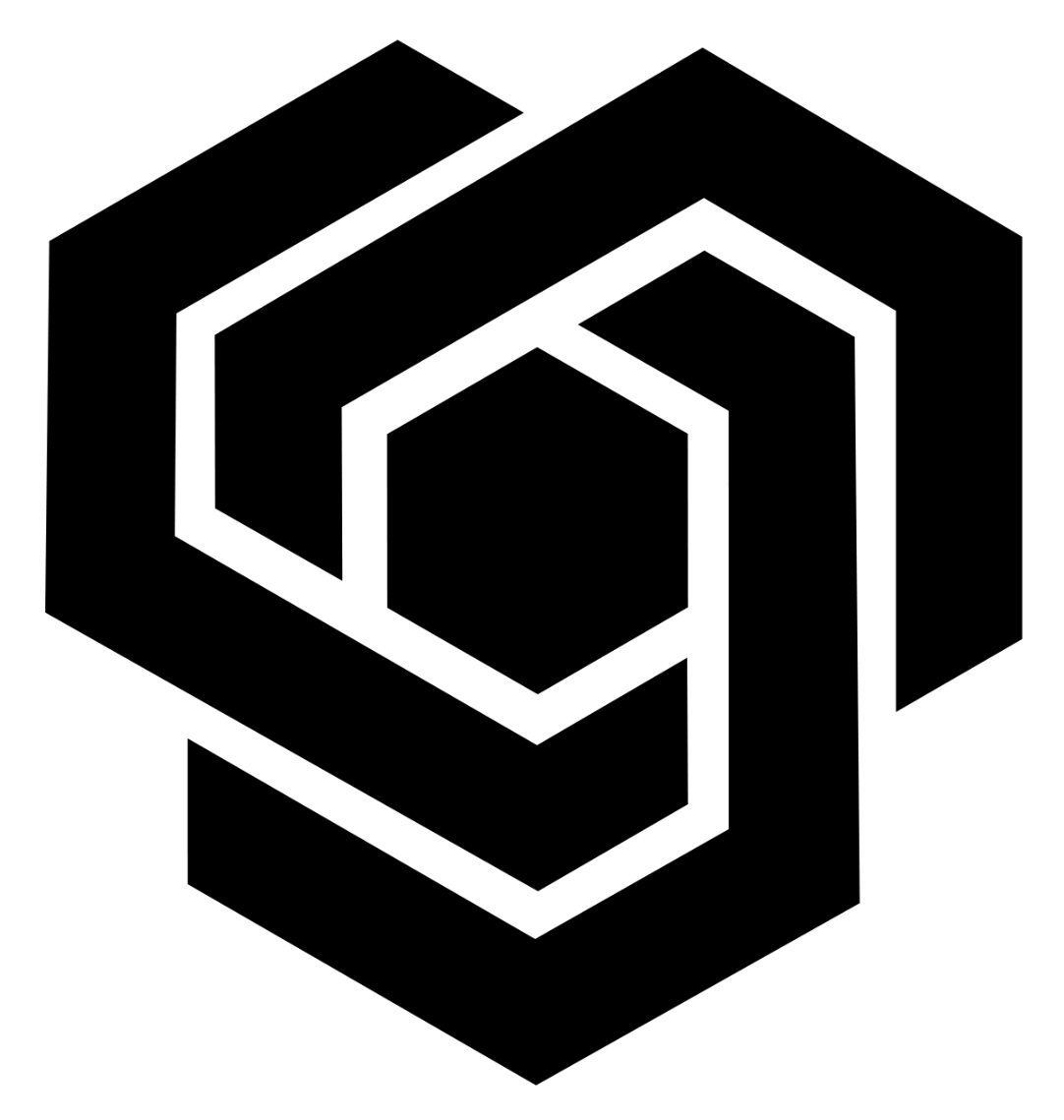
**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

Факултет по телекомуникации



**Курсов проект**

Предмет: Предаване на данни и компютърни комуникации

**Процедура за предаване на данни в каналния слой с отстраняване на грешките**

Изготвил: Николай Станимиров Проданов

Фак. № 111217089

Група: 52

Специалност: Телекомуникации

Дата:.................... Ръководител:......................  
 /ас. Димитър Атамян/

# **1. Задачи:**

1. Да се разучи протокол **HDLC (LAPB)** за предаване на данни в каналния слой с възможност за отстраняване на грешките чрез повторение на данните и непрекъсната работа.
2. Да се състави времедиаграма, показваща последователността от рамки, които се обменят между предавателя и приемника, в съответствие с конкретните изходни данни от заданието. Във време диаграмата да се покаже отстраняването на една грешка в предаваната информация, която да не е в първия или последния пакет/рамка.
3. Да се определи минималната ширина на прозореца (w), необходима за постигане на непрекъснато предаване в канал без грешки.
4. Да се изчисли ефективността на използване на канала за данни и средната скорост на предаване за конкретния пример, показан във времедиаграмата от т.2.

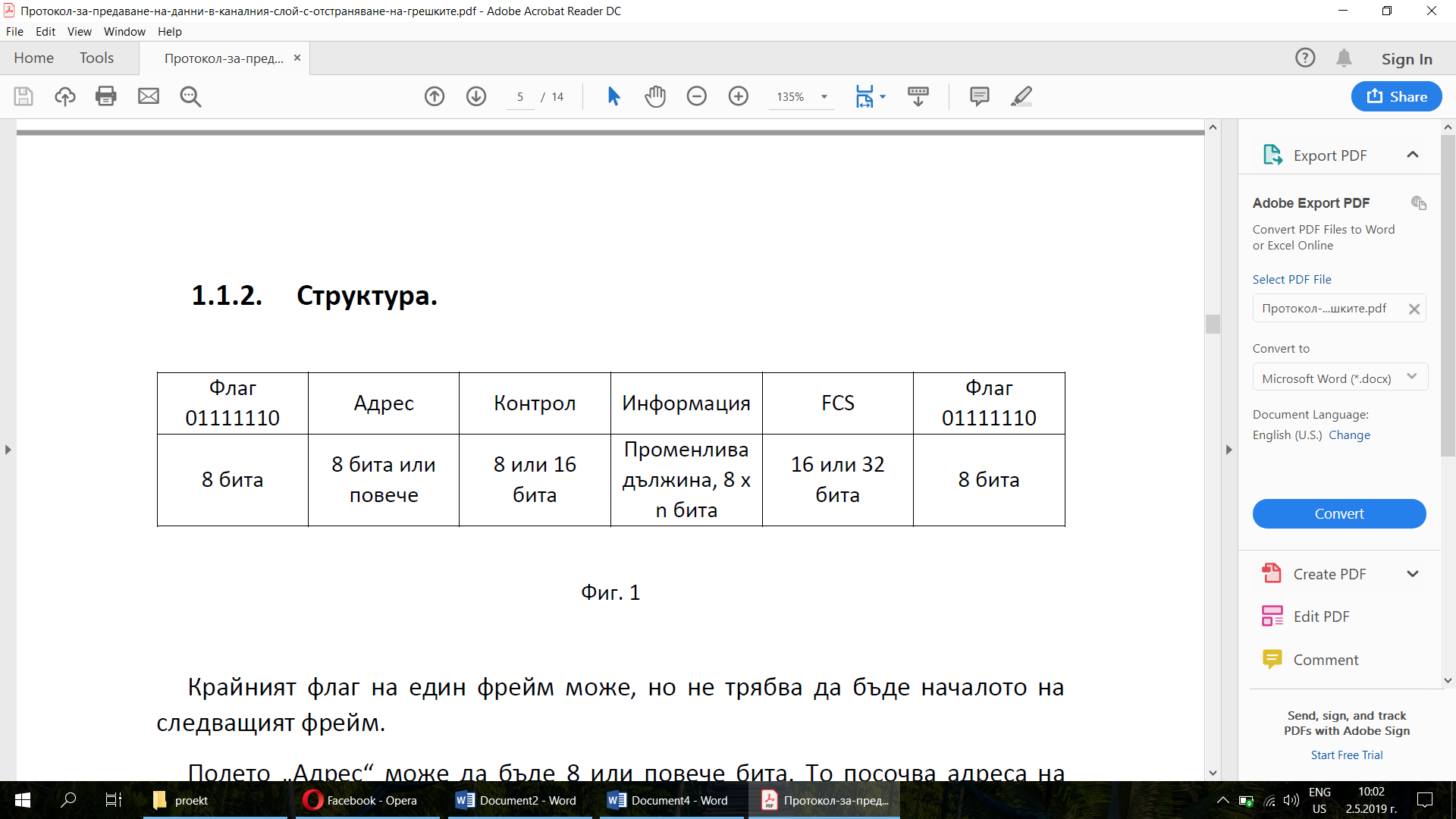
# 2. **Теоретична част:**

## **1.1. HDLC.**

HDLC е битово ориентиран кодово прозрачен протокол за синхронна връзка с данни, разработен от международната организация по стандартизация (ISO). Той осигурява както услуга свързана с връзка, така и без връзка. Може да се използва за свързване от точка до много точки посредством оригиналните режими на нормален отговор (NRM) и асинхронния режим на отговор (ARM), но те вече се използват рядко. В момента за свързване на едно устройство с друго почти винаги се използва асинхронно балансиран режим (ABM).

HDLC фреймовете могат да се предават чрез синхронни или асинхронни серийни комуникационни връзки. Те нямат механизъм, който да маркира началото или края на даден фрейм, затова той трябва да се идентифицира по някакъв начин. Това се постига като се използва уникална последователност от битове като разделител на фреймове или флаг и кодиране на данните, за да се гарантира, че последователността на фрейма никога не се вижда. Всеки фрейм започва и завършва с флаг. Той може също да маркира началото на следващия фрейм. При синхронни и асинхронни връзки последователността на флага е двоично „01111110“ или шестнадесетичен 0x7E, но данните са съвсем различни.

Съдържанието на HDLC рамка е показано в следната таблица:



Крайният флаг на един фрейм може, но не трябва да бъде началото на следващият фрейм.

Полето „Адрес“ може да бъде 8 или повече бита. То посочва адреса на назначението, в случай че фрейма съдържа команда или адреса на източника ако фрейма съдържа отговора.

Полето „Контрол“ е 8 или 16 бита и служи за да разбере какъв тип е фреймът. В допълнение може да съдържа sequence number и други услуги за контрол в зависимост от типа на фрейма.

Фреймовата проверяваща последователност (FCS) е 16 или 32 битова и е изчиследна в полетата адрес, контрол и информация. Чрез нея приемникът може да открие грешки, които може да са били предизвикани по времето на предаването на фрейма.

Служебните битове се взимат от различните слотове и могат да бъдат различни на брой.

## **1.2. Sliding window**

Това е пакетно ориентиран протокол за предаване на данни. Използва се когато е важно последователното и надеждно доставяне на пакетите. Предавателят и приемникът имат текущ пореден номер 𝑛𝑡 и 𝑛𝑟 на изпратеният и съответно полученият пакет, както и ширина на прозореца 𝑤𝑡 и 𝑤𝑟. Чрез този номер приемникът подрежда в правилната последователност пакетите и следи за дубликати или липсващи такива. Под „прозорец“ предавателят разбира броят пакети, които са изпратени и трябва да бъдат потвърдени от приемника.

Когато предавателят трябва да изпраща данни, той може да прати до 𝑤𝑡 пакета напред спрямо последното потвърждение 𝑛А. Ако няма проблем с комуникацията, приемникът скоро получава удостоверение, че те са пристигнали и 𝑛А става равно на 𝑛𝑡. Ако потвърждението не бъде получено, пакетите между 𝑛А и 𝑛𝑡 се изпращат пак. Получателят задължително трябва да потвърди пратените отново данни за да не бъдат изпращани многократно.

В режим „бавен старт“ предавателят започва да изпраща пакети и постепенно увеличава броят им след всяко потвърждение от приемника. За всеки потвърден пакет прозорецът се „плъзга“ напред с един пакет за да може да предаде новия. След достигане на максималният размер на прозореца се изпраща само по един пакет за всеки получен и потвърден.

При всяко получаване на пакет 𝑥 предавателят проверява дали той спада в прозореца за получаване 𝑛𝑟≤𝑥<𝑛𝑟+𝑤𝑟. Ако не попада в границите, той бива отхвърлен и променливите на приемника се изменят. Ако влиза в границите 𝑛𝑟 и 𝑤𝑟 се обновяват. Независимо дали пакетът е приет или не бива изпратено потвърждение, съдържащо текущата стойност на 𝑛𝑟.

Sliding window методът предотвратява претоварвания по мрежата. Примери за протоколи от този тип са stop-and-wait и Go-back-N, разгледан по-долу.

## **1.3. LAPB – Link Access Procedure, Balanced**

Това е протокол, произлизащ от HDLC гарантира, че фреймовете са без грешки и са в правилната последователност. Той се използва за управление на комуникацията и изпращането на пакетите между терминалното и комуникационното устройство в стандарта X.25. По съществото си LAPB е HDLC в асинхронно балансиран режим (ABM). Неговите сесии могат да бъдат установени от DTE или DCE. Станцията, която инициира повикването се счита за основна, а тази която отговаря на него е второстепенна.

Съдържанието на LAPB рамка:

Флагът винаги е със стойност 0x7E(01111110). За да се осигури, че последователността от нитовете на флага няма да се срещне в полето за данните , се използва Bit stuffing.

Полето за адреса не е толкова важно защото протоколът се използва при свързване от точка до точка. То се използва с друга цел – показва дали рамката съдържа команда или отговор.

Полето за контрол служи за идентификация на вида на рамката. Включва също така и поредния номер.

Контролното поле позволява високо ниво на контрол на грешките, като позволява проверка на целостта на данните от предаваната рамка.

Има няколко типа фрейма:

- I-Frames (Information frames) – Съдържат информация от горния слой и някаква контролна информация. Функциите им съдържат създаване на последователности, контрол на потока и откриване и възстановяване на грешки. I-рамкте пренасят и получават номера на поредици.

- S-Frames (Supervisory frames) – Те носят контролна информация. Функциите на S-рамката включват искане и спиране на предаването, отчитане на състоянието и потвърждение за получаването на I-рамки. S-фреймовете носят само номерата на поредицата за получаване.

- U-Frames (Unnumbered frames) – Носят контролна информация. Техните функции включват настройка на връзката и изключване, както и докладване за грешки. U-рамките не носят последователни номера.

LAPB няма взаимоотношения между главен и подчинен възел. Подателят използва Poll бита в командните фреймове за да настоява за незабавен отговор. Във фреймовете на отговора същият този бит се превръща в приемник на крайният бит. Приемникът винаги включва последният бит в отговора си в отговор на командата от подателя. P/F бита обикновено се използва, когато и двата края не са сигурни за правилната последователност на фрейма поради евентуално липсващо потвърждение и е необходимо да се възстанови референтната точка.

## **1.4. Go-Back-N ARQ.**

Този протокол е частен случай на ARQ и Sliding window протоколите и има ширина на прозореца за изпращане N и получаване – 1. Може да изпрати N на брой рамки преди да изисква потвърждение (ACK).

Получателят следи за поредният номер на следващата рамка, която очаква да получи и го изпраща с всяко ACK. Всяка рамка с грешен пореден номер ще бъде отхвърлена, а за всяка правилно получена се изпраща ACK. След като предавателят изпрати всички рамки в неговият прозорец, той ще отчете всички рамки след грешно изпратената като чакащи потвърждение, ще се върне към поредният номер на последното ACK, което е получил и ще започне новият прозорец от него.

# **3. Изчислителна част:**

## **Задача 2.** Да се състави времедиаграма, показваща последователността от рамки, които се обменят между предавателя и приемника, в състояние с конкретните изходни данни от заданието. Във времедиаграмата да се покаже отстраняването на една грешка в предаваната информация, която да не е в първия или последния пакет.

**Изходни данни:**

*скорост на предаване:* ***19200 bps;*** *обем на предаваната информация:* ***13500 B;*** *максимален размер на рамката:* ***1500 B;*** *закъснение във физическия слой* ***1,2 s;***

Отстраняване на грешките: Go-back-N.

-Полезна информация:

1500 𝐵 − 6 𝐵 (Служебна информация)=1494 𝐵

*- Рамки:*

*=9 рамки с макцимален размер и 1 по−малка (10 рамки)*

Максималният размер на рамката е 1500B като от тях 6B са служебна информация => остават максимум 1494 𝐵 за полезната информация. От това следва че ще бъдат предадени 10 рамки. За последната рамка смятаме:

*Добавяме и служебната информация и получаваме:*

***Време за предаване на рамка 0:*** *t0 = (1494\*8)/19200 = 0,6225s t'0 = t0 + td = 0,6225s+1,2s =1,8225s*

*Връща се потвърждение (ACK) и времето е:*

*1,8225s+1,2s+6\*8/19200=1,8225s+1,2s+0,0025s=3,025s*

***Време за предаване на рамка 1:*** *t1 =* *0,6225s+0,6225s=1,245s t'1 = t1 + td = 1,245s+1,2s=2,445s  
Връща ACK, което е 2,445s+1,2s+0,0025s=3,6475s*

***Време за предаване на рамка 2:***

*t2 =1,245s+0,6225s=1,8675s t'2 = t2 + td = 1,8675s +1,2s=3,0675s*

*Връща ACK, което е 3,0675s+1,2s+0,0025s=4,27s*

***Време за предаване на рамка 3:***

*t3 =1,8675s+0,6225s=2,49s t'3 = t3 + td =2,49s+1,2s=3,69s.*

*Връща ACK, което е 3,69s+1,2s+0,0025s=4,8925s*

***Време за предаване на рамка 4:***

*t4 =2,49s+0,6225s=3,1125s t'4 = t4 + td = 3,1125s+1,2s=4,3125s.*

*Връща NACK, което е 4,3125s+1,2s+0,0025s=5,515s.*

*е правилно предадена, има грешка.*

***Време за предаване на рамка 5:***

*t5 =3,1125s+0,6225s=3,735s t'5 = t5 + td = 3,735s+1,2s=4,935s.*

*Връща ACK, което е 4,935s+1,2s+0,0025s=6,1375s. X*

***Време за предаване на рамка 6:*** *t6 =3,735s+0,6225s=4,3575s t'6 = t6 + td =4,3575s+1,2s=5,5575s.  
Връща ACK, което е 5,5575s+1,2s+0,0025s=6.76s. X*

***Време за предаване на рамка 7:*** *t7 = 4,3575s+0,6225s=4,98s t'7 = t7 + td =4,98s+1,2s=6,18s.  
Връща ACK, което е 6,18s+1,2s+0,0025=7,3825s. X*

***Време за предаване на рамка 8:*** *t8 = 4,98s+0,6225s=5,60s t'8 = t8 + td =5,60s+1,2s=6,8s.  
Връща ACK, което е 6,8s+1,2s+0,0025=8,0025s. X*

***Време за предаване на рамка 4:*** *t9 =5,60s+0,6225s=6,225s t'9 = t9 + td =6,225s+1,2s =7,425s.  
Връща ACK, което е 7,425s+1,2s+0,0025=8,6275s.*

***Време за предаване на рамка 5:*** *t10 =6,225s+0,6225s=6,8475s t'10 = t10 + td =6,8475s+1,2s=8,0475s.  
Връща ACK, което е 8,0475s+1,2s+0,0025=9.25s.*

***Време за предаване на рамка 6:***

*t11 =6,8475s+0,6225s=7,47s t'11 = t11 + td =7,47s+1,2s=8,67s.  
Връща ACK, което е 8,67s+1,2s+0,0025=9,8725s.*

***Време за предаване на рамка 7:***

*t12 =7,47s+0,6225s=8,0925s t'12 = t12 + td =8,0925s+1,2s=9,2925s.  
Връща ACK, което е 9,2925s+1,2s+0,0025=10,495s.*

***Време за предаване на рамка 8:***

*t13 =8,0925s+0,6225s=8,715s t'13 = t13 + td =8,715s+1,2s=9,915s.  
Връща ACK, което е 9,915s+1,2s+0,0025=11,1175s.*

***Време за предаване на рамка 9:***

*t14 =8,715s+(60\*8)/19200=8,74s t'14 = t14 + td = 11,0125s+1,2s=9,94s.  
Връща ACK, което е 9,94s+1,2s+0,0025=11,1425s.*

***Време за предаване на информацията:*** *8,74s*

***Общо време за предаване:*** *11,1425s*

