Настройка резервирования канала связи на D-Link DIR-320

Дано: один компьютер (терминал по продаже билетов), корпоративная сеть с доступом в интернет, прямые руки (подразумеваются).
Необходимо: организовать резервный канал с автопереключением для этого компьютера.
В этой публикации я опишу своё **решение** на приснопамятном DLink DIR-320. Я потратил много времени на тщетные поиски: сначала способов и реализаций, потом прошивок и скриптов, прежде чем пришёл к **ответу**. Надеюсь, эта информация окажется полезной.

Решение

В силу некоторых обстоятельств связь в этой точке оставляет желать стабильности. После нескольких прецедентов на столе оказалась разнарядка. Вполне рядовая задача, казалось бы, но я с ней ещё не сталкивался в таком виде. Мыслительный и поисковой процессы произвели на свет следующие варианты:

1. Альтернативного провайдера завести на наш маршрутизатор (cisco 2801) и посредством ip sla организовать переключение.
2. Параллельно с нашим проводным подключением непосредственно на терминал с билетами завести альтернативного провайдера и с помощью прокси-сервера или скриптов организовать переключение.
3. Собственно установить DLink перед терминалом и на нём организовать переключение.

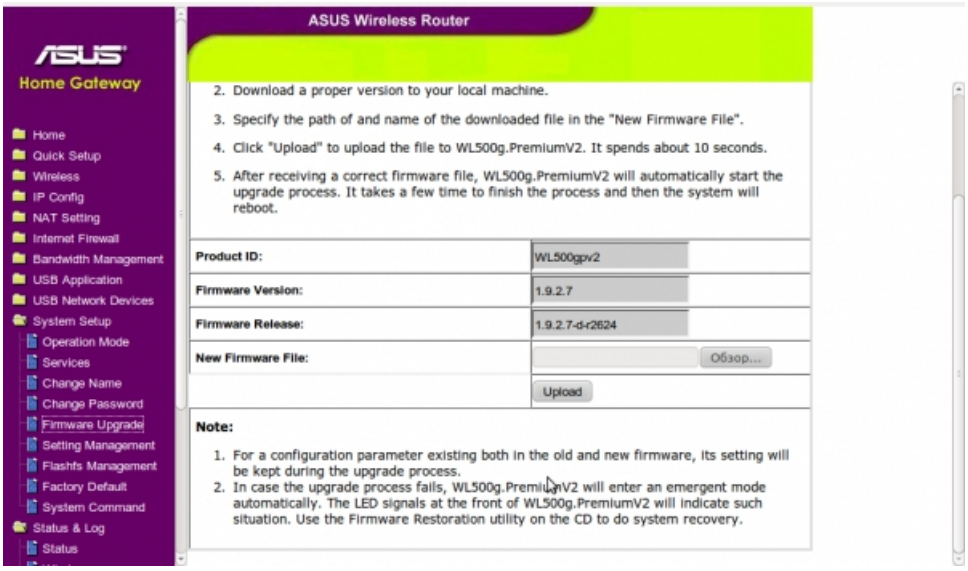
Первый, наиболее предпочитаемый вариант, пришлось отбросить сразу, поскольку указанная циска не поддерживает ip sla (или я плохо искал). Тут вообще интересно то, что из целого парка маршрутизаторов только один 3550 поддерживает его в той мере, в которой нужно.

Вариант с ПО на терминале кажется ненадёжным и плюс к тому водружать ради этих целей какой-нибудь UserGate не очень хотелось. Скрипты на конечной машине, даже если они в планировщике, выглядят, скорее, как костыли. В общем, в конечном итоге, я сделал выбор в пользу «железного» решения: Dlink — тем более что я слышал о существовании альтернативных прошивок с поддержкой резервирования.



Но на деле, как это водится, всё оказалось не так-то просто. Первый и последний, казалось бы, шаг: найти прошивку, поддерживающую 3G-модемы и резервирование. Последняя фирменная вполне себе соединяется с сетью 3G, но, к сожалению, никак нельзя настроить одновременную работу двух каналов и у него нет CLI. Довольно быстро удалось отыскать прошивку с требуемыми параметрами. Ей оказалась некое полуофициальное ПО с поддержкой WiMAX и 3G-модемов. Вполне приятный интерфейс, удобный и логичный. В нём даже был самый настоящий пункт «Резервирование», в котором можно настроить два адреса для проверки доступности, основной канал и резервный. Разумеется, это не работало, иначе не было бы и этой статьи. Переключение происходило лишь в одном направлении: с модема на провод, причём основным считался именно модем. Никакими усилиями исправить ситуацию не удалось. Доступ к командной строке устройства был закрыт, так что свобода полета мысли ограничивалась строгими рамками веб-интерфейса. Поиски продолжились. Часть прошивок не ставилась, другая не имела телнета, третья не понимала подключение USB-устройств.

И вот, наконец, я нашёл её: некрасивую, но свободную — прошивку для Asus'овской модели. Она легко и надолго засела в память. Веб-лицом не особо удалась, но нам с него и мёд не пить. Командная строка во всей красе, USB-устройства вполне определялись, поддержка 3G-модемов из коробки подразумевалась, но выполнить подключение к сети из веб-интерфейса всё-таки не получилось, логи PPP молчат.



Что ж, остался лишь путь настоящего ИТ-шника: консоль. Благо прошивка основана на линуксе. Задача стоит следующая:

1. Держать активными одновременно два подключения: кабельное и 3G;
2. Проверять периодически доступность определённого хоста;
3. В случае его недоступности менять маршрут по умолчанию через интерфейс кабельный, на интерфейс PPP (3G);
4. В случае восстановления канала менять маршрут обратно.



Поскольку настройки модема не работают из веб-интерфейса, его подключение необходимо выполнить скриптом.

Детали и препоны

В ходе работы стало ясно, что без подводных камней не обойтись.

1. В таблице маршрутизации есть одновременно два маршрута по умолчанию с одинаковой метрикой: через интерфейс *ppp0* (3G) и через *vlan1* (WAN). Приоритетным считается проводное соединение. Поскольку просто поменять метрику маршрута в линуксе нельзя, маршрут через интерфейс *vlan1*, в случае чего, будем удалять или добавлять.

2. Тесты показали, что недостаточно просто поменять маршрут по умолчанию. С самого DLinka связь есть, а с конечной машины нет. После недолгих умственных потуг становится ясно, что не отрабатывает NAT. Правила NAT остаются от предыдущего подключения и пакеты от компьютера теряются на DLink'e. В данному случае я, возможно, сделал нерационально, но одновременно с удалением/добавлением маршрута применяется новое правило NAT.

3. Очень неприятный необъяснимый момент: в случае недоступности хоста по проводному соединению, через 15-20 секунд отваливается и 3G. Причём отваливается так, что необходима физическая перезагрузка роутера. Но если успеть перевести маршрут на модем за это время, то всё работает нормально. Crontab не позволяет запускать скрипты чаще одного раза в минуту. Можно

сделать скрипт, который 10 раз, например, в цикле отработает, и раз в минуту запускать его по крону. Но после серии тестов оказалось, что как-то это криво работает, словно не каждые 6 секунд, как должно. Поэтому пришлось остановиться на варианте бесконечного цикла с задержкой, который будет запускаться при старте устройства. Задержку выбрал в 2 секунды.

4.Для настройки модема необходимы параметры *Vendor ID* и *Product ID*. В линуксе их легко узнать, используя команду **lsusb**. Она вернёт список подключённых по USB-интерфейсу устройств, в котором в формате *Vendor ID:Product ID* (две 4-х значные величины) будут необходимые параметры. Два замечания: в DLink'е такой команды нет и к указанным строкам нужно добавить ещё *0x* вначале.

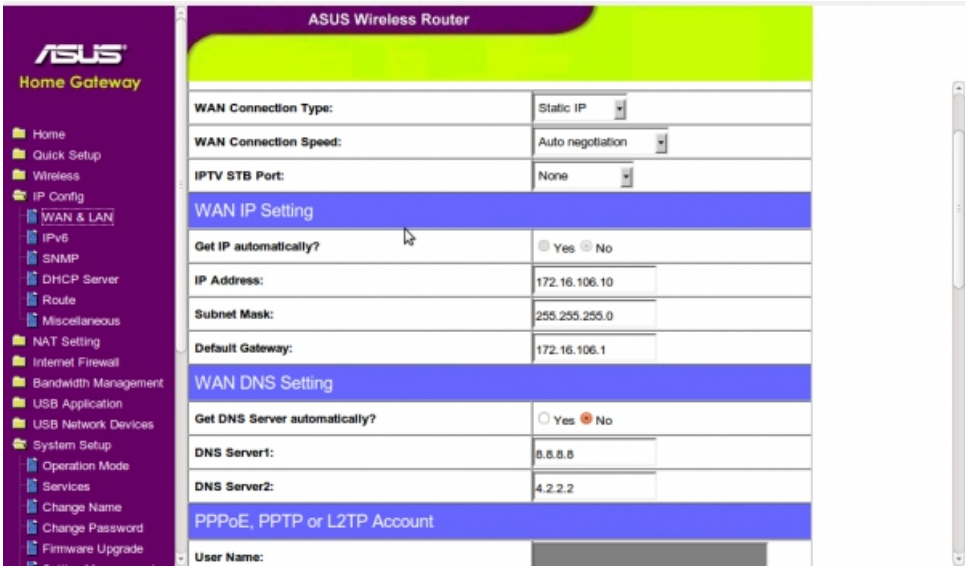
5. Задать хост в интернете для проверки доступа проблематично. Он будет доступен через модем и после восстановления проводного канала, условие не сработает. Я вышел не самым красивым способом, но вполне действенным: я пингую хост своей локальной сети, для этого достаточно в автозагрузке прописать добавление статического маршрута до этой сети через интерфейс WAN. Впрочем, никто не запрещает задать маршрут до адреса x.y.z.w 255.255.255.255 в интернете через этот интерфейс — результат будет тем же.

6. После всех изменений в командной строке, нужно сохранить их следующими командами:

nvram commit
flashfs enable
flashfs save
flashfs commit

Настройка

Минимум настройки через веб-интерфейс: прописать IP адрес WAN:



Впрочем, даже это вполне можно сделать в командной строке (**ifconfig, route**).

Скрипты

Признаться честно, часть скрипта по подключению модема и пост-загрузке, досталась мне по наследству, поэтому автор строк мне не известен, но спасибо большое этому человеку. Все необходимые скрипты с комментариями приложены к статье:

1. *post-boot* — основной скрипт, выполняющийся после старта устройства. Он запускает скрипт управления светодиодами, добавляет нужный нам маршрут, запускает скрипты *modem.sh* и *pingtest.sh*.
2. *ledctl* — скрипт управления светодиодами.
3. *modem.sh* — скрипт выполнения подключения через модем.
4. *pingtest.sh* — скрипт для проверки доступности узла и переключения маршрутов.
5. *Dlink.sh* — сводный файл: скопировал из редактора, вставил в консоль и выполнил.

В оригинале я исправлял лишь необходимые параметры и закомментировал то, что не нужно.

Схема работы следующая:

После старта устройства запускается скрипт *post-boot*, который задаёт нужные параметры, в том числе, добавляет необходимый нам маршрут в локальную сеть.

В этом скрипте запускается *modem.sh* и *pingtest.sh*. *Modem.sh* отрабатывает лишь один раз до тех пор, пока не произойдёт подключение к сети.
Pingtest.sh крутится в памяти роутера постоянно и каждые две секунды проверяет доступность хоста.

Для переноса этих скриптов есть три варианта:

а) скопировать их на флэшку, примонтировать её к фс роутера и перенести в нужный каталог;
б) создать их через встроенный редактор vi (врагу не пожелаю);
в) вставить строки по порядку в файл командой *echo* (этот способ реализован в файле *Dlink.sh*).

Скачать скрипты.

Приложения

Интерфейсы:

br0 — LAN
vlan1 — WAN
ppp0 — интерфейс через 3G-модем

На этой фотографии видно по результатам пинга, как меняется время ответа при переключении канала:



Кстати, при переключении с беспроводного канала на проводной, пинг останавливается до тех пор, пока его не перезапустишь. То есть при сёрфинге то переключение будет совершенно незаметно, а вот скачивание может прерваться.

Таблица маршрутизации:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.64.64.64	*	255.255.255.255	UH	0	0	0	ppp0
10.64.64.64	*	255.255.255.255	UH	0	0	0	ppp0
172.16.106.1	*	255.255.255.255	UH	0	0	0	vlan1
192.168.1.0	*	255.255.255.0	U	0	0	0	br0
172.16.106.0	*	255.255.255.0	U	0	0	0	vlan1
172.16.0.0	172.16.106.1	255.255.0.0	UG	0	0	0	vlan1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	10.64.64.64	0.0.0.0	UG	0	0	0	ppp0
default	172.16.106.1	0.0.0.0	UG	0	0	0	vlan1

Логи (**tail -f /usr/tmp/syslog**) успешного подключения к 3G-сети:

Running modem.sh

Aug 17 05:51:37 kernel: usb.c: registered new driver serial

Aug 17 05:51:37 kernel: usbserial.c: USB Serial support registered for Generic
Aug 17 05:51:37 kernel: usbserial.c: Generic converter detected
Aug 17 05:51:37 kernel: usbserial.c: Generic converter now attached to ttyUSB0 (or usb/tts/0 for devfs)
Aug 17 05:51:37 kernel: usbserial.c: Generic converter detected
Aug 17 05:51:37 kernel: usbserial.c: Generic converter now attached to ttyUSB1 (or usb/tts/1 for devfs)
Aug 17 05:51:37 kernel: usbserial.c: Generic converter detected
Aug 17 05:51:37 kernel: usbserial.c: Generic converter now attached to ttyUSB2 (or usb/tts/2 for devfs)
Aug 17 05:51:37 kernel: usbserial.c: USB Serial Driver core v1.4
Aug 17 05:51:52 pppd[508]: pppd 2.4.5 started by admin, uid 0
Aug 17 05:51:55 pppd[508]: Serial connection established.
Aug 17 05:51:55 pppd[508]: Using interface ppp0
Aug 17 05:51:55 pppd[508]: Connect: ppp0 <--> /dev/usb/tts/2
Aug 17 05:51:58 pppd[508]: Could not determine remote IP address: defaulting to 10.64.64.64
Aug 17 05:51:58 pppd[508]: local IP address 10.117.240.93
Aug 17 05:51:58 pppd[508]: remote IP address 10.64.64.64
Aug 17 05:51:58 pppd[508]: primary DNS address 85.26.231.218
Aug 17 05:51:58 pppd[508]: secondary DNS address 83.149.49.70
Aug 17 05:51:58 Static: connected to ISP