

## **Отчет о тестировании**

# **NSL4 – Network Service Library ver.4.**

Заказчик: ООО “СТЦ”

Исполнитель: ООО “УНИС”

**Содержание:**

- 1. Введение**
- 2. Описание тестируемой системы, NSL4**
- 3. Требования к надежности**
- 4. Требования к составу и параметрам технических средств**
- 5. Требования к информационной и программной совместимости**
- 6. План тестирования**
- 7. Тестирование API с использованием программы nsI4\_messenger\_test.**
  - 7.1 Функциональные возможности программы**
  - 7.2 Описание работы с программой**
  - 7.3 Проверка подключения к серверу**
    - 7.3.1 Тесты подключения к серверу
    - 7.3.2 Проверка регистрации функций обратного вызова информационных сообщений.
    - 7.3.3 Проверка регистрации функций обратного вызова для данных.
    - 7.3.4 Проверка регистрации функций обратного вызова для данных. В процессе работы клиента.
    - 7.3.5 Проверка отсоединения от сервера.
  - 7.4 Проверка отправки и получения сообщений.**
    - 7.4.1 Проверка отправки сообщений.
  - 7.5 Информация от сервера.**
    - 7.5.1 Получение списка клиентов.
    - 7.5.2 Запрос описания клиента.
    - 7.5.3 Проверка событий.
- 8. Тесты на скорость, с использованием программы nsI4\_speed\_test.**
  - 8.1 Функциональные возможности программы
  - 8.2 Описание работы с программой
  - 8.3 Результаты тестирования
- 9. Тесты на скорость передачи данных серверу и обратно, с использованием программы nsI4\_ping\_test.**
  - 9.1 Функциональные возможности программы
  - 9.2 Описание работы с программой
  - 9.3 Расчет времени задержки

## Введение

### О данном документе

Данное руководство предназначено для тестирования (ПО) NSL4. В нем содержится информация, необходимая пользователю для настройки и проведения тестов в различных программах. Так же данное руководство содержит результаты тестов.

## 2 Описание тестируемой системы, NSL4

Тестируемая система NSL4 предназначена для организации сетевого взаимодействия программ на прикладном уровне с использованием протоколов транспортного уровня (TCP, UDP). NSL4 построено по технологии «клиент-сервер» и состоит из серверной части (сервер) и клиентских библиотек (клиент).

## 3 Требования к надежности

NSL4 должен обеспечивать надежное функционирование всех своих составных частей. В случае возникновения внештатных ситуаций, связанных с некорректно заданными параметрами или с неустраняемыми ошибками во входных данных, должны возвращаться соответствующие коды ошибок, описанные в документации.

При разрыве соединения клиент должен оповестить использующее его приложение в рамках программного интерфейса (API), изложенного в прил. 1., и устанавливать повторное соединение с сервером.

В режиме «клиент-клиент» NSL4 должен обеспечивать скорость передачи данных не ниже 70% от скорости сетевого соединения.

При работе двух клиентов на одной ОС семейства MS Windows должна быть обеспечена возможность осуществлять передачу данных со скоростью не менее 300 Мбит/с.

ПО должно обеспечивать следующие максимально допустимые задержки при передаче 1000 сообщений по 100 байт от одного клиента к другому и обратно:

Сценарий	Макс. задержка, мкс	Ср. задержка, мкс
сервер и оба клиента на одном компьютере	330	100
сервер и клиент на одном компьютере, второй клиент на другом компьютере	1700	1000

ПО должно обеспечивать следующие минимально допустимые скорости передачи данных от одного клиента к другому и обратно в случае, когда клиент и сервер запущены на одном компьютере, а второй клиент запущен на другом компьютере:

Размер сообщения, байт	Мин. допустимая скорость, Кбит/с	Мин. кол-во отосланных блоков в секунду.
10	430	5100
100	3800	4800
1000	38000	4800
10000	91500	1150
100000	92000	115

Значения приводятся для типовой конфигурации (см. разд. 4.4).

ПО должно корректно использовать динамическое распределение и освобождение памяти. Отсутствие утечек памяти должно быть показано на тестовых примерах.

При ошибках в работе аппаратных средств, подаче электроэнергии, а также ошибках, связанных со сторонним программным обеспечением (драйвера устройств, модули операционной системы), восстановление функций ПО NSL4 возлагается на операционную систему.

#### 4. Требования к составу и параметрам технических средств

ПО NSL4 должно полноценно функционировать на компьютере со следующей типовой конфигурацией: Intel Core i5 2ГГц, 4Гб ОЗУ, сетевой интерфейс 1 Гбит/с.

#### 5. Требования к информационной и программной совместимости

Поддерживаемые операционные системы:

Модуль	Список ОС
Сервер	<ul style="list-style-type: none"> <li>• MS Windows NT 5.1 и выше</li> <li>• GNU Linux 2.4 и выше (MCBC 3.0 80001-12 изм. 4 и выше, AstraLinux SE)</li> </ul>
Клиент	<ul style="list-style-type: none"> <li>• MS Windows NT 5.1 и выше</li> <li>• GNU Linux 2.4 и выше (MCBC 3.0 80001-12 изм. 4 и выше, AstraLinux SE)</li> <li>• Android 2.2 и выше</li> </ul>

*Среды разработки и компиляторы*

1. Microsoft Visual C++ 2010
2. GCC 4.1.X (Linux)
3. Android NDK (2.2.X)

**Используемые сторонние библиотеки**

Qt 4.6.3

## 6. План тестирования

**Проверка работы функций API.** В план тестирования API входит

- 1) Тестирование подключения в различных режимах.
- 2) Проверка работы функций обратного вызова.
- 3) Отключение от сервера.
- 4) Проверка передачи и получения сообщений.
- 5) Получение данных с информацией о состоянии сервера.

**Замер скорости передачи и получения данных.**

**Замер времени задержек.**

## 7. Тестирование API с использованием программы nsl4\_messenger\_test.

### 7.1 Функциональные возможности программы

Программа позволяет проверить:

- 1) Подключение клиентов к серверу в различных режимах.
- 2) Получение и передача сообщений.
- 3) Получение информационных данных от сервера.

### 7.2 Описание работы с программой

После запуска программы, на экране появится главное окно программы (Рис. 7.1).

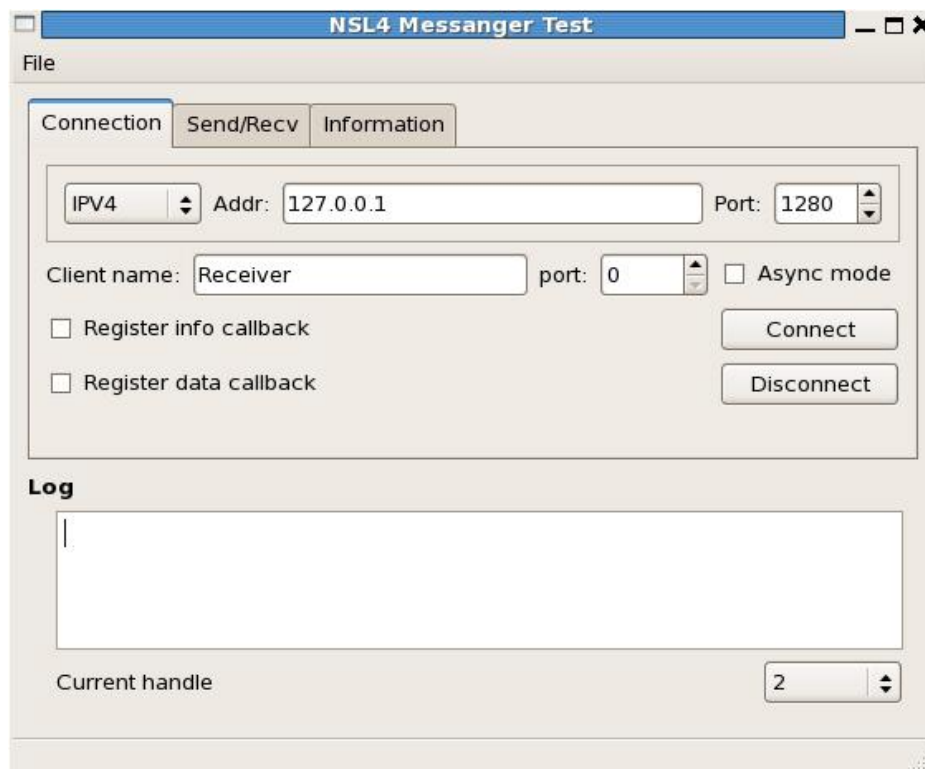


Рис. 7.1. Главное окно.

Программа разбита на три основных части. Панель Connection отвечает за проверку подключения клиентов к серверу, позволяет подключить клиентов в синхронном или асинхронном режиме, подключить функции обратного вызова,

отсоединить клиента. Панель SendRecv позволяет проверить прием и передачу сообщений, отправленных как одному клиенту, так и группе клиентов. Панель Information, позволяет запросить информацию: о клиенте, о списке клиентов.

### 7.3 Проверка подключения к серверу

Для подключения к серверу необходимо перейти в панель «Connection».

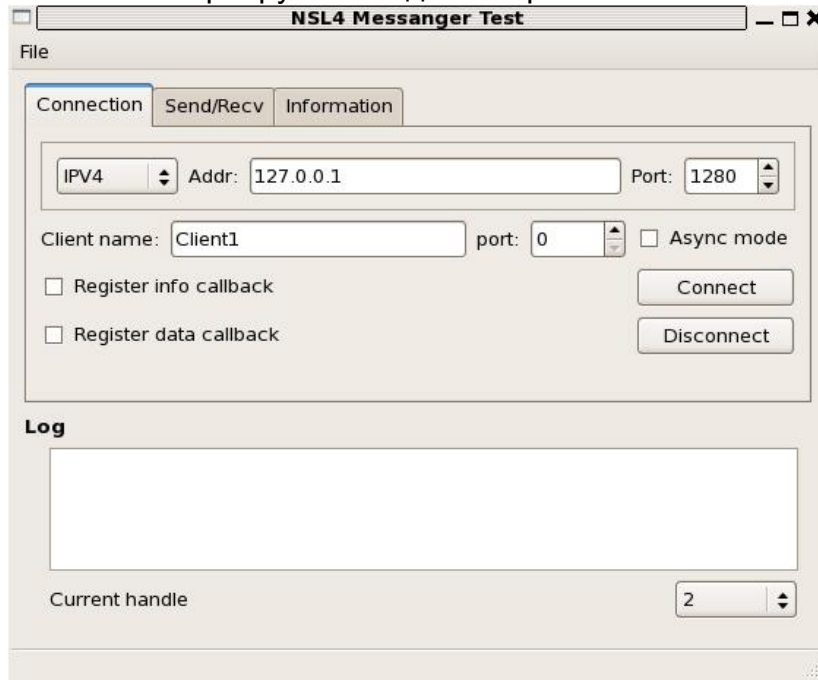


Рис. 7.3. Панель «Connection».

Далее необходимо:

- Выбрать версию протокола (IPV4/IPV6).
- Задать IP адрес.
- Задать номер порта.
- Задать имя клиента

Дополнительные параметры:

- Задать порт для клиента.
- Выбрать режим подключения(синхронный/асинхронный)
- Зарегистрировать функцию обратного вызова для информационных сообщений
- Зарегистрировать функцию обратного вызова для данных

### 7.3.1 Тесты подключения к серверу

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1  
Port: 0

4) Для первого теста не использовали функции обратного вызова, подключение производилось в синхронном режиме. Ряд проведенных тестов с различными комбинациями флагов, можно посмотреть в таблице ниже.

Sync mode	Register info callback	Register data callback	Connect
true	false	false	ok
true	false	true	ok
true	true	false	ok
true	true	true	ok
false	false	false	ok
false	false	true	ok
false	true	false	ok
false	true	true	ok

### 7.3.2 Проверка регистрации функций обратного вызова.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1  
Port: 0  
Sync mode: false  
Register info callback: true  
Register data callback: false
- 4) Подключаемся и в поле **log** отображается сообщение, о том что функция обратного вызова зарегистрирована.



### 7.3.3 Проверка регистрации функций обратного вызова для данных.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1  
Port: 0  
Sync mode: false  
Register info callback: false  
Register data callback: true
- 4) Запускаем второй MessengerTest с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1  
Port: 0  
Sync mode: false  
Register info callback: false  
Register data callback: false

#### Описание теста.

Для проверки необходимо запустить две программы MessengerTest. В первой программе MessengerTest подключаем клиента и регистрируем data callback. Во второй программе MessengerTest подключаем клиента без регистрации data callback. Далее отправляем сообщение, к клиенту у которого зарегистрирован data callback, в программе MessengerTest сразу отобразится результат. Это значит, что сообщение не хранится в буфере, а сразу идет на обработку функции обратного вызова.

### 7.3.4 Проверка регистрации функций обратного вызова для данных. В процессе работы клиента.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: Client1  
Port: 0  
Sync mode: false  
Register info callback: false  
Register data callback: false
- 3) Запускаем второй MessengerTest с параметрами:  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: Client2  
Port: 0  
Sync mode: false  
Register info callback: false  
Register data callback: false

#### Описание теста.

Для проверки необходимо запустить две программы MessengerTest.

Параметры и заполнение полей в программах можно посмотреть на Рис. 7.3.4.

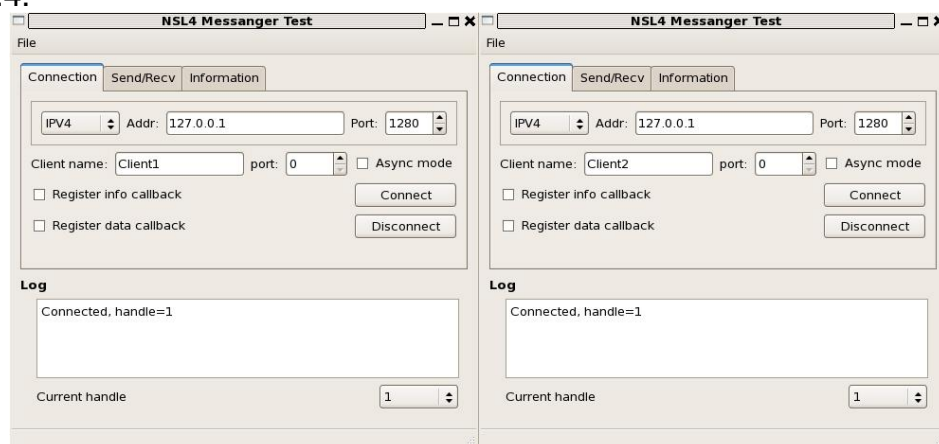


Рис. 7.3.4 Параметры программ MessengerTest.

В программе NSL4 Monitor мы видим два подключенных клиента Рис. 7.3.5



Id	Name	Version	Address	Send, b/s	Receive, b/s
23	NSL4 Monitor	4.0.0.1	127.0.0.1	0	0
9	Client1	4.0.0.1	127.0.0.1	0	0
16	Client2	4.0.0.1	127.0.0.1	0	0

Clients: 3

Рис. 7.3.5. NSL4 Monitor

Для того, что бы отправить сообщения Client2, в первой программе MessengerTest переходим на вкладку «SendRecv» и отправляем несколько сообщений клиенту Client2, у которого ID=16, это видно из программы NSL4 Monitor Рис. 7.3.5.

Далее клиенту Client2 регистрируем функцию обратного вызова Рис. 7.3.6.

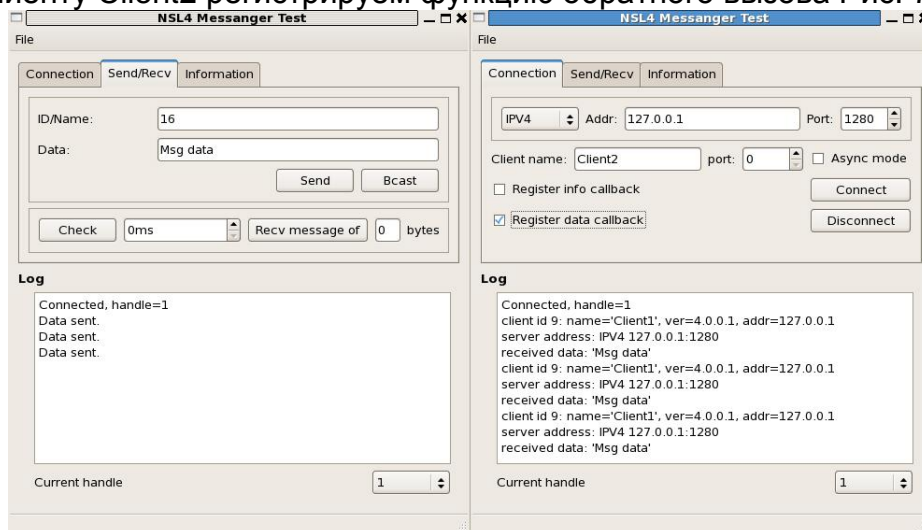


Рис. 7.3.6. Параметры программ MessengerTest.

После чего в поле **Log** отображается информация о полученных сообщениях. Это значит, что сообщения, которые хранились в буфере, после регистрации функции обратного вызова, незамедлительно пошли через функцию и считались из буфера.

### 7.3.5 Проверка отсоединения от сервера.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest с параметрами  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1  
Port: 0
- 4) В открытом MessengerTest подключаем второго клиента  
Протокол IPV4  
Addr: 127.0.0.1  
Port 1280  
Client name: client2  
Port: 0

Для отключения клиента необходимо выбрать Current handle данного клиента и нажать кнопку Disconnect. Отключение обоих клиентов прошло успешно.

Убедиться в этом можно посмотрев список подключенных клиентов в программе NSL4 Monitor.

#### 7.4 Проверка передачи и получения сообщений.

Для работы необходимо перейти в панель «SendRecv». Рис. 7.4.

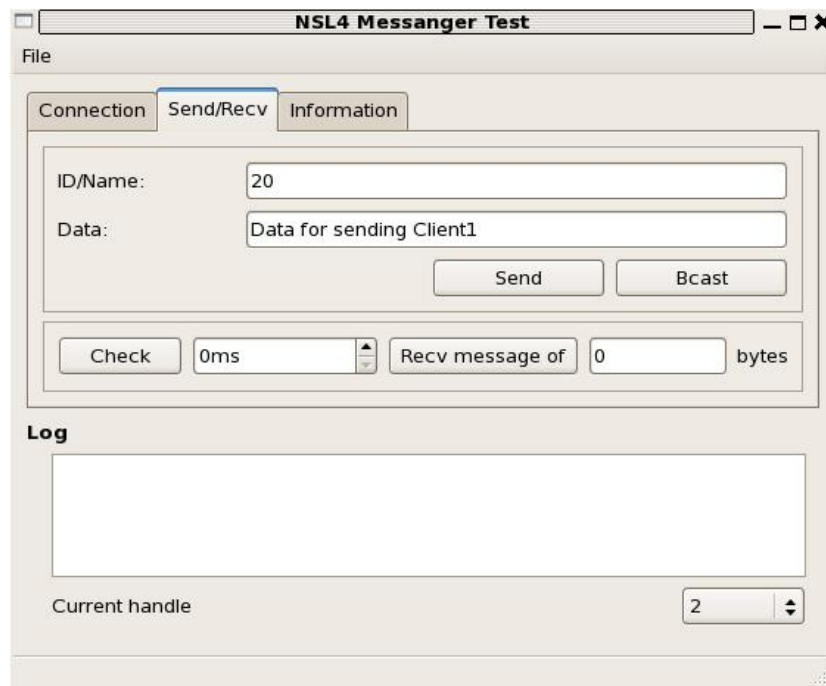


Рис. 7.4. Панель «SendRecv».

Далее необходимо указать:

- Идентификатор или имя клиента.
- Данные для отправки.

Дополнительные параметры:

- Указать задержку.
- Количество данных, которые хотим получить

**7.4.1 Проверка отправки сообщений.****Используемые инструменты для теста.**

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

**Условия и параметры.**

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами  
 Протокол IPV4.  
 Addr: 127.0.0.1  
 Port 1280
- 3) Запускаем MessengerTest с параметрами  
 Протокол IPV4.  
 Addr: 127.0.0.1  
 Port 1280  
 Все тесты проведены с протоколами IPV4  
 В таблице приведены тесты для IPV4  
 Client name(id): client1(9), client1(15), client1(18), client2(20)  
 Port: 0  
 Async mode: false  
 Register info callback: false  
 Register data callback: false

В таблице 7.4.1 приведены условия и результаты отправки сообщений.

В первом тесте отсылалось одно сообщение, размером 24 байта, клиенту client2, у которого id = 20, эту информацию можно посмотреть в NSL4 Monitor. Проверяем количество сообщений, получили одно сообщение, размером 24 байта. Во втором тесте отсылалось сообщение размером 24 байта, а при получении указывали 20 байт, программа вернула сообщение, обрезанное до 20 байт. В третьем тесте отослали 10 сообщений по 24 байта, все 10-ть сообщений приняты в полном размере. В четвертом тесте использовалась рассылка по имени Bcast. Имя клиента client1, у каждого клиента свой id, размер сообщения 24 байт, все клиенты получили сообщение, размером 24 байт. В пятом тесте передача сообщений по имени осуществлялась несколько раз, все клиенты получили сообщения успешно.

Кол-во исходящих сообщений	Размер сообщения в байтах	Получатели	Кол-во полученных сообщений	Размер каждого сообщения	Целостность сообщения
1	24	client2(20)	1	24	true
1	24	client2(20)	1	20	false
10	24	client2(20)	10	24	true
1	24	client1(9), client1(15), client1(18)	1	24	true
10	24	client1(9), client1(15), client1(18)	10	24	true

Таблица 7.4.1 Условия и результаты отправки сообщений

## 7.5 Информация от сервера.

Для работы необходимо перейти в панель «Information». Рис. 7.5.

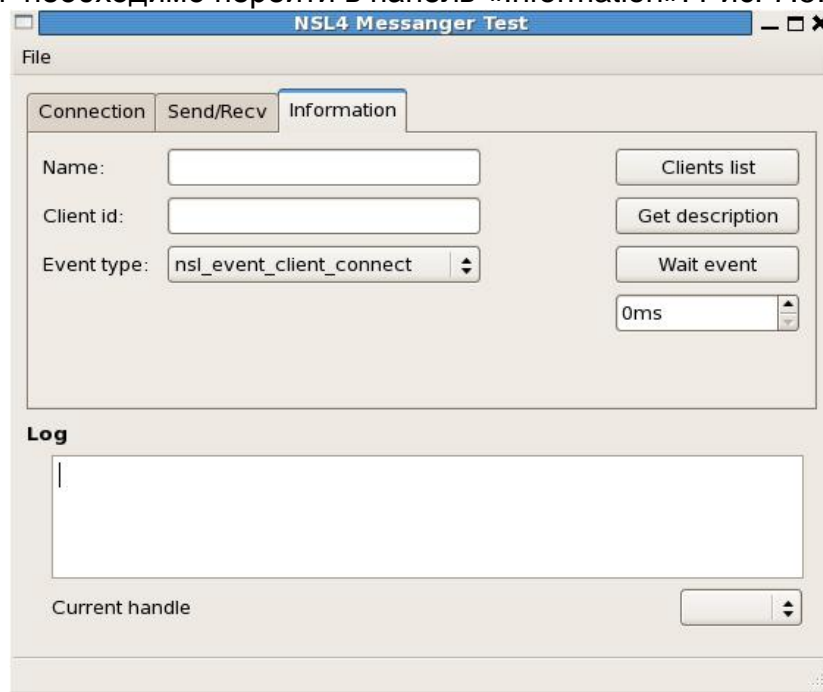


Рис. 7.4. Панель «Information».

Далее необходимо указать:

- Имя клиента.
- Идентификатор клиента.

### 7.5.1 Получение списка клиентов.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest с параметрами  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 4) Подключаем три клиента с именем Client name: client1, и подключаем клиента с именем Client name: client2.
- 5) Переходим в панель «Information».
- 6) Запрашиваем информацию о клиентах. В информационном поле видим программу монитор, которая запущена как клиент, три клиента client1, один клиент client2.
- 7) Для вызова информации о клиентах с определенным именем, вводим необходимое имя в поле name: client1. Запрашиваем информацию о клиентах. В информационном поле видим программу монитор, которая запущена как клиент, три клиента с именем client1.



### 7.5.2 Запрос описания клиента.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest и подключаем клиента  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1
- 4) Из программы NSL4 Monitor, берем идентификатор клиента.
- 5) В программе MessengerTest переходим на вкладку «Information» и указываем идентификатор клиента, нажимаем Get description, в информационном поле отображается информация по данному клиенту.

### 7.5.3 Проверка событий.

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер nsl4-server.

В качестве программы тестирования используем NSL4 MessengerTest.

Для просмотра результата тестирования используем NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.
- 2) Запускаем программу NSL4 Monitor. И подключаемся с параметрами  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280
- 3) Запускаем MessengerTest и подключаем клиента  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client1
- 4) В панели «Information»,  
Ставим задержку 10000ms.  
Event type: nsl\_event\_client\_connect
- 5) Запускаем второй MessengerTest и подключаем клиента  
Протокол IPV4.  
Addr: 127.0.0.1  
Port 1280  
Client name: client2
- 6) В программе MessengerTest, где открыта панель «Information», нажимаем кнопку «Wait event», в информационном поле список параметров о подключенном client2, поле идентификатора пустое, так как это запрос на подключение
- 7) В программе MessengerTest, где открыта панель «Information», выбираем event type: nsl\_event\_server\_connect, нажимаем кнопку «Wait event», в информационном поле список параметров о подключенном client2, поле идентификатора уже заполнено корректно, так как сервер присвоил данному клиенту уникальный идентификатор.
- 8) В программе MessengerTest, где подключен client2, нажимаем «Disconnect». Переходим в программу MessengerTest, где открыта панель «Information», выбираем event type: nsl\_event\_client\_disconnect, нажимаем кнопку «Wait event», в информационном поле список параметров об отсоединившемся client2.

## 8. Тесты на скорость, с использованием программы `ns14_speed_test`.

### 8.1 Функциональные возможности программы

Программа позволяет протестировать:

- 1) Скорость получения и передачи данных.

### 8.2 Описание работы с программой

Программа `ns14_speed_test` имеет следующие опции:

-?, -h, --help

Вывод справки и выход.

-d, --domain

Тип протокола (IPV4, IPV6).

-a, --address

IP адресс.

-p, --port

Порт сервера.

-s, --sender

Режим запуска программы(для отправки сообщений).

-r, --receiver

Режим запуска программы(для получения сообщений).

-c, --client-id

Для режима отправки сообщений, необходимо указать Id клиента, которому осуществляется передача сообщений

-m, --message -size

Для режима отправки сообщений, можно указать размер сообщений, по умолчанию 100 байт.

-n, --sending-attempts-num

Для режима отправки сообщений, можно указать количество сообщений, по умолчанию 1000 сообщений отправляется.

-t, --sending-time-duration

Задержка перед каждой отправкой данных в мс. (0 по умолчанию).

### 8.3 Замер скорости

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер `ns14-server`.

В качестве программы тестирования `ns14_speed_test`.

Для просмотра результата тестирования NSL4 Monitor.

#### Условия и параметры.

- 1) Запускаем сервер, указываем порт 1280.

- 2) Запускаем программу `ns14_speed_test`, для получения сообщений, со следующими параметрами.

Протокол IPV4.

Addr: 127.0.0.1

Port 1280

Режим: receiver

`./ns14_speed_test -d IPV4 -p 1280 —receiver`

3) Запускаем программу NSL4 Monitor, подключаемся и смотрим id клиента, который запущен для получения сообщений.

3) Запускаем программу `ns14_speed_test`, для отправки сообщений, со следующими параметрами.

Протокол IPV4.

Addr: 127.0.0.1

Port 1280

`./ns14_speed_test -d IPV4 -a 127.0.0.1 -p 1280 --sender -c 8 -m 100 -n 1000`

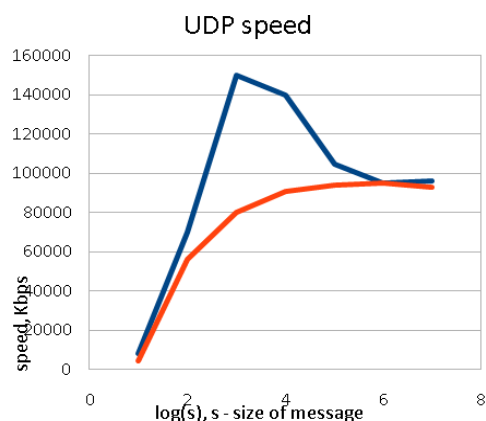
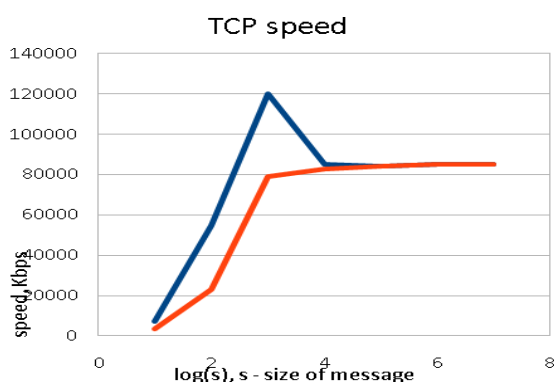
В данном примере клиенту с id=8 отправлено 1000 сообщений, размер сообщений 100 bytes, ниже приведены тесты для 1000 сообщений.

Задержки (1000 сообщений of 100 байт):

	TCP, average/maximal (mks)	UDP, average/maximal (mks)
Local	32 / 15625	62 / 15625
100Mbit eth	35 / 1340	65 / 600

Скорость (100Mbit eth):

Message size (bytes)	TCP send / recv(Kbps)	UDP send / recv (Kbps)
10	2500 / 2280	1200 / 1150
100	25600 / 21780	12800 / 11450
1000	91300 / 86000	21000 / 20050
10 <sup>4</sup>	97000 / 93500	89800 / 89400
10 <sup>5</sup>	95000 / 94600	90940 / 90920
10 <sup>6</sup>	95000 / 94000	90940 / 90800
10 <sup>7</sup>	92300 / 91400	90350 / 90330



## 9. Тесты на скорость передачи данных туда и серверу, с использованием программы `nsl4_ping_test`.

### 9.1 Функциональные возможности программы

Программа позволяет протестировать:

1) Время задержки.

### 9.2 Описание работы с программой

Программа `nsl4_ping_test` имеет следующие опции:

-?, -h, --help

Вывод справки и выход.

-d, --domain

Тип протокола (IPV4, IPV6).

-a, --address

IP адресс..

-p, --port

Порт сервера.

-e, --echor

Режим запуска программы(слушать или отправлять сообщения)

-c, --echo-client-id

Для режима отправки сообщений, необходимо указать Id клиента, которому осуществляется передача сообщений

-s, --size-of-message

Для режима отправки сообщений, можно указать размер сообщений, по умолчанию 100 байт.

-n, --sending-attempts-num

Для режима отправки сообщений, можно указать количество сообщений, по умолчанию 1000 сообщений отправляется.

### 9.3 Расчет времени задержки

#### Используемые инструменты для теста.

Для теста необходимо запустить сервер `nsl4-server`.

В качестве программы тестирования `nsl4_ping_test`.

Для просмотра результата тестирования NSL4 Monitor.

#### Условия и параметры.

1) Запускаем сервер, указываем порт 1280.

2) Запускаем программу `nsl4_ping_test`, для получения сообщений, со следующими параметрами.

```
./nsl4_ping_test -d IPV4 -p 1280 -e
```

3) Запускаем программу NSL4 Monitor, подключаемся и смотрим id клиента, который ожидает сообщения сообщений.

3) Запускаем программу `nsl4_ping_test`, для отправки сообщений, со следующими параметрами.

```
./nsl4_ping_test -d IPV4 -a 127.0.0.1 -p 1280 -c 7 -s 100 -n 1000
```

В данном примере клиенту с id=7 отправлено 1000 сообщений, размер сообщений 100 bytes, передача сообщений происходит локально. Ниже приведены результаты измерений задержек по сети и локально.

Задержки для 1000 сообщений по 100 байт:

	TCP, Макс. задержка, мкс	TCP, Ср. задержка, мкс
Local	8800	900
100Mbit eth	23000	1060