Система бесперебойных соединений *ui*TCP Быстрый старт (ПО NSG Linux 1.0)

Система бесперебойных соединений *ui*TCP — фирменная технология VPN 4 уровня, разработанная компанией NSG для банковских и других специфических приложений. Главное отличие от других аналогичных решение (STunnel, OpenVPN, KerioNet и др.) состоит в механизме гарантированной доставки пакетов при множественных обрывах связи и переходах на резервные каналы связи. Система реализована в виде дополнительного модуля (*plug-in*) поверх основного программного обеспечения NSG Linux 1.0.

Данный документ кратко описывает основные процедуры для построения тестовой инсталляции *ui*TCP из сервера и одного клиента (по состоянию на версию *ui*TCP 0.23 и NSG Linux 1.0 build 4 rc1). Используется наиболее частое применение — передача трафика TCP. Подробно процедуры настройки основного программного обеспечения NSG Linux 1.0 и *ui*TCP рассмотрены в документах NSG:

> *Мультипротокольные маршрутизаторы и коммутаторы пакетов NSG. Программное обеспечение NSG Linux. Руководство пользователя.*
>
> *Мультипротокольные маршрутизаторы и коммутаторы пакетов NSG. Программное обеспечение NSG Linux. ui TCP. Система обеспечения бесперебойных соединений.*

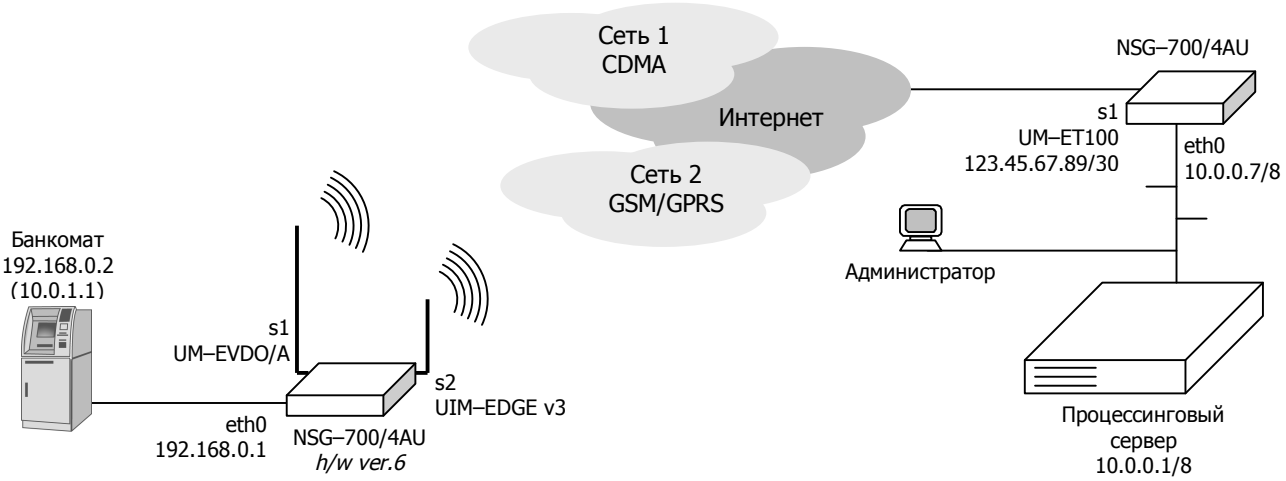
В качестве клиентского устройства используется NSG–700/4AU. В качестве сервера для реального построения крупномасштабных систем (десятки, сотни клиентов) следует использовать специализированные коммуникационные серверы NSG–1000/GW. Однако для целей тестирования и оценки возможностей данной технологии, а также для небольших систем (до 30 клиентов) в качестве сервера может использоваться второе устройство NSG–700/4AU.

ПРИМЕЧАНИЕ Данный документ описывает настройки *ui*TCP применительно к основному программному обеспечению NSG Linux 1.0.
В программном обеспечении NSG Linux 2.0 система *ui*TCP является штатной компонентой и настраивается аналогичным образом, заодно с остальными компонентами программного обеспечения, с помощью Web-интерфейса или консольной утилиты nsgsh.

1. Типовые конфигурации

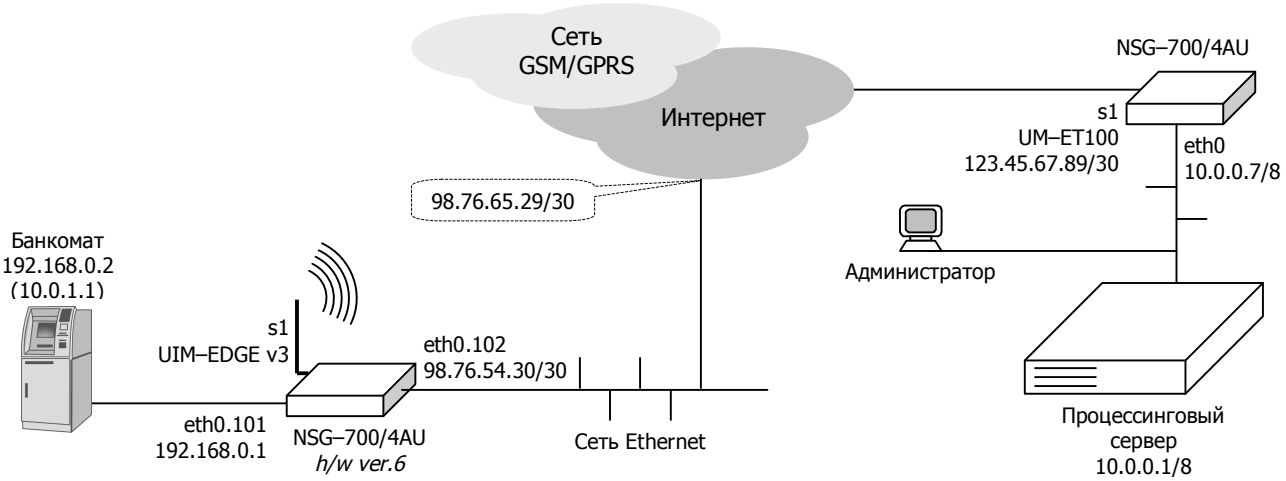
Будем предполагать, что поставлена задача подключить один банкомат к процессинговому серверу, используя двух операторов. Оба оператора предоставляют только базовую услугу — доступ в Интернет с динамическими приватными IP-адресами, наиболее сложную для целей построения корпоративных сетей.

Конфигурация а). Основной является сеть CDMA (Скайлинк), резервной — GRPS (для определённости, Мегафон). Соответственно, в качестве клиента используется шасси NSG–700/4AU *h/w ver.6* с модулями UM–EVDO/A *h/w ver.5* (разъём s1) и UIM–EDGE *h/w ver.3* (разъём s2).



Конфигурация б). Основным оператором является МТС, резервным — Билайн. Используется шасси NSG–700/4AU (аппаратная версия несущественна) с 2-симчатым модулем UIM–EDGE *h/w ver.3* (разъём s1).

Конфигурация в). Основное соединение по наземному каналу Ethernet, резервное — Мегафон. Используется шасси NSG–700/4AU (версия несущественна) с модулем UIM–EDGE *h/w ver.3* (разъём s1).



В качестве сервера используется устройство NSG–700/4AU (аппаратная версия несущественна). Чтобы удовлетворить самым строгим требованиям безопасности, в шасси установлен дополнительный физический порт Ethernet — модуль UM–ET100. Встроенным портом eth0 устройство включено в локальную сеть процессингового центра.

Банкомат устанавливает TCP-соединение к процессинговому серверу в локальной сети банка. На процессинге он зарегистрирован под IP-адресом 10.0.1.1. Кроме того, предполагается (хотя это не обязательно), что каждый банкомат устанавливает соединение на уникальный TCP порт сервера, для данного банкомата — 20001.

Помимо этого, в сети банка имеется администратор(ы), которому требуется устанавливать соединения со своей стороны к банкомату и к клиентскому устройству NSG.

2а. Настройка основного ПО (вариант CDMA + GPRS)

Конфигурация сервера:

```
hostname uitcp_srv
!
nsg
    users
        user-name nsg md5 12345
        user-name root md5 12345
        exit
    card s1 um-et100
    port eth0
        ip address 10.0.0.7/8
        exit
    port s1
        ip address 123.45.67.89/30
        exit
!
ip route 0.0.0.0/0 123.45.67.90
!
```

Основная конфигурация клиентского устройства. SIM-карта Мегафон установлена в основное (верхнее) гнездо на модуле IM–EDGE, PIN-код с обеих карт снят, все перемычки (аппаратного рестарта и 1-/2-симчатого режима) на обоих модулях установлены.

```
hostname ATM001
!
nsg
    users
        user-name nsg md5 qwerty
        user-name root md5 qwerty
        user-name mobile open internet
        user-name gdata open gdata
        exit
    virtual-template 1
        keepalive 10 retry 3
        ppp ipcp accept-address yes
        ppp sent-username mobile
        exit
    virtual-template 2
        keepalive 10 retry 3
        ppp ipcp accept-address yes
        ppp sent-username gdata
        exit
    chat-script CDMA "TIMEOUT 40 XXX-\rAT-OK ATD#777 CONNECT ''"
    chat-script GPRS "TIMEOUT 30 XXX-\rAT-OK AT+CGDCONT=1,\"IP\",\"internet\" OK ATD*99***1# CONNECT ''"
    card s1 uim-cdma
    card s2 uim-edge
    port eth0
        ip address 192.168.0.1/24
        exit
    port s1
        encapsulation ppp
        virtual-template 1
        chat-script CDMA
        exit
    port s2
        encapsulation ppp
        virtual-template 2
        chat-script GPRS
        exit
!
```

Проверка доступности сервера через сотовые сети и Интернет. Для работы *ui*TCP эти маршруты не нужны, поэтому после проверки они удаляются (полужирным шрифтом выделен ввод пользователя):

```
ATM001(config)# ip route 123.45.67.89/32 s1
^Z
ATM001# ping 123.45.67.89
.....
^C
ATM001# configure terminal
ATM001(config)# no ip route 123.45.67.89/32 s1
ATM001(config)# ip route 123.45.67.89/32 s2
^Z
ATM001# ping 123.45.67.89
.....
^C
ATM001# configure terminal
ATM001(config)# no ip route 123.45.67.89/32 s2
```

Включение SSH для управления устройствами (на обоих устройствах). Помимо прямого назначения, SSH можно использовать как удобный инструмент для передачи файлов с одного устройства на другое; для этой же цели выше были установлены пароли для пользователя root.

```
ATM001(config-nsg)# services ssh keygen
This command may require a lot of time
Repeat the previous command and wait.
ATM001(config-nsg)# services ssh keygen
$Generating SSH1 RSA host key:success
$Generating SSH2 RSA host key:success
$Generating SSH2 DSA host key: success
ATM001(config-nsg)# services ssh enable
```

Настройка сервера NTP (на обоих устройствах). Корректная установка системного времени критически важна для работы *ui*TCP, поскольку сертификаты X.509 имеют ограниченный срок действия (как сверху, так и снизу). Рекомендуется синхронизировать все устройства в системе от одного сервера. В данном примере используется ntp.psn.ru [194.149.67.130].

```
!
nsg
    ntp client
        host 194.149.67.130
        adm-state up
        exit
!
ip route 194.149.67.130/32 s1              (на клиенте)
ip route 194.149.67.130/32 s2 2            (на клиенте)
    или
ip route 194.149.67.130/32 123.45.67.90    (на сервере)
!
```

Настройка сервера DHCP на клиентском устройстве. Он позволяет заранее прописать сетевые настройки банкомата в типовой конфигурации устройства NSG. Для удобства массовой инсталляции все удалённые площадки имеют одинаковую конфигурацию: устройство NSG — 192.168.0.1, банкомат — 192.168.0.2.

```
!
nsg
    dhcp 1
        ip-address 192.168.0.2 through 192.168.0.2
        mask-length 24
        interface eth0
        exit
!
```

На банкомате, по умолчанию, включена динамическая конфигурация параметров TCP/IP; таким образом, их можно не настраивать вообще. Кроме того, как будет показано ниже, банкомат будет всегда обращаться к процессинговому серверу по адресу 192.168.0.1 (т.е. к своему устройству NSG, работающему как TCP proxy) и по порту TCP 20000. Подстановка уникального IP-адреса банкомата и/или номера порта TCP будет выполняться на сервере *ui*TCP.

Таким образом, в данной задаче уникальными параметрами на всей площадке оказываются, как будет видно из дальнейшего, только имя клиентского устройства NSG в системе *ui*TCP, его ключ и сертификат X.509. Все остальные параметры одинаковы для всех клиентов и могут быть загружены в них в виде готовой типовой конфигурации.

2б. Настройка основного ПО (вариант 2×GPRS)

Конфигурация сервера идентична предыдущему случаю, поскольку она не зависит от способа подключения клиентов. Это же относится к настройке SSH, клиента NTP (кроме маршрутов на NTP сервер) и сервера DHCP на клиентском устройстве. Основная конфигурация клиентского устройства приведена ниже. SIM-карта МТС установлена в верхнее гнездо, Билайн — в нижнее. Перемычка аппаратного рестарта установлена, перемычка одно-/двухсимчатого режима снята.

```
!
hostname ATM1234
nsg
    users
        user-name nsg md5 qwerty
        user-name root md5 qwerty
        user-name mts open mts
        user-name beeline open beeline
        exit
    virtual-template 1
        keepalive 10 retry 3
        ppp ipcp accept-address yes
        ppp sent-username mts
        exit
    virtual-template 2
        keepalive 10 retry 3
        ppp ipcp accept-address yes
        ppp sent-username beeline
        exit
    chat-script MTS "TIMEOUT 20 XXX-\rAT-OK AT+CGDCONT=1,\"IP\",\"internet.mts.ru\" OK ATD*99***1#  ↵
CONNECT ''"
    chat-script BEELINE "TIMEOUT 20 XXX-\rAT-OK AT+CGDCONT=1,\"IP\",\"internet.beeline.ru\" OK  ↵
ATD*99***1# CONNECT ''"
    card s1 uim-edge
    port s1
        encapsulation ppp
        virtual-template 1 aux 2
        chat-script MTS aux BEELINE
        prio main 1 aux 1
        exit
    port eth0
        ip address 192.168.0.1/24
        exit
!
```

Проверка доступности сервера через сотовые сети и Интернет. Чтобы проверить работу через разных операторов, следует войти в командную оболочку ОС Linux и там принудительно завершить процесс pppd; в соответствии с настройками порта, он рестартует уже с другой SIM-картой.

```
ATM001(config)# ip route 123.45.67.89/32 s1
^Z
ATM001# start-shell
root@ATM001 #  ping 123.45.67.89
.....
^C
root@ATM001 # killall pppd
    (подождать рестарта модуля 20-25 сек)
root@ATM001 #  ping 123.45.67.89
.....
^C
root@ATM001 # exit
ATM001# configure terminal
ATM001(config)# no ip route 123.45.67.89/32 s1
```

Чтобы непосредственно убедиться, что соединение имеет место с другим оператором, можно проверить состояние интерфейса командой ifconfig s1 — в сетях разных операторов адреса будут выделяться, как правило, из разных диапазонов.

2в. Настройка основного ПО (вариант Ethernet+GPRS)

Конфигурация сервера идентична предыдущим двум случаям, поскольку она не зависит от способа подключения клиентов. Это же относится к настройке SSH, клиента NTP и сервера DHCP на клиентском устройстве. Основная конфигурация клиентского устройства приведена ниже. SIM-карта Мегафон установлена в верхнее гнездо, все перемычки на модуле установлены.

```
hostname ATM001
!
nsg
    users
        user-name nsg md5 qwerty
        user-name root md5 qwerty
        user-name gdata open gdata
        exit
    virtual-template 2
        keepalive 30 retry 3
        ppp ipcp accept-address yes
        ppp sent-username gdata
        exit
    chat-script GPRS "TIMEOUT 30 XXX-\rAT-OK AT+CGDCONT=1,\"IP\",\"internet\" OK ATD*99***1# CONNECT ''"
    ethernet-switch
        mode vlan
        phy 0 vlan vlan-groups "101-103" deny-other
        phy 1 norm vlan-group 101
        phy 2 norm vlan-group 102
        phy 3 norm vlan-group 103
        phy 4 norm vlan-group no
        exit
    card s1 uim-edge
    port eth0
        vlan 101
            ip address 192.168.0.1/24
            exit
        vlan 102
            ip address 98.76.54.30/30
            exit
        exit
    port s1
        encapsulation ppp
        virtual-template 2
        chat-script GPRS
        exit
!
```

Проверка доступности сервера:

```
ATM001(config)# ip route 123.45.67.89/32  98.76.54.29
^Z
ATM001# ping 123.45.67.89
.....
^C
ATM001# configure terminal
ATM001(config)# no ip route 123.45.67.89/32  98.76.54.29
ATM001(config)# ip route 123.45.67.89/32 s1
^Z
ATM001# ping 123.45.67.89
.....
^C
ATM001# configure terminal
ATM001(config)# no ip route 123.45.67.89/32 s1
```

3. Установка *ui*TCP

Для установки *ui*TCP необходимо загрузить с FTP-сервера NSG файл вида uitcp-X.Y (где X.Y — номер версии). Рекомендуется поместить его на ресурсе, доступном локально (хотя можно загружать и непосредственно с сайта утилитой ftpget).

Установка и настройка *ui*TCP выполняются в командной оболочке NSG Linux. В неё можно перейти с помощью команды

```
nsg# start-shell
```

либо войти в систему под именем root. Все изменения необходимо сохранять командой savecfg.

Файл необходимо загрузить на устройство любым удобным способом (рекомендуется во временную директорию), назначить ему права на исполнение и исполнить:

```
root@uitcp_srv # cd /tmp
root@uitcp_srv tmp # tftp -g Z.Z.Z.Z -r uitcp-X.Y
root@uitcp_srv tmp # chmod +x uitcp-X.Y
root@uitcp_srv tmp # ./uitcp-X.Y
```

В результате на устройстве создаются директории и файлы, необходимые для работы *ui*TCP и Web-сервера:

```
root@uitcp_srv tmp # ls /etc/uitcp
bin
root@uitcp_srv tmp# cd /etc/uitcp/bin
root@uitcp_srv bin# ls
certs.conf      tools           uitcp           uitcp_web
.....
```

Установка *ui*TCP на клиенте выполняется, до этого момента, таким же образом. Для доступа к TFTP серверу по локальной сети можно временно назначить порту Ethernet любой IP-адрес, приемлемый для локальной сети:

```
root@ATM001 ifconfig eth0 10.0.1.123/8
```

Адрес, заданный таким образом, действует до рестарта системы и не сохраняется в конфигурации.

Для дальнейшей настройки будет использоваться утилита uish (*ui*TCP shell). Можно запустить её сейчас для проверки, заодно создать файл конфигурации *ui*TCP и настроить один критически важный параметр — режим клиент/сервер (в противном случае, для изменения этого параметра впоследствии придётся рестартовать *ui*TCP):

```
root@uitcp_srv bin # ./tools/uish
Error loading config file /etc/uitcp/config: cannot open /etc/uitcp/config: No such file or directory
Can't find config. Create new config.
Are you sure?[yes/no] yes
NSG Device Configurator,  Copyright (C) 2008-2009, nsg.ru
Scheme file - ./tools/../uitcpScheme.lua
Config file - /etc/uitcp/config
(type _help for usage information)
uitcp>mode
mode = client   или   server   соответственно
= "client"
uitcp>_show
{
    mode = "client"
}
uitcp>_write
Configuration will be saved to file /etc/uitcp/config
Are you sure?[yes/no] yes
.....
(II) savecfg: Configuration saved successfully. Exit.
Configuration saved
uitcp>_quit
```

Чтобы закончить установку, нужно создать (и на клиенте, и на сервере) запись в файле /etc/inittab для автоматического запуска *ui*TCP, сохранить изменения и рестартовать систему:

```
root@uitcp_srv echo "#Autostarting uitcp" >> /etc/inittab
root@uitcp_srv echo null::respawn:/etc/uitcp/bin/uitcp >> /etc/inittab
root@uitcp_srv savecfg
root@uitcp_srv reboot
```

Вместо echo можно внести изменения в файл вручную с помощью редактора nano.

После рестарта следует выполнить команду ps и убедиться, что uitcp присутствует в списке процессов.

4. Настройка Web-сервера

Для автоматического запуска Web-сервера необходимо добавить в файл /etc/inittab строку:

```
null::respawn:/usr/sbin/httpd -c /etc/uitcp/bin/uitcp_web/http.uitcp.conf -h /etc/uitcp/bin/uitcp_web
```

(с помощью команды echo или редактора nano). После рестарта следует выполнить команду ps и убедиться, что httpd присутствует в списке процессов. Теперь для мониторинга и управления системой можно использовать Web-интерфейс.

Web-сервер нужен, в основном, на сервере, чтобы с его помощью наблюдать работу клиентов. Настройка и клиента, и сервера может производиться с помощью утилиты uish. При необходимости можно разово запустить Web-сервер на исполнение, чтобы, например, единожды выполнить конфигурацию клиентского устройства:

```
root@ATM001 # /usr/sbin/httpd -c /etc/uitcp/bin/uitcp_web/http.uitcp.conf -h /etc/uitcp/bin/uitcp_web
```

5. Генерация сертификатов X.509

Поскольку в практических применениях *ui*TCP необходима, как минимум, надёжная аутентификация клиентов, целесообразно включить в данную минимальную инсталляцию настройку SSL. Это, в свою очередь, требует установки взаимосвязанных сертификатов и ключей X.509 на серверные и на клиентские устройства. Сертификаты могут быть выданы сторонним удостоверяющим центром и помещены в соответствующие директории в виде файлов формата *.pem . Однако в данной тестовой инсталляции будет продемонстрировано создание самоподписанных сертификатов при помощи встроенных инструментов *ui*TCP.

На сервере необходимо создать директории, используемые *ui*TCP по умолчанию, и скопировать образцы конфигурации сертификатов:

```
root@uitcp_srv # mkdir /etc/uitcp/certs
root@uitcp_srv # mkdir /etc/uitcp/certs/capath
root@uitcp_srv # cp -r /etc/uitcp/bin/certs.conf /etc/uitcp/
```

Образцы конфигураций, скопированные в директорию /etc/uitcp/certs.conf, следует отредактировать должным образом при помощи редактора nano. Как минимум, необходимо в файлах root.cnf и server.cnf в секции [req_distinguished_name] в поле commonName вписать реальный IP-адрес (123.45.67.89) или доменное имя сервера.

ВНИМАНИЕ Перед созданием сертификатов необходимо убедиться, что системное время на сервере установлено правильно.

После этого можно приступать непосредственно к созданию сертификатов:

```
root@uitcp_srv # cd /etc/uitcp/bin/tools/
root@uitcp_srv tools # ./cert-root
root@uitcp_srv tools # ./cert-server
root@uitcp_srv tools # ./cert-client ATM001
root@uitcp_srv tools # savecfg
```

В результате на сервере создаются необходимые сертификаты и ключи: корневой, самого сервера, и клиента под именем ATM001. Для последующих клиентов необходимо повторять только последний скрипт с новым именем клиента. В заключение необходимо составить локальный список клиентских сертификатов с хэшами в качестве имён, для проверки их действительности:

```
root@uitcp_srv tools # rehash
```

Корневой сертификат и сертификаты сервера готовы к использованию. Набор сертификатов и ключей, необходимых клиенту, упакован в один архив /etc/uitcp/certs/*имя_клиента*/cert-client.*имя_клиента*.tgz . Его необходимо переместить на клиентское устройство любым безопасным способом (на USB Flash, через промежуточный FTP или TFTP сервер в локальной сети и т.п.) и распаковать в нужную директорию. В нижеприведённом примере файл копируется на клиента по SSH (предполагается, что клиент временно подключён к локальной сети банка):

```
root@ATM001 # mkdir /etc/uitcp/certs
root@ATM001 # cd /etc/uitcp/certs
root@ATM001 certs # ifconfig eth0 10.0.1.123/8
root@ATM001 certs # ssh root@10.0.0.7 "cat /etc/uitcp/certs/ATM001/cert-client.ATM001.tgz" |
cat > cert-client.ATM001.tgz
root@ATM001 certs # tar -xzf ./cert-client.ATM001.tgz
root@ATM001 certs # rm cert-client.ATM001.tgz
root@ATM001 certs # savecfg
```

6а. Создание туннеля *ui*TCP (вариант CDMA+GPRS)

Теперь можно приступать непосредственно к настройке *ui*TCP. Первый этап — создание отказоустойчивого туннеля между клиентским и серверным устройствами. Настройку рекомендуется производить с помощью утилиты uish.

Настройка SSL на обоих устройствах. Устанавливаются только рекомендуемые опции для максимальной безопасности; все сертификаты уже находятся в файлах и директориях, используемых по умолчанию, и специальной настройки не требуют.

```
uitcp>ssl
uitcp.ssl>options
uitcp.ssl.options = {"all", "no_sslv2"}
= {"all","no_sslv2"}
uitcp.ssl>verify
uitcp.ssl.verify = {"peer","fail_if_no_peer_cert"}
= {"peer","fail_if_no_peer_cert"}
uitcp.ssl>_show
{
    options = {
        [1] = "all",
        [2] = "no_sslv2"
    },
    verify = {
        [1] = "peer",
        [2] = "fail_if_no_peer_cert"
    }
}
uitcp.ssl>(пробел)(enter)
uitcp>
```

Настройка журналов для отладки (на обоих устройствах):

```
uitcp>log
uitcp.log>level
uitcp.log.level = d
= "debug"
uitcp.log>logFileMaxSize
uitcp.log.logFileMaxSize = 10485760
= 10485760
uitcp.log>_s
{
    level = "debug",
    logFileMaxSize = 10485760
}
uitcp.log>(пробел)(enter)
uitcp>
```

Как видно, команды в uish (в том числе и имена объектов, созданных пользователем) можно сокращать. Далее на сервере нужно создать запись для клиента, хотя никаких настроек по существу пока делать не будем:

```
uitcp>clients
uitcp.clients>_n
uitcp.clients._.new = ATM001
uitcp.clients>A
uitcp.clients.ATM001>d
uitcp.clients.ATM001.description = test ATM
= "test ATM"
uitcp.clients.ATM001>(пробел)(пробел)(enter)
uitcp>_apply
Config to apply
{
    clients = {
        ATM001 = {
        description = "test ATM"
        }
    },
    mode = "server"
}
Are you sure?[yes/no] yes
uitcp>
```

На сервере uitcp ждёт входящих соединений, по умолчанию, на порту TCP 50005 (при необходимости можно настроить другой порт, и избирательно указать интерфейсы). Теперь можно выйти из uish и запустить просмотр системного журнала, чтобы следить за попытками клиента подключиться:

```
root@uitcp_srv # tail -f /var/log/uitcp
2010-02-15 00:15:30 INFO :============= uiTCP-0.23 beginning =============
2010-02-15 00:15:31 INFO :Reopen log file. Next output may be in other file
2010-02-15 00:15:31 INFO :============== uiTCP-0.23 started ==============
2010-02-15 00:15:34 INFO :Server mode
2010-02-15 00:15:34 INFO :Waiting connection from talker on 0.0.0.0:50005
2010-02-15 00:15:34 INFO :Update task is created: 2
2010-02-15 00:15:34 INFO :Waiting connection from talker on 0.0.0.0:50006
2010-02-15 00:15:34 INFO :Start main loop
```

(для завершения вывода нужно нажать CTRL-C).

Все операции по контролю работоспособности туннеля, переходу на резервный канал, возвращению на основной и т.п. в *ui*TCP возлагаются на клиента, поэтому основная часть настроек производится на нём:

```
uitcp>tunnel
uitcp.tunnel>name
uitcp.tunnel.description = ATM001
= "ATM001"
uitcp.tunnel>des
uitcp.tunnel.description = test ATM
= "test ATM"
uitcp.tunnel>servers
uitcp.tunnel.servers>_n
uitcp.tunnel.servers._.new>1
uitcp.tunnel.servers>1
uitcp.tunnel.servers.1 = 123.45.67.89:50005
= "123.45.67.89:50005"
uitcp.tunnel.servers.1>(пробел)(enter)
uitcp.tunnel.servers>(пробел)(enter)
uitcp.tunnel>keep
uitcp.tunnel.keepalive = 10
= 10
uitcp.tunnel>links
uitcp.tunnel.links>_i
uitcp.tunnel.links._.insert = 1
uitcp.tunnel.links>1
uitcp.tunnel.links.1>des
uitcp.tunnel.links.1.description = SKYLINK
= "SKYLINK"
uitcp.tunnel.links.1>dev
uitcp.tunnel.links.1.dev = s1
= "s1"
uitcp.tunnel.links.1>restart
uitcp.tunnel.links.1.restart>_i
uitcp.tunnel.links.1.restart._.insert = 1
uitcp.tunnel.links.1.restart>1
uitcp.tunnel.links.1.restart.1>sc
uitcp.tunnel.links.1.restart.1.script = config-nsg port s1 adm down; config-nsg port s1 adm up
= "config-nsg port s1 adm down; config-nsg port s1 adm up"
uitcp.tunnel.links.1.restart.1>(пробел)(enter)
uitcp.tunnel.links.1.restart>(пробел)(enter)
uitcp.tunnel.links.1>(пробел)(enter)
uitcp.tunnel.links>_i
uitcp.tunnel.links._.insert = 2
uitcp.tunnel.links>2
uitcp.tunnel.links.2>des
uitcp.tunnel.links.2.description = MEGAFON
= "MEGAFON"
uitcp.tunnel.links.2>dev
uitcp.tunnel.links.2.dev = s2
= "s2"
uitcp.tunnel.links.2>restart
uitcp.tunnel.links.2.restart>_i
uitcp.tunnel.links.2.restart._.insert = 1
uitcp.tunnel.links.2.restart>1
uitcp.tunnel.links.2.restart.1>sc
```

```
uitcp.tunnel.links.2.restart.1.script = config-nsg port s2 adm down; config-nsg port s2 adm up
= "config-nsg port s2 adm down; config-nsg port s2 adm up"
uitcp.tunnel.links.2.restart.1>(пробел)(enter)
uitcp.tunnel.links.2.restart>(пробел)(enter)
uitcp.tunnel.links.2>(пробел)(enter)
uitcp.tunnel.links>p
uitcp.tunnel.links.priority = 300
= 300
```

Переход в корень конфигурации и её применение:

```
uitcp.tunnel.links>(пробел)(пробел)(enter)
uitcp>_a
Config to apply
{
    mode = "client",
    tunnel = {
        description = "test ATM",
        keepalive = 10,
        links = {
            [1] = {
                description = "SKYLINK",
                dev = "s1",
                restart = {
                    [1] = {
                    script = "config-nsg port s1 adm down; config-nsg port s1 adm up",
                    timeout = 180
                    }
                }
            },
            [2] = {
                description = "MEFAGON",
                dev = "s2",
                restart = {
                    [1] = {
                    script = " config-nsg port s2 adm down; config-nsg port s2 adm up ",
                    timeout = 120
                    }
                }
            },
            priority = 300
        },
        name = "ATM001",
        servers = {
            [1] = "123.45.67.89:50005"
        }
    }
}
Are you sure?[yes/no] yes
uitcp>
```

Теперь в логе сервера должны быть видны попытки соединения со стороны клиента:

```
2010-02-15 00:29:09 INFO :Accept connection X.X.X.X:Y>>123.45.67.89:50005
2010-02-15 00:29:09 WARN :[test ATM]Can't resume tunnel, no such tunnel
2010-02-15 00:29:10 INFO :[test ATM]New tunnel initiated from X.X.X.X:Y
2010-02-15 00:29:10 INFO :[test ATM]Certificate /etc/uitcp/certs/server.pem expires in 365 days
2010-02-15 00:29:10 INFO :[test ATM]Start SSL handshake
2010-02-15 00:29:12 INFO :[test ATM]SSL handshake successfully done
```

Если соединение успешно, то можно переходить к следующему этапу.

ПРИМЕЧАНИЕ Если конфигурация выполняется с помощью Web-интерфейса, то следует воспроизвести в нём дерево конфигурации, показываемое выше по команде _apply. Нужные элементы дерева создаются (+/−) или активируются (○/●) по мере необходимости. Для сохранения и применения изменений необходимо нажать кнопку .

6б. Создание туннеля *ui*TCP (вариант 2×GPRS)

Настройка SSL, журналов и сервера выполняется так же, как в предыдущем случае. Конфигурация туннеля на клиентском устройстве (для краткости приведён лишь конечный результат):

```
{
    mode = "client",
    tunnel = {
        description = "ATM001",
        links = {
            [1] = {
                description = "GPRS",
                keepalive = 10,
                dev = "s1",
                dualSim = true,
                restart = {
                    [1] = {
                        description = "MTS",
                        script = "config-nsg port s1 prio main none aux 1"
                    },
                    [2] = {
                        description = "BEELINE",
                        script = "config-nsg port s1 prio main 1 aux none"
                    },
                    timeout = 120
                }
            }
        },
        name = "ATM001",
        servers = {
            [1] = "123.45.67.89:50005"
        }
    }
}
```

Особенность данной конфигурации состоит в скриптах, используемых для управления сотовым модулем. При разрыве соединения с МТС (верхняя SIM-карта) порт переконфигурируется для работы только с нижней SIM-картой и начинает работать через Билайн; при разрыве соединения с Билайн — наоборот. Приоритет между двумя каналами связи выключен, поскольку услуги обоих операторов предполагаются примерно равными по стоимости и по качеству.

6в. Создание туннеля *ui*TCP (вариант Ethernet+GPRS)

Настройка SSL, журналов и сервера выполняется так же, как в предыдущих двух случаях. Конфигурация туннеля на клиентском устройстве:

```
{
    mode = "client",
    tunnel = {
        description = "test ATM",
        keepalive = 10,
        links = {
            [1] = {
                description = "Ethernet",
                gw = "98.76.54.29"
            },
            [2] = {
                description = "MEFAGON",
                dev = "s2",
                restart = {
                    [1] = {
                    script = " config-nsg port s2 adm down; config-nsg port s2 adm up "
                    }
                }
            },
            priority = 300
```

```
            },
            name = "ATM001",
            servers = {
                [1] = "123.45.67.89:50005"
            }
        }
    }
```

В конфигурации основного канала связи указан не интерфейс, а IP-адрес следующего шлюза, как это необходимо для Ethernet.

Рестартовать порт Ethernet бессмысленно, поскольку проблема находится, как правило, где-то дальше по каналу связи. (А в данном случае это и не возможно, поскольку все внешние порты коммутируются на разные VLAN на одном физическом порту процессора, и рестартовать можно только их все вместе.) Опционально можно добавить какой-нибудь информационный скрипт, например, для переключения светодиодных индикаторов, удобных для восприятия находящейся при данном устройстве блондинкой:

```
            links = {
                [1] = {
                    ...................................,
                    restart = {
                        [1] = {
                        script = " echo test-off > /proc/sys/nsg/led/l1/client; echo test-on > /proc/sys/nsg/led/l2/client "
                        }
                    }
                },
                [2] = {
                    ...................................,
                    restart = {
                        [1] = {
                            script = " config-nsg port s2 adm down; config-nsg port s2 adm up;
echo test-on > /proc/sys/nsg/led/l1/client; echo test-off > /proc/sys/nsg/led/l2/client "
                        }
                    }
                }
```

В данном случае настраиваемый индикатор LED1 будет гореть при работе через Ethernet, LED2 — при работе через GPRS.

7. Настройка соединений в туннеле

Заключительный этап настройки *ui*TCP — создание TCP-соединений в построенном туннеле. Эта часть конфигурации уже никак не зависит от числа и типа используемых каналов связи и является общей для всех трёх типовых задач.

Соединения могут устанавливаться как в направлении от клиента к серверу, так и наоборот, поэтому здесь важно уже не отношение клиент/сервер (они относятся только к функционированию туннеля в целом), а отношение вызывающая/отвечающая сторона. На вызывающей стороне создаётся TCP-сокет, принимающий локальные соединения из сети источника. На отвечающей стороне настраиваются правила NAT, определяющие, с какими IP-адресами и номерами портов TCP пакеты будут препровождаться в сеть назначения.

Учитывая специфику массовых инсталляций, действительные адреса и порты, используемые для работы с конкретной площадкой, удобно прописывать на сервере *ui*TCP.

Настройка туннеля от банкомата к процессинговому серверу. Напомним, что на процессинге данный банкомат зарегистрирован под IP-адресом 10.0.1.1 и должен обращаться на порт TCP 20001; в то же время в типовой конфигурации всех банкоматов записаны адрес 192.168.0.1 и порт 20000.

На клиенте:

```
uitcp.tunnel>lo
uitcp.tunnel.localListener>_i
uitcp.tunnel.localListener._.insert = 1
uitcp.tunnel.localListener>1
uitcp.tunnel.localListener.1>h
uitcp.tunnel.localListener.1.host = 192.168.0.1
= "192.168.0.1"
uitcp.tunnel.localListener.1>p
uitcp.tunnel.localListener.1.port = 20000
= 20000
uitcp.tunnel.localListener.1>(пробел)(enter)
uitcp.tunnel.localListener>(пробел)(enter)
uitcp.tunnel>_a
Config to apply
.....
localListener = {
    [1] = {
        host = "*",
        port = 20000
    }
}
.....
Are you sure?[yes/no]
```

На сервере:

```
uitcp>clients
uitcp.clients>A
uitcp.clients.ATM001>n
uitcp.clients.ATM001.nat>_i
uitcp.clients.ATM001.nat._.insert = 1
uitcp.clients.ATM001.nat>1
uitcp.clients.ATM001.nat.1>inDstP
uitcp.clients.ATM001.nat.1.inDstPort = 20000
= 20000
uitcp.clients.ATM001.nat.1>outDstA
uitcp.clients.ATM001.nat.1.outDstAddr = 10.0.0.1
= "10.0.0.1"
uitcp.clients.ATM001.nat.1>outDstP
uitcp.clients.ATM001.nat.1.outDstPort = 20001
= 20001
uitcp.clients.ATM001.nat.1>outS
uitcp.clients.ATM001.nat.1.outSrcAddr = 10.0.1.1
= "10.0.1.1"
uitcp.clients.ATM001.nat.1>(пробел)(enter)
uitcp.clients.ATM001.nat>(пробел)(enter)
uitcp.clients.ATM001>_a
Config to apply
{
    nat = {
        [1] = {
        inDstPort = 20000,
        outDstAddr = "10.0.0.1",
        outDstPort = 20001,
        outSrcAddr = "10.0.1.1"
        }
    }
}
Are you sure?[yes/no]
```

Настройка соединения от административной станции к банкомату. Станция обращается серверу NSG по адресу 10.0.0.7 и номеру порта TCP 30001 (для других площадок — 30002 и т.д.) в локальной сети банка; это соединение принимается сервером *ui*TCP и транслируется на удалённую площадку. Для краткости приведены только результирующие фрагменты конфигураций:

На клиенте:

```
uitcp.tunnel>_show
.....
    nat = {
        [1] = {
        inDstPort = 30001,
        outDstAddr = "192.168.0.2"
        outDstPort = 23,
        }
    }
.....
```

На сервере:

```
uitcp.clients.ATM001>_show
.....
    localListener = {
        [1] = {
            host = "10.0.0.7",
            port = 30001
        }
    }
.....
```

Настройка соединения от административной станции к устройству NSG. Станция обращается по адресу 10.0.0.7 и номеру порта TCP 40001 (40002 и т.д.) и попадает по Telnet на клиентское устройство:

На клиенте:

```
uitcp.tunnel>_show
.....
    nat = {
        [2] = {
        inDstPort = 40001,
        outDstAddr = "127.0.0.1",
        outDstPort = 23
        }
    }
.....
```

На сервере:

```
uitcp.clients.ATM001>_show
.....
    localListener = {
        [2] = {
            host = "10.0.0.7",
            port = 40001
        }
    }
.....
```

8. Настройка HTTPS (опционально)

Для безопасного доступа к серверу *ui*TCP из внешнего мира рекомендуется использовать HTTPS с двусторонним обменом сертификатами X.509. В рамках уже готовой инфраструктуры для работы с самоподписанными сертификатами это настраивается достаточно несложно.

Сгенерировать сертификат и ключ для пользователя basile при помощи готового скрипта:

```
root@uitcp_srv # /etc/uitcp/bin/tools/cert-client basile
```

Перейти в директорию с сертификатами пользователя и распаковать архив cert-client.basile.tgz (поскольку скрипт удаляет файлы, не нужные на сервере):

```
root@uitcp_srv # cd /etc/uitcp/certs/basile
root@uitcp_srv basile # tar -xzf cert-client.basile.tgz
```

Преобразовать сертификаты и ключи в формат, пригодный для установки на персональный компьютер или ноутбук удалённого администратора:

```
root@uitcp_srv basile # openssl x509 -in root.pem -outform DER -out root.der
root@uitcp_srv basile # openssl pkcs12 -export -out basile.p12 -in client.pem -inkey clientkey.pem ↵
-name basile@uitcpsrv
```

(В последующих версиях *ui*TCP для подготовки сертификатов пользователей будет сделан отдельный скрипт.) По ходу выполнения последней команды будет запрошен пароль для защиты файла basile.p12; этот пароль потребуется позже для установки сертификата на ПК пользователя.

Создать файл /etc/stunnel/stunnel.conf следующего вида:

```
CAfile = /etc/uitcp/certs/_root/root.pem
key=/etc/uitcp/certs/serverkey.pem
cert=/etc/uitcp/certs/server.pem
options = NO_SSLv2
options = ALL
sslVersion = SSLv3
verify = 2
debug = 3
output = /var/log/ssl.log
pid = /var/run/https.pid
foreground = yes

[stunnel]
    accept=443
    connect=127.0.0.1:80
```

Добавить в inittab строку для запуска stunnel:

```
root@uitcp_srv # echo null::respawn:/usr/sbin/stunnel >> /etc/inittab
```

Сохранить полученную конфигурацию и рестартовать устройство:

```
root@uitcp_srv # savecfg
root@uitcp_srv # reboot
```

Полученные сертификаты удостоверяющего центра root.der и данного пользователя basile.p12 нужно выгрузить (безопасным способом) на ПК пользователя и установить в Web-броузер на нём. В Firefox это делается в меню Настройки —> Дополнительно —> Шифрование —> Просмотр сертификатов. Следует проверить, что установлена опция "Использовать SSL 3.0". Для первого входа необходимо также установить опцию "Сертификаты… Спрашивать каждый раз"; после первого входа (сертификат будет запрошен около десятка раз на одной странице — при каждом HTTPS-запросе) восстановить эту опцию в значение "Отправлять автоматически".

Для Microsoft Internet Explorer сертификаты устанавливаются в общесистемное хранилище с помощью мастеров, которые запускаются просто двойным щелчком (или через меню Сервис —> Свойства обозревателя —> Содержание —> Сертификаты). После этого необходимо на странице Сервис —> Свойства обозревателя —> Дополнительно отключить опцию SSL 2.0 и проверить, что включена опция SSL 3.0.

9. Настройка фильтров

Для безопасной работы любых устройств, подключённых к Интернет, необходимо настроить на всех внешних интерфейсах фильтры, отсекающие все посторонние пакеты. Настройка фильтров выполняется в основном ПО. Пример настройки для вышеприведённых конфигураций:

На клиенте:

```
!
nsg
    access-list std-ip 1
        add 1 permit host 123.45.67.89
        add 2 permit host 194.149.67.130
        exit
    access-list std-ip 2
        add 1 deny any
        exit
    port s1
        access-group local input 1
        access-group transit input 2
        exit
    port s2
        access-group local input 1
        access-group transit input 2
        exit
```

На сервере:

```
!
nsg
    access-list ext-ip 101
        add 1 permit ip host 194.149.67.130 host 123.45.67.89
        add 2 permit tcp any host 123.45.67.89 eq 50005
        add 3 permit tcp any host 123.45.67.89 eq 50006
        add 4 permit tcp any host 123.45.67.89 eq 443
        exit
    access-list std-ip 2
        add 1 deny any
        exit
    port s1
        access-group local input 1
        access-group transit input 2
        exit
```

(порт 50005 используется *ui*TCP, по умолчанию, для приёма входящих соединений от клиентов, 50006 — для обновления *ui*TCP на клиентах)

10. Клонирование конфигурации

Для переноса полученной конфигурации на другие клиентские устройства рекомендуется следующая последовательность действий:

1. Сохранить полученную конфигурацию клиентского устройства при помощи утилиты nsg-config-export. Утилита архивирует целиком директорию /etc (вместе со всеми настройками основного ПО, файлами и настройками *ui*TCP, файлами сертификатов и стартовыми файлами) и выгружает её на указанный сервер TFTP или FTP.
2. Восстановить заводскую конфигурацию второго клиентского устройства и загрузить на него сохранённую конфигурацию с помощью утилиты nsg-config-import.
3. Изменить на втором устройстве имя клиента на какое-нибудь нейтральное (например, XXX), чтобы не возникало конфликтов при одновременной работе двух клиентов под одним именем. Удалить файлы сертификатов и сохранить изменения:

   ```
   rm -r /etc/uitcp/certs/*
   savecfg
   ```

4. Сохранить полученную конфигурацию при помощи nsg-config-export. Эта конфигурация готова к загрузке на третье и все последующие клиентские устройства.
5. Установить новое имя клиента (например, ATM002). Сгенерировать на сервере набор сертификатов для него, загрузить на клиента и установить. Рестартовать устройство. Эти же уникальные настройки выполняются на третьем и последующих клиентах.

ПРИМЕЧАНИЕ В зависимости от конфигурации, у каждого клиента могут быть также другие уникальные параметры: статические IP-адреса, имя/пароль для аутентификации у сотового оператора (в частности, для получения статического адреса в Скайлинк) и т.п.

ООО «Эн–Эс–Джи»
Россия 105187 Москва
ул. Вольная, д.35
Тел./факс: (+7–495) 727–19–59 (многоканальный)

http://www.nsg.ru/
mailto:info@nsg.net.ru
mailto:sales@nsg.net.ru
mailto:support@nsg.net.ru