Методи передачі даних канального рівня

У протоколів канального рівня є дві сфери дії: перша - доставлення кадрів даних у межах однієї локальній мережі, а друга - зв'язок типу "точка-точка" у глобальних мережах.

Основними характеристиками протоколів канального рівня є наступні:

- асинхронний або синхронний

- символьно-орієнтований або біт-орієнтований

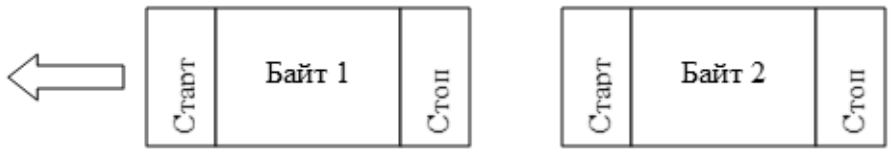
- із встановленням з'єднання або дейтатаграмний

- із виявленням викривлених даних або без виявлення

- з підтримкою динамічної компресії даних або без підтримки

Асинхронні протоколи

Асинхронні протоколи - це найбільш старий спосіб зв'язку. Використовуються для зв'язку телетайпів, клавіатур та дисплеїв з комп'ютерами. Одиницею передаваних даних є не кадр даних, а окремий символ. Кожен байт даних супроводжується спеціальними сигналами - "старт" і "стоп".



Сигнал "старт" необхідний, щоб сповістити приймач про прихід даних, а "стоп" служить для того, щоб дати час приймача для підготовки до прийому наступного байта.

Асинхронним даний режим роботи називається тому, що кожен байт може бути зміщений у часі щодо побітових тактів попереднього байта. Така асинхронність не вплине на коректність прийнятих даних, за рахунок застосування сигналів "старт".

Причинами використання асинхронних протоколів є низька якість лінії зв'язку і наявність пристроїв, що генерують байти даних в випадкові моменти часу, наприклад, клавіатури.

Поступово асинхронні протоколи ускладнювалися і стали поряд з окремими символами використовувати цілі блоки даних, наприклад протокол XMODEM.

Синхронні символьно - орієнтовані і біт - орієнтовані протоколи

У синхронних протоколах всі обміни даними здійснюються кадрами, які в загальному випадку мають заголовок, поле даних і кінцевик. Всі біти кадру передаються безперервним синхронним потоком, що значно прискорює передачу даних.

Символьно-орієнтовані протоколи використовуються в основному для передачі блоків відображуваних символів, наприклад текстових файлів. Синхронізація досягається за рахунок того, що передавач додає один або два керуючих символу SYN, перед кожним блоком символів. Після цього зазвичай додається символ границі початку кадру STX (Start of TeXt). Інший символ відзначає закінчення кадру ETX (End of TeXt).

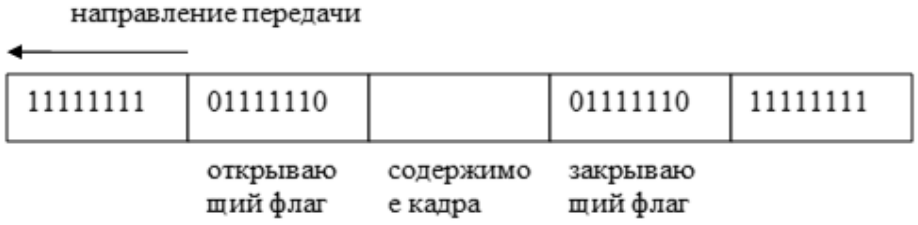
У разі передачі двійкових даних усередині кадру можуть зустрітися символи STX або ETX, що призведе до помилкового прийому кадру. Тому перед керуючими символами всередині кадру передавач вставляє символ DLE (Data Link Escape) (наприклад в протоколі BSC компанії IBM). Така процедура називається стаффінг символів або байт-стаффінг.

Недоліками символьно-орієнтованих протоколів є залежність від кодування і велика надмірність даних через байт-стаффинг. Тому в даний час для передачі як символьних так і двійкових даних застосовується біт-орієнтована передача.

Біт-орієнтовані протоколи відрізняються один від одного різним способом позначення початку і кінця кадру.

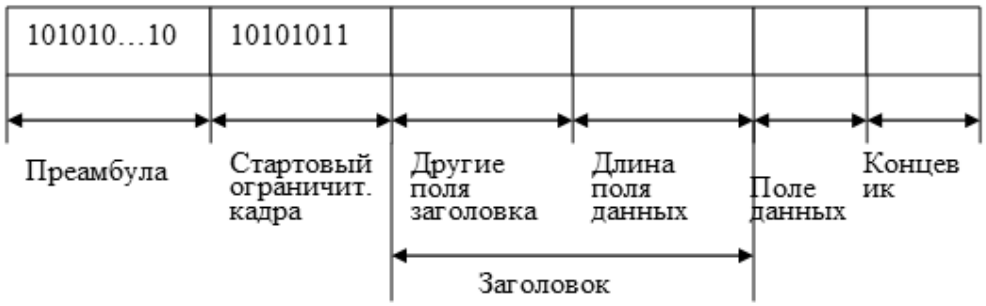
У першому способі, початок і кінець кадру позначається однією і тією ж 8-бітової послідовністю - 01111110, так званим прапором. Щоб забезпечити синхронізацію приймача, передавач посилає послідовність байтів простою, що складаються одиниць і передує стартовому прапору. Для досягнення прозорості даних в цьому способі необхідно, щоб прапор не був присутнім в поле даних кадру. Це досягається за допомогою прийому, відомого як вставка нульового біта або біт-стаффінг. Якщо в поле даних передавач виявляє п'ять одиниць поспіль, то він автоматично вставляє додатковий 0. У приймачі діє зворотна схема - якщо після п'яти 1 виявляється 0, то він видаляється з поля даних. [Біт-стаффінг економічніший ніж байт-стаффінг, так як замість зайвого байта вставляється один біт].

Біт-стаффінг використовується у протоколі HDLC (High-Level Data Link Control) від IBM, а також у USB.



У другому способі для позначення початку кадру використовується тільки початковий прапор, а для визначення кінця кадру використовується поле довжини кадру або довжини поля даних кадру. У таких мережах для позначення факту незайнятості середовища взагалі не передається ніяких сигналів. Щоб всі інші станції увійшли в побітову синхронізацію, передавач спочатку посилає послідовність нулів і одиниць 101010 ... відому як преамбула. Приймач досліджуючи преамбулу, знаходить байт початку кадру 10101011, який грає роль символу STX. Далі йде заголовок кадру, в якому є поле довжини кадру

Даний спосіб використовується у протоколі Ethernet.



Третій спосіб використовує для позначення початку і кінця кадру прапори, які включають заборонені для даного коду символи. Наприклад, при манчестерському кодуванні замість зміни потенціалу в середині тактового інтервалу рівень сигналу залишається незмінним і низьким - заборонений сигнал J, або незмінним і високим - заборонений сигнал K. Початок кадру відзначається послідовністю JK0JK000, а кінець JK1JK100. Даний спосіб дуже економічний, так як не вимагає ні біт-стаффинга, ні поля довжини, але залежить від методу фізичного кодування. При використанні надлишкових кодів роль сигналів J і K грають заборонені символи.

Даний спосіб використовується у протоколі Token Ring.

Передача із встановленням з'єднання і без встановлення з'єднання

При передачі без встановлення з'єднання, яка ще називається дейтаграмною, кадр передається до мережі без попередження. Протокол не несе відповідальності за втрату кадру. Дейтаграмний метод працює швидко, але він не гарантує доставку пакету.

Передача із встановленням з'єднання більш надійна, але потребує більше часу для передачі даних і затрат на обчислення від кінцевих вузлів.

У цьому випадку встановлення з'єднання називається т. н. "триразовим рукостисканням". На першому етапі вузол – ініціатор відправляє вузлу-приймачу службовий кадр спеціального формата з пропозицією установити з'єднання. Якщо вузол-приймач згоден із цим, то він посилає у відповідь інший службовий кадр, який підтверджує встановлення з'єднання і пропонує деякі параметри, наприклад, ідентифікатор з'єднання і т. п. На третьому етапі вузол-ініціатор з'єднання відправляє третій службовий кадр, в якому повідомляє, що запропоновані параметри йому підходять.

Після передачі всіх даних вузол ініціює розрив логічного з'єднання, а інший вузол повинен підтвердити розрив цього з'єднання.

Виявлення і корекція помилок.

Більша частина протоколів канального рівня тільки виявляють помилки, перекладаючи роботу з корекції помилок на протоколи верхніх рівнів. Так діють протоколи Ethernet, Token Ring, FDDI та інші. Тим не менше існують і протоколи канального рівня, які самостійно відновлюють втрачені і помилкові кадри, наприклад, LLC2 або LAP-2.

[Це залежить від умов, в яких працюють протоколи. Якщо мережа надійна і помилки при передачі відбуваються рідко то немає сенсу коригувати їх на канальному рівні. І навпаки, якщо помилки часті, то вигідніше коригувати їх на канальному рівні (як в глобальній мережі першого покоління X.25). Хоча протокол верхнього рівня і відновить пакети, це буде зроблено з великими тайм-аутами і затримками.].

Методи виявлення помилок.

Контроль по паритету найпростіший і найменш надійний алгоритм контролю даних. Метод полягає у визначенні суми по модулю 2 всіх біт контрольованої інформації. Наприклад, для даних 100101011 результатом суми за модулем 2 буде 1. Результат передається разом із даними. Якщо при передачі зміниться значення одного біту даних, результат суми буде

відрізнятися від прийнятого контрольного розряду, що говорить про помилку. Але подвійна помилка, наприклад, 110101010 буде прийнята за коректні дані. Тому даний метод застосовується для невеликих порцій даних, як правило до байтів, що дає коефіцієнт надлишковості 1/8. Через це він рідко застосовується в інформаційних мережах.

Вертикальний і горизонтальний контроль по паритету розглядає дані у вигляді матриці, рядки якої складають байти даних. Контрольний розряд підраховується окремо для кожного рядка і кожного стовпця матриці. Цей метод виявляє більшість подвійних помилок, але має велику надлишковість і тому на практиці майже не використовується.

Циклічний надлишковий контроль (Cyclic Redundency Check, CRC) зараз є найпопулярнішим. Метод заснований на розгляді вихідних даних як одного багато розрядного двійкового числа. Контрольною сумою є залишок від ділення цього числа на деякий відомий дільник R (ділити числа у двійковому вигляді так само як і в десятковому https://msn.khnu.km.ua/pluginfile.php/147599/mod_resource/content/3/Lekz12_13.pdf Приклад 12.23.). Зазвичай число R є 17 або 33 розрядним числом, щоб залишок від ділення мав довжину 16 або 32 розрядів. При отриманні кадру контрольна сума розширюється до довжини даних і знаходиться її доповнення (доповнення це еквівалент від'ємного числа в комп'ютерах. Як знаходити доповнення - див. примітку). Далі знаходиться сума вихідних даних і доповнення контрольної суми (фактично від даних віднімається залишок від ділення даних на R), а потім результат ділиться на R. Якщо залишок від цього ділення дорівнює нулю, то кадр передано без помилок.

Примітка. Доповнення числа у двійковому вигляді знаходиться в два етапи. Спочатку число інвертується, тобто замість нулів записуються одиниці і навпаки. Потім до нього додається один. Приклад. Дано 8-бітне число 10000111. Інвертування: Not10000111=01111000. Додається 1 : 01111000 +1 = 01111001 - це і є доповнення числа 10000111.

Приклад CRC. Нехай вихідні дані довжиною 8 біт - 10101010, дільник R=1011. Тоді контрольна сума дорівнює 10101010Mod1011=101. Після прийому кадру знаходимо доповнення контрольної суми Not00000101+1=11111010+1=11111011. Сума вихідних даних і доповнення контрольної суми 10101010+11111011= 10100101. Знаходимо контрольну суму даного числа: 10100101Mod1011=0 – значить дані передані без помилок.

Метод CRC виявляє одиночні та подвійні помилки і має невисоку ступінь надлишковості. Наприклад для кадра Ethernet довжиною 1024 байта контрольна інформація довжиною 4 байта складає тільки 0,4%.

Методи відновлення помилкових і втрачених кадрів.

Методи корекції помилок у комп’ютерних мережах засновано на повторній передачі кадру у випадку його втрати або помилки при передачі. Відправник нумерує кадри і для кожного кадру чекає від приймача на позитивну квитанцію. Це службовий кадр, який сповіщає, що вихідний кадр дійшов до приймача і дані в ньому коректні. Якщо за певний час позитивна квитанція не отримана, кадр вважається втраченим. Приймач у випадку отримання кадру з невірною інформацією може відправити негативну квитанцію.

Існують два підходи до організації процесу обміну квитанціями – з простоями і з організацією ковзного (скользящего) вікна (sliding window).

Метод з простоями (Idle Source) вимагає, щоб відправник чекав на позитивну або негативну квитанцію і тільки після її отримання відсилав кадр. Через затримки у відправці кадрів продуктивність методу низька.

В методі "ковзного вікна" передатчику дозволяється передавати деяку кількість кадрів у безперервному режимі без отримання квитанцій. Ця кількість кадрів називається розміром вікна.

Ефективність методу "ковзного вікна" дуже залежить від розміру вікна і величини тайм-ауту чекання на квитанцію. В надійних мережах для підвищення швидкості обміну даними розмір вікна треба треба збільшувати. Навпаки, у ненадійних мережах розмір вікна треба зменшувати, так як при частих помилках різко збільшується об’єм кадрів які повторно передаються. Пропускна здатність мережі при цьому буде падати.

Метод "ковзного вікна" реалізований у багатьох протоколах LLC2, LAP-B, X.25, TCP.

Компресія даних

Компресія (стиснення) даних використовується для зменшення часу їх передачі. Багато апаратних і програмних засобів мережі здатні виконувати динамічну компресію даних, тобто компресію під час передачі – модеми, мости, комутатори і маршрутизатори.

На практиці використовуються декілька алгоритмів компресії, кожен з яких ефективний для певного типу даних. Деякі модеми пропонують адаптивну компресію, при якій в залежності від типу даних обирається певний алгоритм.

Деякі загальні теоретичні алгоритми компресії .

Десяткова упаковка. Якщо всі дані у кадрі складаються із десяткових цифр, то можна замість ASCII кодів цифр використовувати просте двійкове кодування цифр. При цьому на одну цифру кількість біт зменшується з 8 до 4. (коефіцієнт стиснення, тобто відношення об’єму вихідних даних до вхідних, 50%, або 1:2)

Відносне кодування. При передачі числових даних із невеликими відхиленнями між послідовними цифрами передаються тільки ці відхилення разом із відомим опорним значенням.

Алгоритм RLE (Run Length Encoding). Заміна послідовності однакових байт, які повторюються спеціальною трибайтовою послідовністю. В ній вказується значення байту, кількість його повторень, а також починають дану послідовність спеціальним керуючим символом. Наприклад. Вхідна послідовність aaaaabbbb, вихідна послідовність ~a5~b4, коефіцієнт стиснення k=6/9*100=67%, або 2:3.

Алгоритм найефективніший для графічних даних.

Статистичне кодування. У кадрі різні символи можуть зустрічатися з різною частотою. При статистичному кодуванні коди символів, які зустрічаються частіше заміняють кодами меншої довжини, а ті що зустрічаються частіше – кодами більшої довжини. Один із найвідоміших алгоритмів статистичного кодування є алгоритм Хафмана, який дозволяє будувати коди автоматично на основі відомих частот символів.

Алгоритм найефективніший для текстових даних, найменш ефективний для двійкових даних, де символи розподілені майже рівномірно, наприклад для кодів програм..

Протоколи компресії, які використовуються на практиці використовують в своїй роботі комбінацію із теоретичних алгоритмів компресії. Існують стандартні протоколи компресії, такі як V.42bis, а також нестандартні, фірмові протоколи. Реальний коефіцієнт компресії залежить від типу даних, наприклад, графічні і текстові дані стискаються добре, а коди програм погано.