизмами, защитой контента от действий злоумышленников и остается не решенной проблема жесточайшей неэффективности использования сетевых ресурсов.

Последний фактор настолько возмутителен, что операторы даже начали бороться с таким "паразитным", по мнению операторов, трафиком. Действительно, пиринговые сети, все эти Torrent и DC++ при первой же возможности забивают сети мусорным трафиком, пользователи, имеющие терабайтные хранилища данных считают своим долгом тут же эти терабайты загрузить информационным мусором, копии фильмов и музыки расходятся миллионами цифровых копий, 90% из которых никогда не откроют с локального диска.

Все же, использование трети всего upstream-трафика на "торренты" - это довольно много:

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	33.20%	YouTube	19.27%	YouTube	17.38%
2	HTTP	10.07%	HTTP	17.46%	HTTP	16.26%
3	YouTube	7.67%	BitTorrent	11.10%	BitTorrent	14.71%
4	SSL	5.63%	SSL	6.19%	SSL	6.10%
5	Skype	4.54%	Facebook	3.88%	Facebook	3.95%
6	Facebook	4.29%	RTMP	3.66%	RTMP	3.27%
7	eDonkey	3.64%	MPEG	3.54%	MPEG	3.21%
8	Dropbox	2.11%	Netflix	3.23%	Netflix	2.98%
9	MPEG	1.51%	Flash Video	2.37%	Flash Video	2.17%
10	iTunes	1.30%	iTunes	2.23%	iTunes	2.08%
		72.66%		70.69%		70.01%

sandvine

Table 6 - Top 10 Peak Period Applications - Europe, Fixed Access

Хотя и ситуация меняется - с каждым годом меняется в сторону уменьшения использования P2P большинством пользователей, но актуальность "резать торренты" в случае их неадекватно большой величины остается.

Итак, мы подходим к цели нашего повествования - к сетям, построенным на совершенно другом принципе, к контентно-центричным сетям нового поколения.

Named data networking

"Сеть именованных данных" (наиболее адекватный перевод термина, который я не сразу даже и перевел) - явление весьма новое. Настолько, что на русском языке материалов почти нет, да и на английском материалы стали публиковаться совсем недавно. Собственно, сама концепция NDN была представлена общественности в 2006 году американским компьютерным исследователем Ван Якобсоном - одним из разработчиков TCP/IP стека вообще, и автор многочисленных алгоритмов передачи данных (например, утилиты traceroute и tcpdump - дело его рук).

Работа над архитектурой NDN все еще идет, и с даже небольшой долей вероятности трудно сказать, когда она будет закончена - официальный сайт (http://named-data.net/) дает лишь общие черты хода

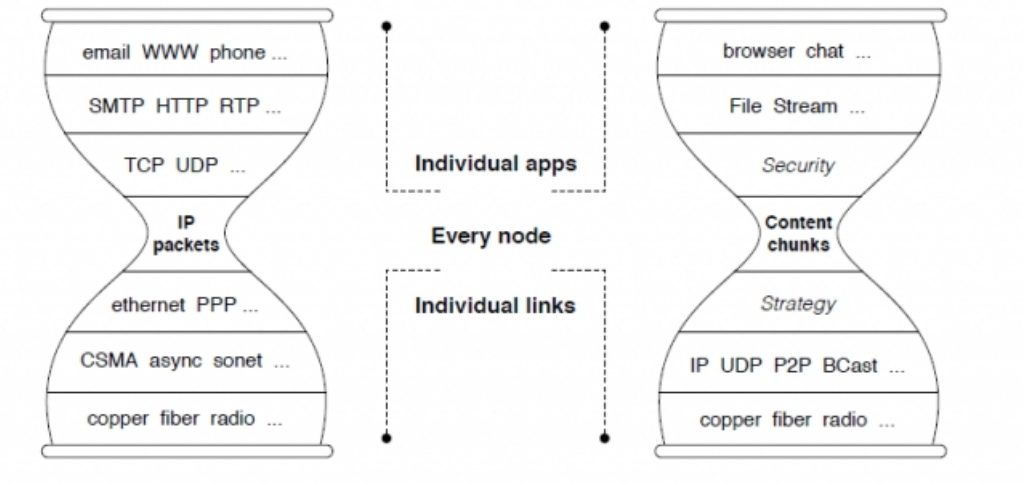
проекта, а текущая версия официальных документов находится в 0.2.0 версии. То есть, о реальных внедрениях даже речи не идет. Но кое-какие успехи в моделируемой среде все же имеются.

Ну, и финансирование проекта в размере $13,5 миллионов тоже получено. Основным держателем проекта является Южно-Калифорнийский Университет Лос-Анжелеса (UCLA - кто бы мог подумать, что не Сколково), а спонсором - угадайте с первого раза? Конечно Cisco.

Однако сама идея не так уж и нова - NDN базируется на концептуальном подходе Content Centric Networking, которая, в свою очередь, базируется на идее Теда Нельсона Project Xanadu - самом долгостройном софтовом проекте в мире, который длился 54 (!) года. Так что очень свеже-революционно-прорывной идею NDN тоже назвать нельзя. Впрочем...

Итак, как это работает:

Почти все материалы по ликбезу в области NDN начинаются с вот такой картинки:



Где противопоставляются две идеи: традиционная TCP/IP и, разумеется, NDN.

Разница в подходах заключается в том, что основным и главным блоком архитектуры NDN являются не атомарные пакеты, на которые разбивается исходная информация в TCP/IP, а самодостаточные кусочки контента, значимые сами по себе. Если для TCP/IP важным является факт организации сеанса связи (метафора, оставшаяся от коммутации каналов), то в концепции NDN эти кусочки имеют собственное имя (по этом и "именованные") и могут принципиально "жить" сами по себе, в отрыве от метафоры "хост-сервер".

Поначалу этот подход кажется просто взрывающей мозг запредельной абстракцией. Я садился за написание этой статьи несколько раз, перечитывал доступные материалы, силился понять, как же в итоге это все может функционировать. Честно признаюсь - так и не дошел до конечного просветления. Но попробую (как тот "профессор", который шесть раз объяснял студентам, сам понял, а они нет) объяснить, не вдаваясь в матан маршрутизации и практических схем реализации.

Во-первых, для NDN необходимы специальные коммуникационные приложения, которые на песочных часах находятся в верхней части - они запрашивают и получают данные, и как-то их обрабатывают с целью взаимодействия более верхнего уровня (показать человеку, например, хотя и не обязательно). В нижней части часов находится медиа-инфраструктура, которая, собственно, базируется на дейтаграммах, сиречь, старом добром IP, но в широком смысле, потому что тут могут быть и какие-то новые технологии и алгоритмы.

Во-вторых, пакеты в концепции NDN бывают двух типов:

[Телеком-эволюция: Интернет третьего поколения]

... Interest и Data. Впрочем, оба типа пакетов имеют собственные, которые идентифицируют часть данных передаваемых пакете.

Коротко о содержании NDN-пакетов:

Interest Packet (Интерес): потребитель /волшебным образом/ помещает имя требуемой порции данных в пакет и отправляет его в Сеть. Маршрутизаторы используют это имя для того, чтобы направить "интерес" в сторону производителя(ей) этих данных.

Data Packet (Данные): После того, как "интерес" достигает узла, который содержит запрошенные данные, узел должен вернуть пакет данных, который содержит как имя, так и собственно

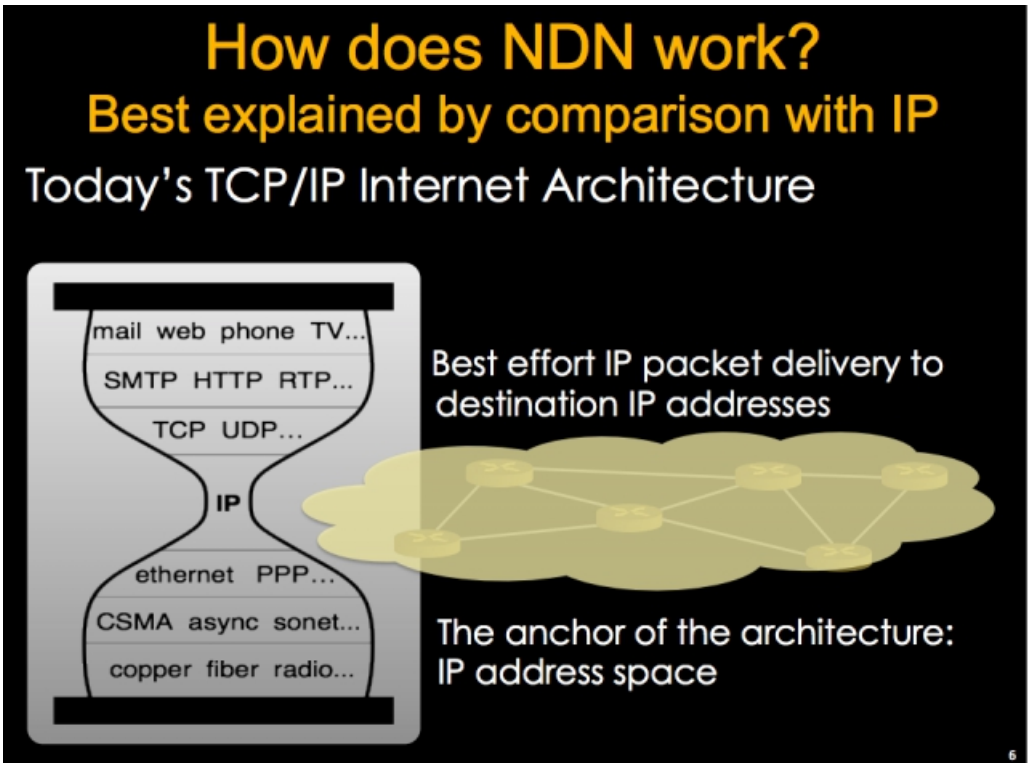
запрашиваемый контент. Кроме того, в Data Packet включена сигнатура подписи "производителя", чтобы заверить получателя, что это именно та самая запрашиваемая информация.

Маршрутизация же всех пакетов представляет собой отдельную сетевую магию, которая заключается в том, что потребителю совершенно не принципиально, где именно хранится запрашиваемая информация - мы просто отправляем интерес "в сеть" и получаем данные.

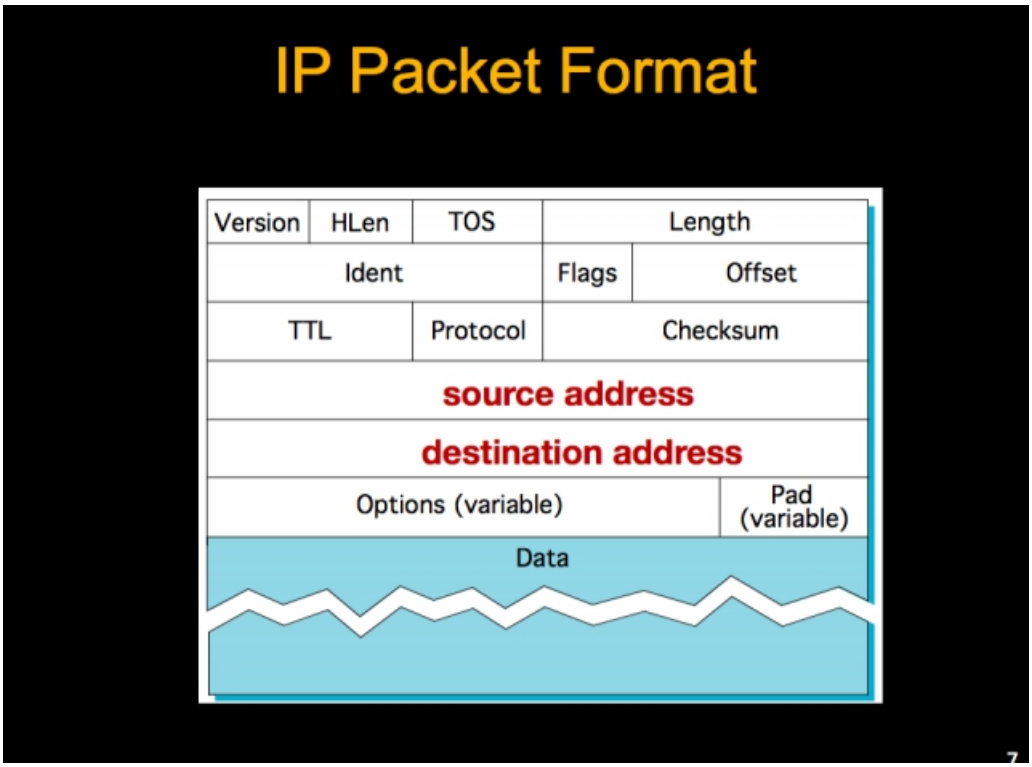
Попробуем продемонстрировать принцип в картинках от первоисточника. Целиком презентацию можно пролистать прямо здесь:

Но для тех, кто плохо владеет английским, избранные слайды с комментариями приведу - в конце концов, профессор сам должен понять предмет.

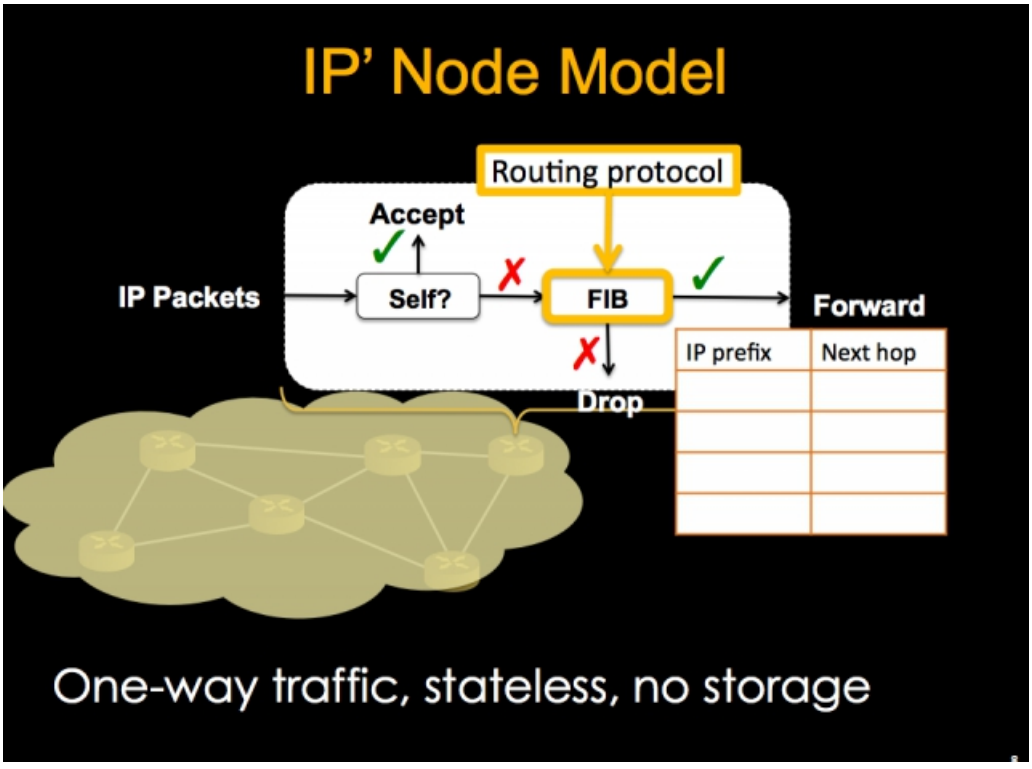
Итак, сегодня Интернет, как "сеть сетей", базируется на стеке протоколов (там много протоколов, для гуманитариев) TCP/IP, суть которого сводится к тому, чтобы в огромной сети адресовать запрос и получить ответ от конкретного IP-адреса. Ключ технологии - адресное пространство IP.



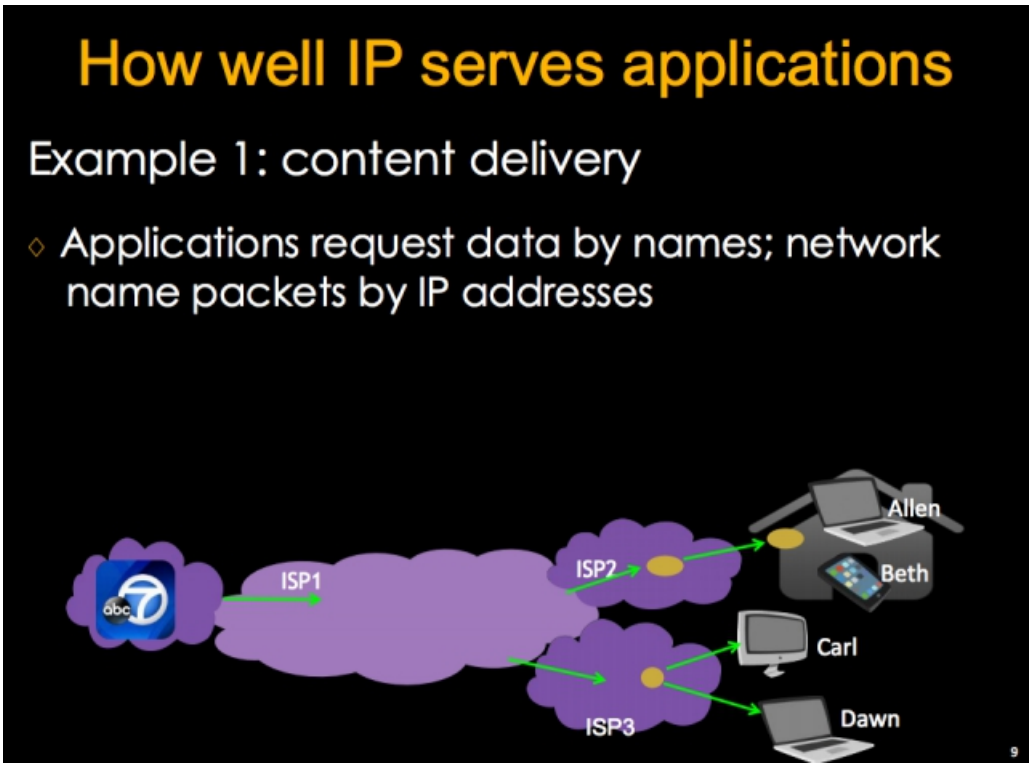
Каждый IP-пакет имеет определенный стандартами формат, главные составляющие которого - адрес отправителя информации и адрес получателя:



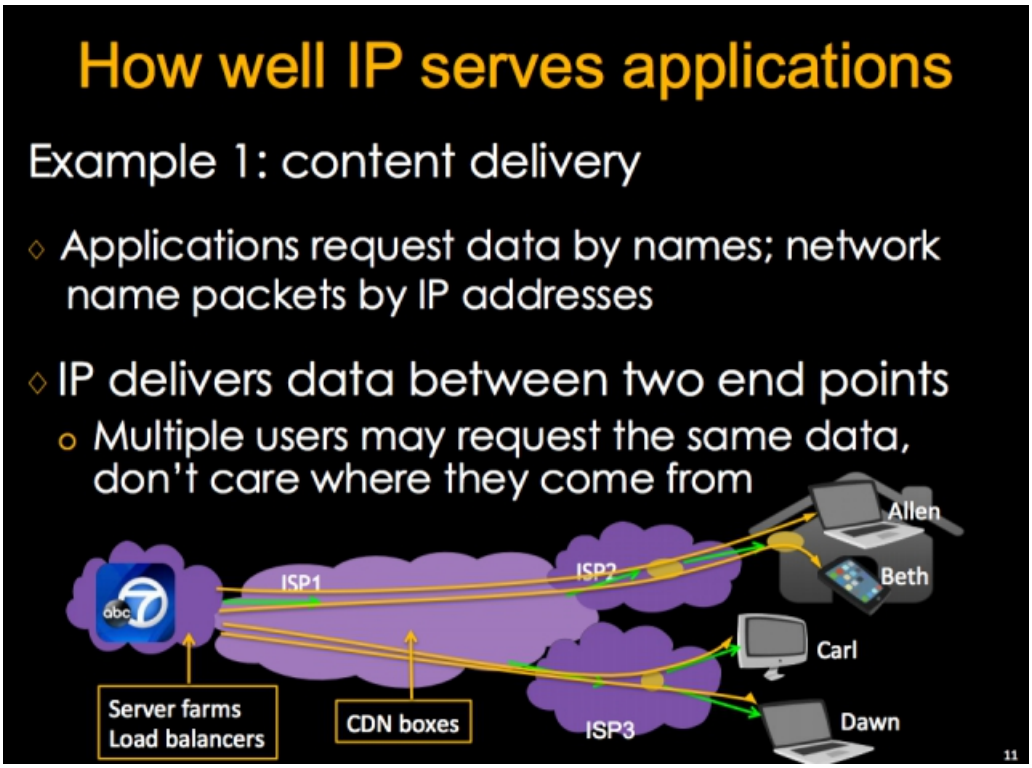
В процессе передачи данных, каждый пакет проходит определенный маршрут, который задается маршрутизаторами в соответствии с логикой, установленной владельцем этого устройства. В процессе следования маршруту, все пакеты а) нигде не хранятся, а только транслируются и б) вообще не обязательно, что дойдут до адресата.



Посему, в сеть идет не запрос информации, а запрос по конкретному адресу. В свою очередь, "конкретный адрес" отправляет запрошенную информацию в ответ. Интересно, что из разных сетей, различных поставщиков, процесс передачи может отличаться по определению - разные маршруты, разные способы переадресации пакетов, разные временные задержки. Но источник - один.

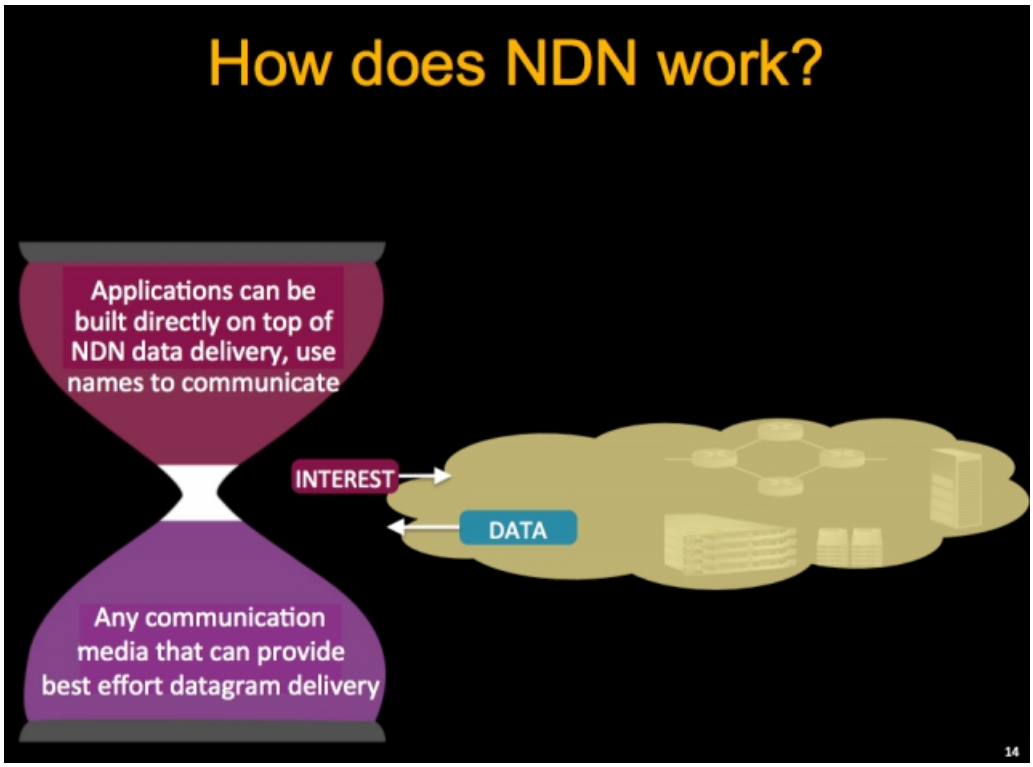


Для того чтобы обслуживать из одного адреса огромное количество запросов (представьте себе, что на один адрес https://facebook.com ежесекундно обрушивается миллиард запросов от миллиарда пользователей), в текущей архитектуре построения Интернет, приходится прибегать к ухищрениям в виде различных подсистема распределения контента (content delivery) с серверными стойками и дата-центрами, CDN-системами, кеширующими серверами, и прочими proxy.



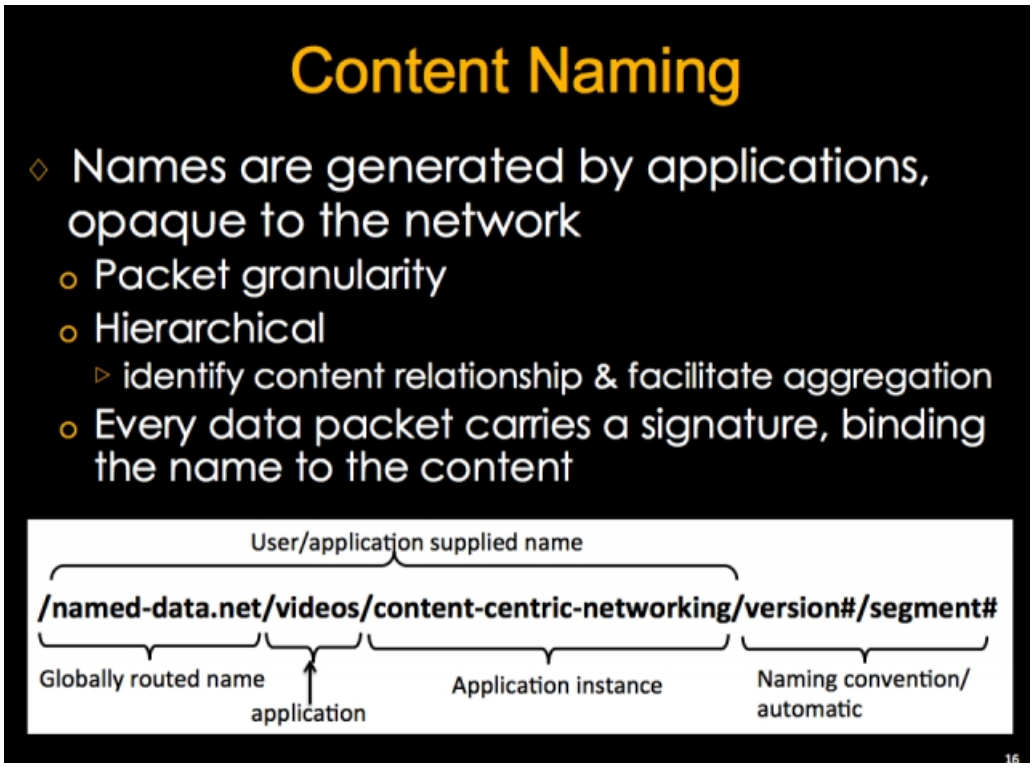
И чем больше потребителей в Сети, чем больше запросов они отправляют, тем сложнее перебрасывать эти пакеты через расстояния буквально планетарного масштаба.

Собственно, для этого и изобретают сеть нового поколения - чтобы более эффективно использовать вычислительные, коммуникационные и хранящие данные ресурсы. В общем виде, в NDN запрос к данным должен происходить прозрачно для пользователя - отправил "интерес" -> получил данные:



Разумеется, "интерес" готовит специальная программа-приложение, которая умеет правильно строить интересы и отображать соответствующие ответы (повторяюсь).

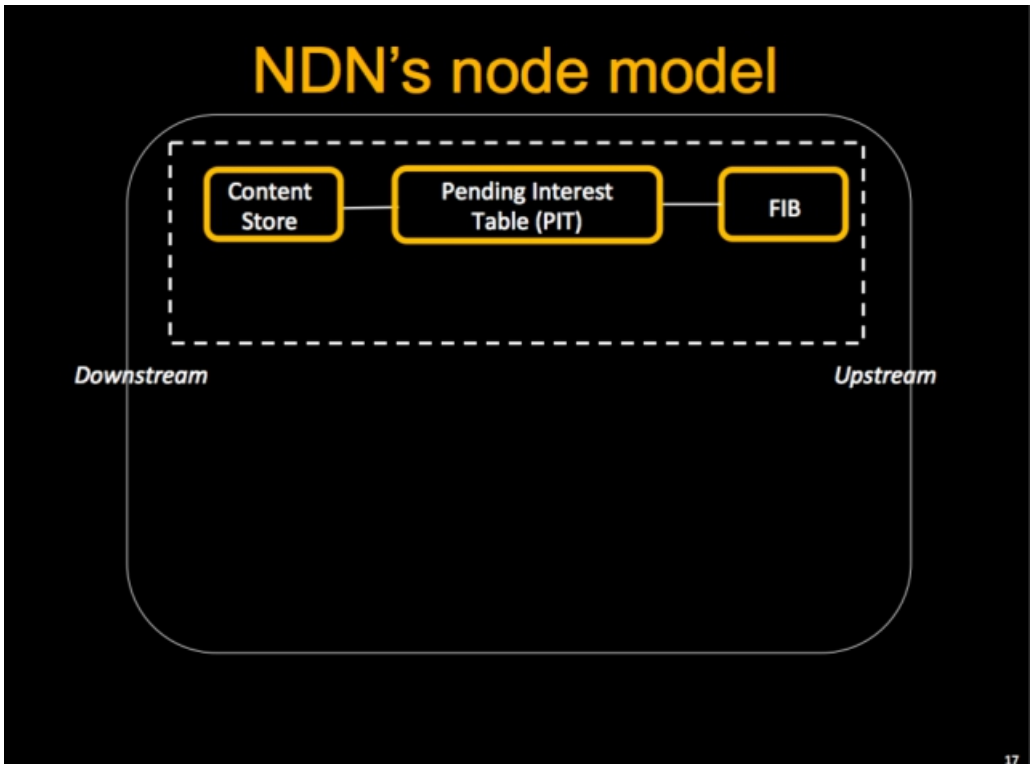
Структуру пакетов в очень общем виде мы уже видели, а вот как формируется запрос к конкретным данным и его общий вид, можно продемонстрировать на следующем слайде:



Имя контента генерируется приложением, в соответствии с определенными правилами и протоколами (тысячи их!).

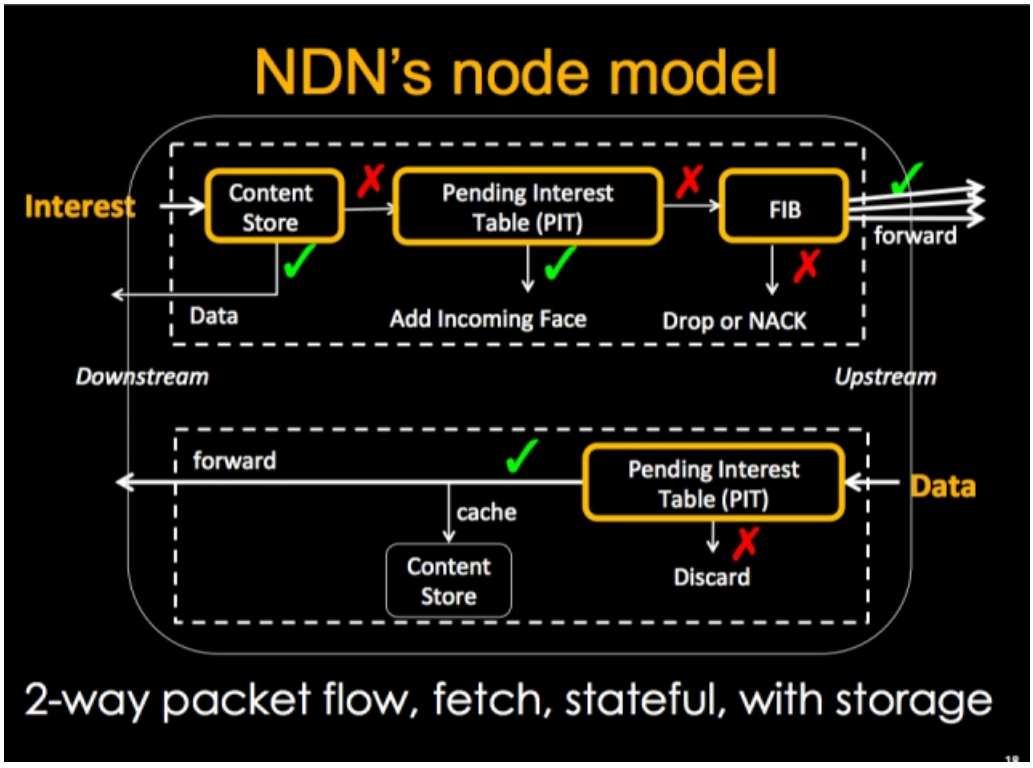
Каждый пакет - гранулирован (что бы это ни значило).

Имена данных иерархичны по своей природе, причем, содержит (может содержать) глобальное имя производителя контента -> приложение, в котором данные будут обработаны -> экземпляр приложения (например, вкладка в браузере). Кроме того, в имени данных могут быть и другие нужные данные - версии, назначение, сегменты etc.



И самое главное - пакеты с данными теперь могут храниться на коммуникационных узлах. То есть, по сути - сеть NDN это глобальный кэш данных, хранилище, и даже в некотором смысле - обработчик данных. Обработчик, потому что никто не запрещает производить манипуляции с данными прямо на узлах NDN и исполнять, например, программный код. Получается глобальный распределенный сервер, который действительно трудно "уронить".

Всевозможные аппликации и реалтайм-общение, разумеется, тоже предусматриваются.



И самое главное - серьезно экономятся коммуникационные ресурсы сети. Теперь нет необходимости отправлять поток "один-к-одному" в случае многократных однотипных запросов - все хранится "в сети", что означает, что единожды появившийся и запрошенный контент становится (со временем) достоянием всей сети.

Но, хочу это подчеркнуть, NDN - это НЕ P2P сеть, как мне пару раз встретилось в русскоязычной прессе. Даже близко нет тут никакого "пиринга". На самом деле, ближе всего это глобальный кеш Google, но сильно умнее и универсальнее.

Вот более подробная "архитектурная схема" NDN с точки зрения различных приложений:

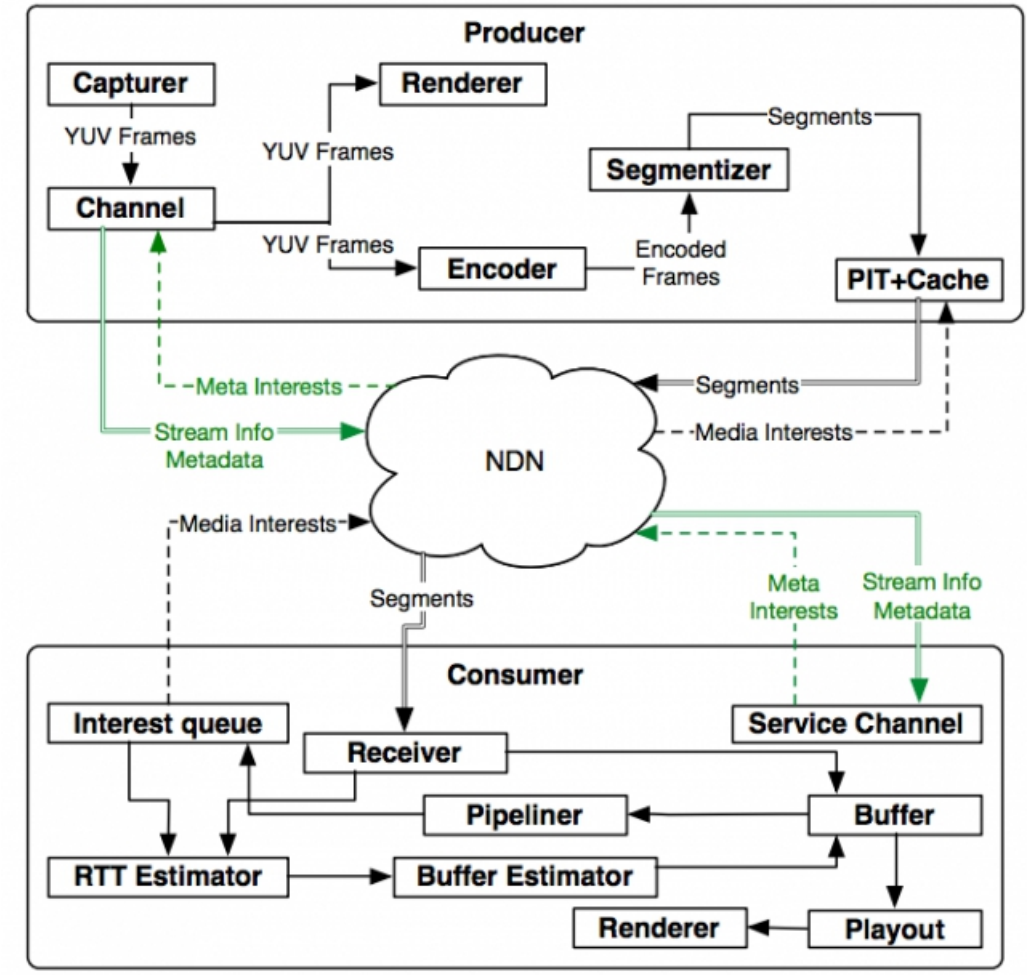
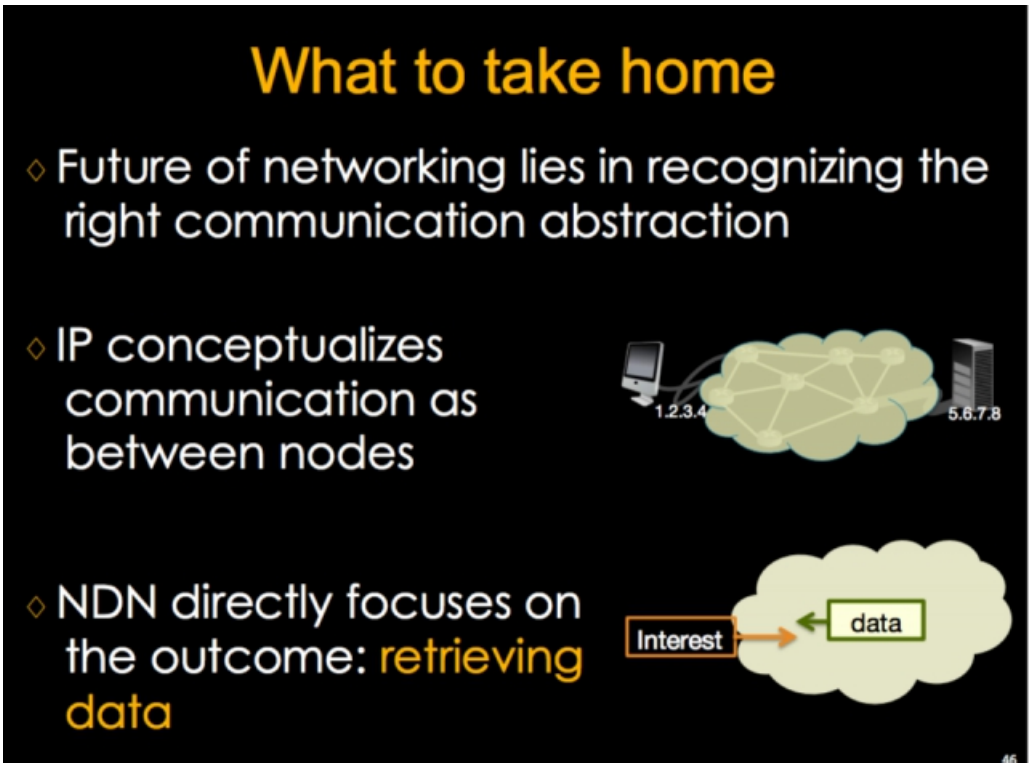


Figure 3.5: ndnrtc Application Architecture.

И еще раз, глобальное отличие NDN-архитектуры, от TCP/IP - пользователь "просто запрашивает данные".



Еще раз хочу оговорить, что данная архитектура существует пока только в качестве экспериментальной проработки. Пока не существует реальных внедрений, а математика процессов находится в глубокой проработке. Не решены очень много задач и проблем. Но наука не стоит на месте. Например, прямо сейчас UCLA, Washington University, CERN и Shanghai University проводят натурные эксперименты по развитию вот такой сети:



В основе разработки, надо отметить, лежит не менее передовая концепция Software Delivery Network, которую довольно активно обсуждали на Наге. Использование SDN делает новые эксперименты с протоколами менее затратными и более гибкими - не нужно менять полностью оборудование, а просто меняются прошивки, а сами сетевые устройства по воле администрации сети меняют свои ключевые свойства по всему миру.

Ну, и в завершение, хочу отметить один факт - обратите внимание, что в проекте не принимает участие ни ведущий российский вуз МГУ, ни "центр инновации" Сколково. Хотя в референсах на сайте NDN я часто встречаю русские фамилии, как, впрочем, и китайские, и французские, и греческие. Но можно назвать одного из ведущих инженеров проекта NDN - Александра Афанасьева, в прошлом сотрудника московского RTI Systems и Videofon MV, а ныне ведущего рисёрчера UCLA по нашей теме.

Факт, на самом деле, весьма печальный. Ничего не хочу сказать плохого, ни про МГУ, ни про Сколково, ни про родной УрФУ. Просто сам факт, что российские вузы и исследовательские институты вообще никак не участвуют в разработке прорывных технологий подобного класса, удручает. На мой взгляд, эта тема должна быть в первых строках запросов грантов, если Россия действительно желает быть самостоятельной ИТ- и телеком- державой. Нужно заниматься реальными научными исследованиями, а не догонять ушедший поезд в поисках "импортозамещения" устаревающих "отечественных операционных систем", "офисных пакетов" и СУБД. Хотя бы потому, что когда наступит эра NDN все эти субъекты внимания властей устареют и станут анахронизмом по примеру подкованной Левшой блохи, которая гвоздями подкована, да, вот беда - утратила

способности танцевать.

Не хочется мне переписывать Лескова сызнова...