

Владивостокский государственный университет  
экономики и сервиса  
Институт информатики, инноваций и бизнес систем  
Кафедра информационных систем и компьютерных  
технологий

**Предмет:**  
**«Телекоммуникационные технологии»**

Руководитель: Сачко Максим Анатольевич, ст.  
преподаватель

# Тема 3

## Протокол ТСП

## Содержание:

1. Функции протокола ТСР.
2. Заголовок ТСР-сегмента.
3. Промежуточные состояния соединения.

---

# 1. Функции протокола ТСР:

- Базовая передача данных;
- Обеспечение достоверности;
- Разделение каналов;
- Управление соединениями;
- Управление потоком;

---

Протокол TCP (Transmission Control Protocol, Протокол контроля передачи) обеспечивает сквозную доставку данных между прикладными процессами, запущенными на узлах, взаимодействующих по сети. Стандартное описание TCP содержится в RFC-793.

---

## Базовая передача данных

Модуль TSP выполняет передачу непрерывных потоков данных между своими клиентами в обоих направлениях. Клиентами TSP являются прикладные процессы, вызывающие модуль TSP при необходимости получить или отправить данные процессу-клиенту на другом узле.

---

## **Обеспечение достоверности.**

Модуль ТСП обеспечивает защиту от повреждения, потери, дублирования и нарушения очередности получения данных.

---

## Разделение каналов.

Протокол TSP обеспечивает работу одновременно нескольких соединений. Заголовок TSP-сегмента содержит номера портов процесса-отправителя и процесса-получателя. При получении сегмента модуль TSP анализирует номер порта получателя и отправляет данные соответствующему прикладному процессу



---

## Управление соединениями.

Соединение - это совокупность информации о состоянии потока данных, включающая сокет, номера посланных, принятых и подтвержденных октетов, размеры окон.

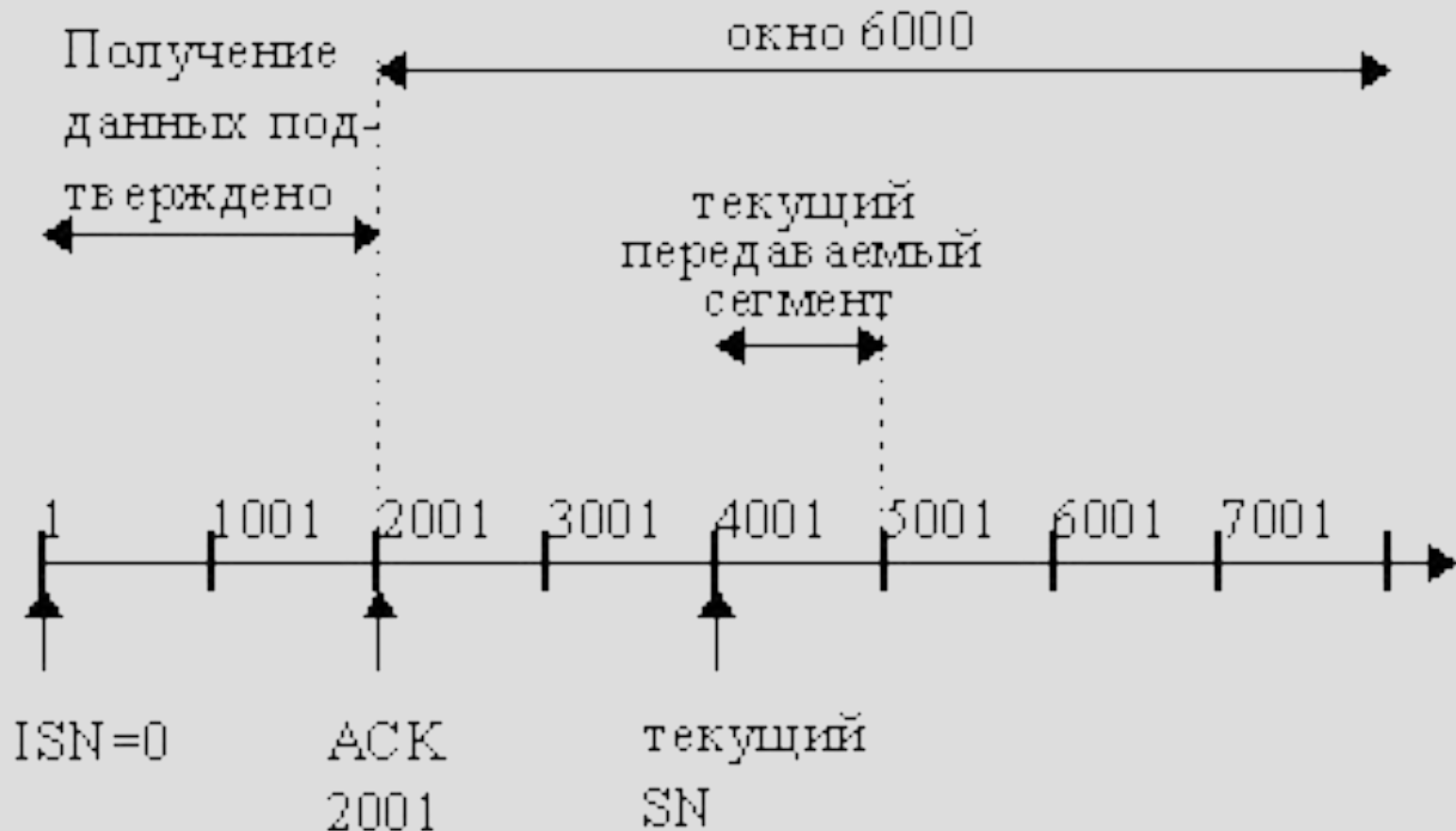
Каждое соединение уникально идентифицируется в Интернет парой сокетов.

# Установка TCP-соединения



# Управление потоком.

## Метод скользящего окна



## 2. Заголовок ТСР-сегмента.

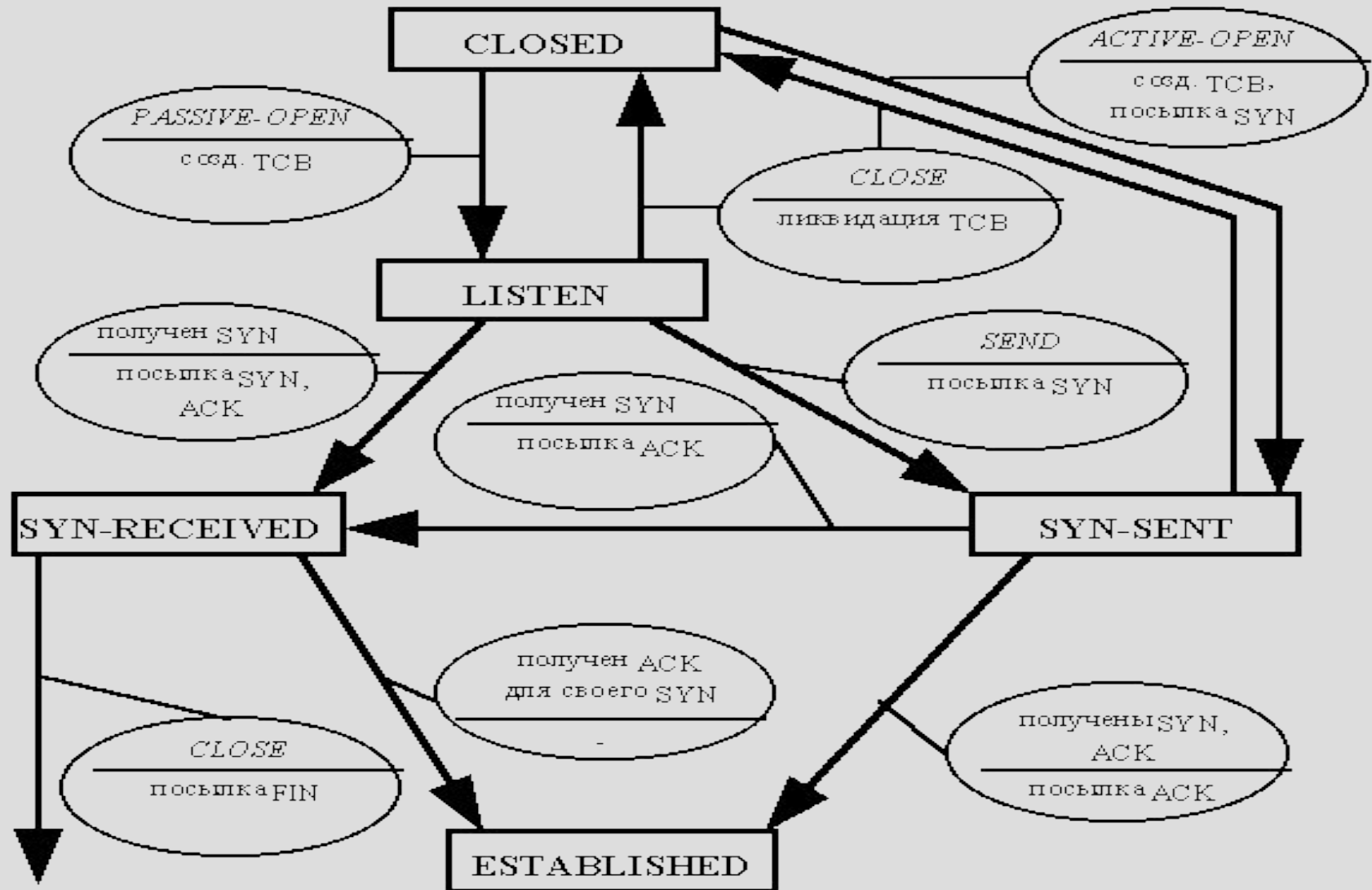
ТСР-сегмент состоит из заголовка и данных.

|                             |                |   |   |    |   |                  |   |        |  |         |  |    |  |  |  |
|-----------------------------|----------------|---|---|----|---|------------------|---|--------|--|---------|--|----|--|--|--|
| 0                           |                | 7 |   | 15 |   |                  |   | 23     |  |         |  | 31 |  |  |  |
| Source Port                 |                |   |   |    |   | Destination Port |   |        |  |         |  |    |  |  |  |
| Sequence Number (SN)        |                |   |   |    |   |                  |   |        |  |         |  |    |  |  |  |
| Acknowledgment Number (ACK) |                |   |   |    |   |                  |   |        |  |         |  |    |  |  |  |
| Data Offset (0-3)           | reserved (4-9) | U | A | P  | R | S                | F | Window |  |         |  |    |  |  |  |
|                             |                | R | C | S  | S | Y                | I |        |  |         |  |    |  |  |  |
|                             |                | G | K | H  | T | N                | N |        |  |         |  |    |  |  |  |
| Checksum                    |                |   |   |    |   | Urgent Pointer   |   |        |  |         |  |    |  |  |  |
| Options                     |                |   |   |    |   |                  |   |        |  | Padding |  |    |  |  |  |

---

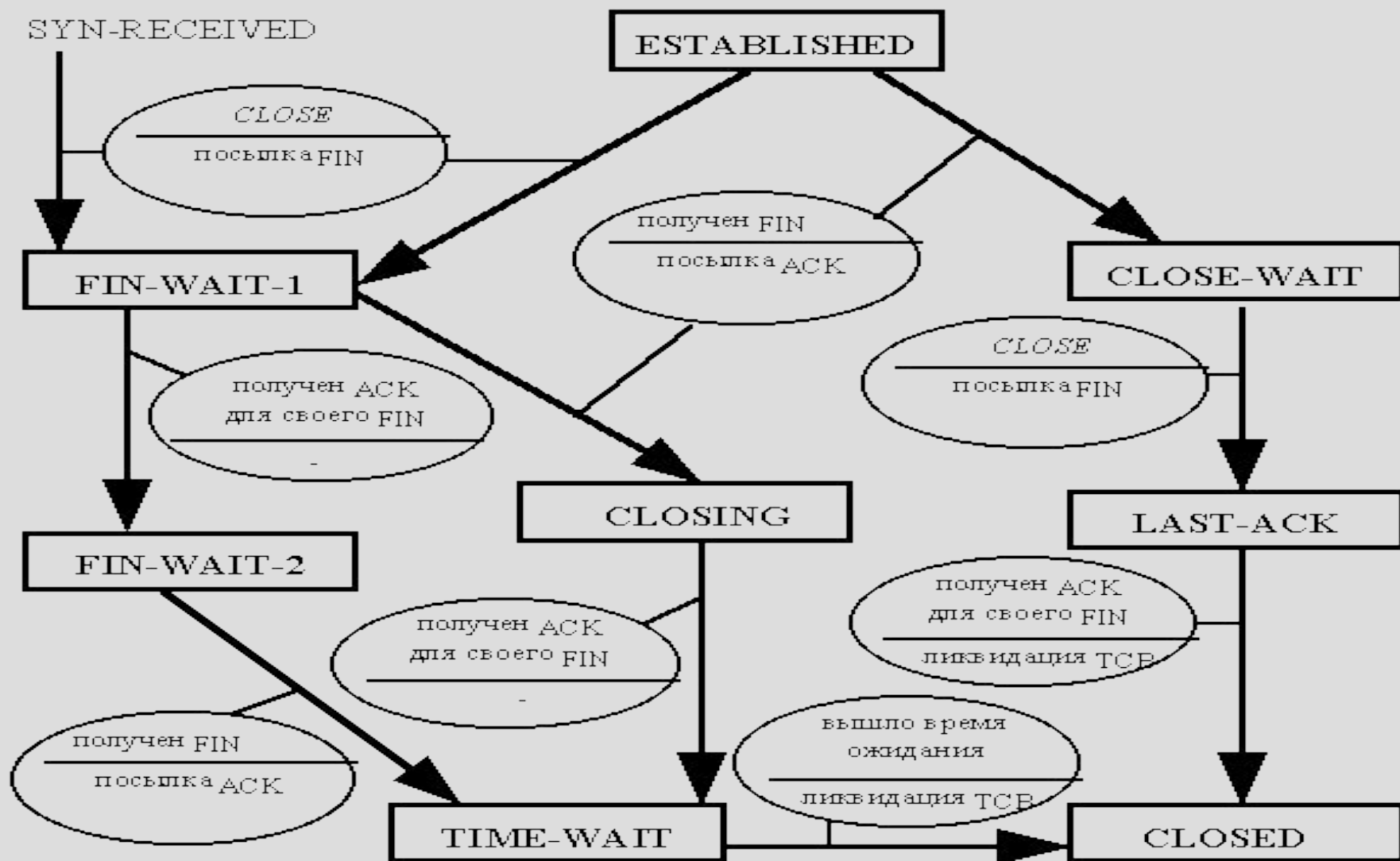
Промежуточные состояния соединения TCP-соединение во время функционирования проходит через ряд промежуточных состояний. Это состояния LISTEN, SYN-SENT, SYN-RECEIVED, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME-WAIT, а также фиктивное состояние CLOSED. (Состояние CLOSED является фиктивным, поскольку оно представляет отсутствие соединения.)

# Фаза установления соединения



FIN-WAIT-1

# Фаза закриття соединения



---

Определены следующие запросы процесса-клиента модулю ТСР (с каждым запросом, кроме OPEN, передается имя соединения):

ACTIVE-OPEN - активное открытие соединения;

PASSIVE-OPEN - пассивное открытие соединения



---

**SEND** - отправка данных (передается указатель на буфер данных, размер буфера, значения флагов URG и PSH);

**RECEIVE** - получение данных (передается указатель на буфер данных, размер буфера; возвращается счетчик полученных октетов, значения флагов URG и PSH);

**STATUS** - запрос состояния соединения;

---

CLOSE - закрытие соединения  
(производится досылка всех  
неотправленных данных и обмен  
сегментами с битом FIN);

ABORT - ликвидация соединения  
(уничтожаются блок TCB и все  
неотправленные данные, посылается  
сегмент с битом RST).

---

## Состояния соединения.

**LISTEN** - процесс пассивно ждет запроса со стороны чужих сокетов.

**SYN-SENT** - процесс отправил свой SYN и ждет чужого SYN.

**SYN-RECEIVED** - процесс получил чужой SYN, отправил (раньше или только что) свой SYN и ждет ACK на свой SYN.

---

ESTABLISHED - процесс отправил ACK на чужой SYN, получил ACK на свой SYN; соединение установлено.

FIN-WAIT-1 - процесс первый отправил свой FIN и ждет реакцию той стороны; при этом он, возможно, продолжает получать данные.

---

ESTABLISHED - процесс отправил ACK на чужой SYN, получил ACK на свой SYN; соединение установлено.

FIN-WAIT-1 - процесс первый отправил свой FIN и ждет реакцию той стороны; при этом он, возможно, продолжает получать данные.

---

FIN-WAIT-2 - процесс получил ACK на свой ранее отправленный FIN, но не получил чужой FIN; ждет чужой FIN; при этом, возможно, продолжает получать данные.

CLOSE-WAIT - процесс, не отправив свой FIN (возможно, не собираясь прекращать соединение), получает чужой FIN; он отправляет ACK на чужой FIN, но при этом, возможно, продолжает отправлять данные.

LAST-ACK - процесс отправил свой FIN, но ранее он уже получил FIN с той стороны и отправил на него ACK; поэтому процесс ожидает чужой ACK на свой FIN для окончательного закрытия соединения.

CLOSING - процесс ранее отправил свой FIN и еще не получил от него подтверждение, но получил чужой FIN (и отправил на него ACK); ждет ACK на свой FIN.

TIME-WAIT - процесс ранее отправил свой FIN и получил на него подтверждение, получил чужой FIN и только что отправил на него ACK; теперь процесс ждет некоторое время (два времени жизни сегмента, обычно 4 минуты) для гарантии того, что та сторона получит его ACK на свой FIN, после чего соединение будет окончательно закрыто.

CLOSED - соединение отсутствует.



# Вопросы для самопроверки:

1. Каковы недостатки протокола ТСР? Подходы к их решению.
2. Как приложение взаимодействует со стеком ТСР/IP?
3. На каком уровне стека ТСР/IP находится протокол ТСР?
4. В чем состоит принцип асинхронной и синхронной передачи данных?
5. Чем идентифицируется прикладной процесс в протоколе ТСР?
6. Что такое метод скользящего окна?

# Рекомендуемая литература:

1. Мамаев М.А. Телекоммуникационные технологии (Сети TCP/IP). – Владивосток: Изд-во ВГУЭС, 2004.
2. Леинванд А., Пински Б. Конфигурирование маршрутизаторов Cisco. 3-е издание. – М.: "Вильямс", 2007.
3. Мамаев М., Петренко С. Технологии защиты информации в Интернете. Специальный справочник. – СПб: "Питер", 2005.
4. Doyle J. "Routing TCP/IP. Volume I" – Cisco Press, 2007.

- **Использование материалов презентации**

- Использование данной презентации, может осуществляться только при условии соблюдения требований законов РФ об авторском праве и интеллектуальной собственности, а также с учетом требований настоящего Заявления.
- Презентация является собственностью авторов. Разрешается распечатывать копию любой части презентации для личного некоммерческого использования, однако не допускается распечатывать какую-либо часть презентации с любой иной целью или по каким-либо причинам вносить изменения в любую часть презентации. Использование любой части презентации в другом произведении, как в печатной, электронной, так и иной форме, а также использование любой части презентации в другой презентации посредством ссылки или иным образом допускается только после получения письменного согласия авторов.