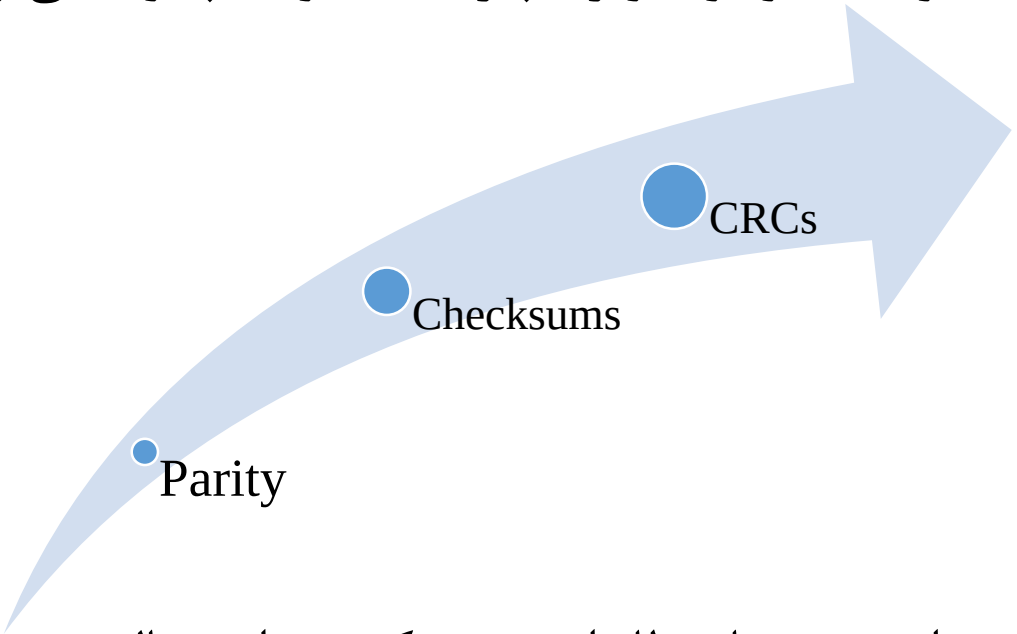
شبکه‌های مخابراتی

سید حمید صفوی
دانشکده فنی و مهندسی
دانشگاه محقق اردبیلی
نیمسال دوم ۹۸-۹۹

تشخیص خطا

سرفصل

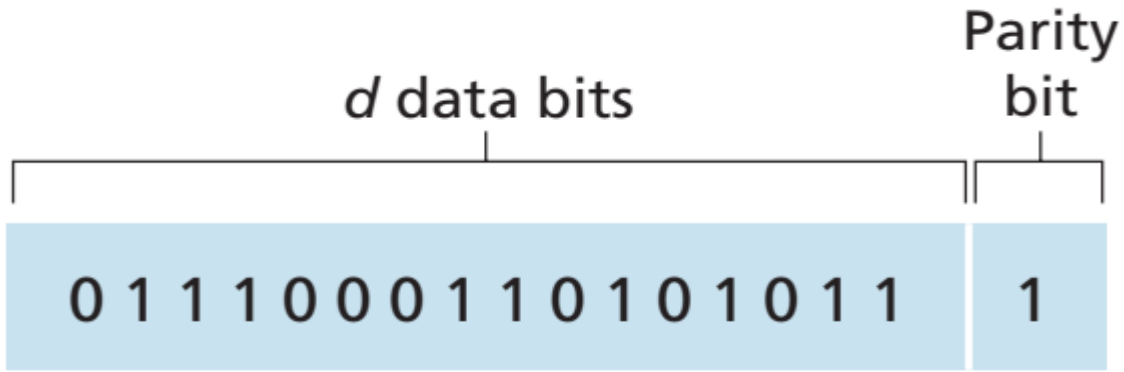
- ممکن است بعضی از بیت‌ها با توجه به وجود نویز دچار خطا شوند. چگونه می‌توانیم این بیت‌ها را شناسایی کنیم؟



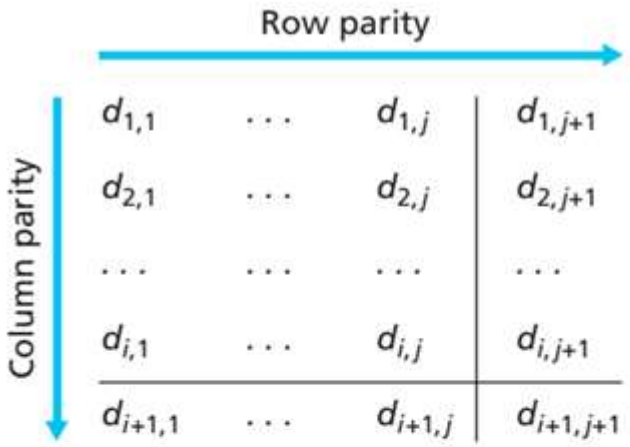
- شناسایی خطا به ما این فرصت را می‌دهد تا خطا را تصحیح کنیم برای مثال، به وسیله ارسال مجدد.
(بعداً بررسی خواهد شد)

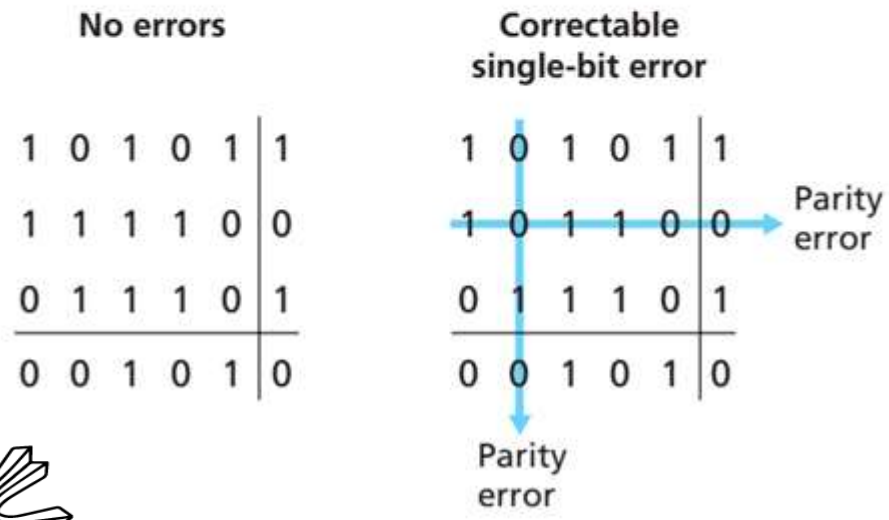
شناسایی خطای ساده – بیت توازن

- به ازای هر D بیت داده یک بیت توازن به محتوا اضافه می‌کندکه این بیت توازن حاصل‌جمع D بیت داده است.
  - حاصل جمع در مبنای ۲ یا XOR است.



بیت توازن





- **Parity چقدر خوب می تواند عمل کند؟**

– فاصله همینگ (distance) این کد چقدر است؟ ۲

– چند خطا را می‌تواند شناسایی و اصلاح کند؟

یک بیت خطا را شناسایی می‌کند ولی نمی‌تواند خطایی را اصلاح کند.

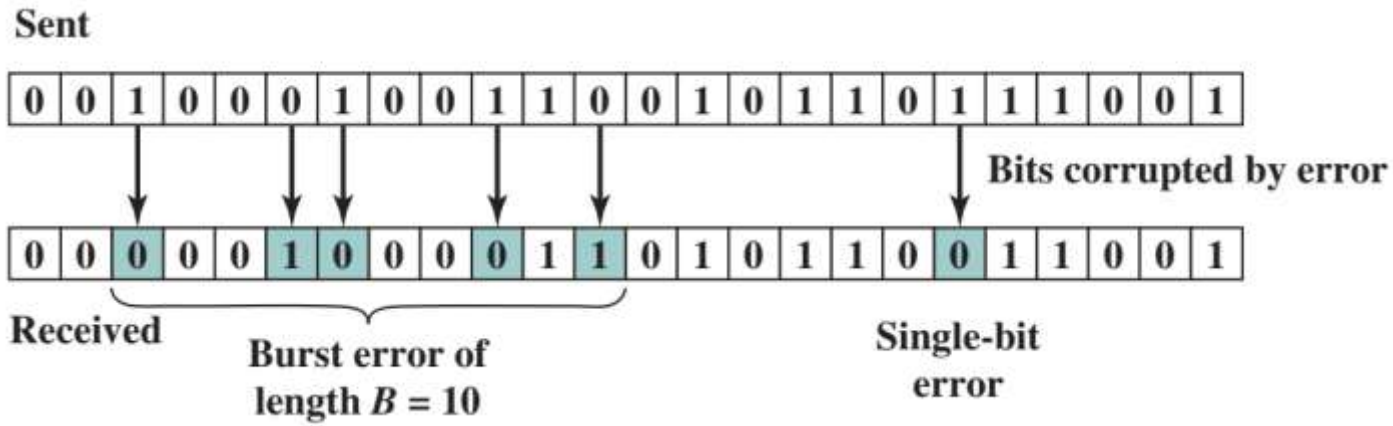
- **در مورد خطاهای بزرگ‌تر موفق عمل می‌کند؟**

خیر. فقط تعداد فرد خطاها را تشخیص می‌دهد.

- **Parity های دوبُعدی**

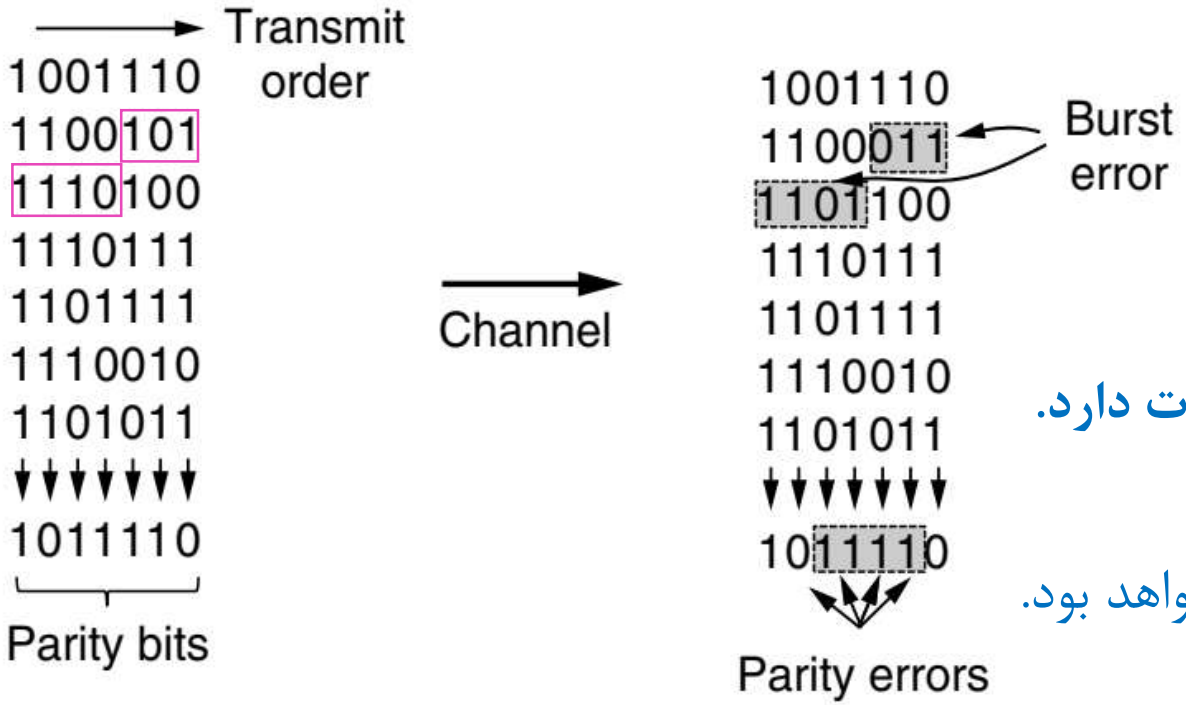
تشخیص خطای انفجاری با بیت توازن

- ایده‌ای برای تشخیص خطای انفجاری با بیت توازن دارید؟
- **خطای انفجاری** به این مفهوم نیست که از بسته مثلا M تایی همه بیت‌ها خطا باشند. به این مفهوم است که حداقل بیت ابتدا و انتهای بسته M تایی دارای خطا باشد.



بیت توازن با Interleaving

- بیت توازن با Interleaving برای شناسایی **خطاهای انفجاری (Burst error)** کاربرد دارد.



- **ایده اصلی:**
  - **ترتیب ارسال با ترتیب محاسبه بیت توازن تفاوت دارد.**
- داده ها را در n ستون k سطری مرتب می کنیم.
- بدیهی است که تعداد خطای n+1 قابل تشخیص نخواهد بود.

بیت توازن محاسبه شده: 0000

Checksums

Checksums

- ایده اصلی: داده‌ها را در کلمه‌های N بیتی (N-bit words) جمع می‌زند.
  - به صورت گسترده در TCP ، IP ، UDP استفاده می‌شود.

- نسبت به بیت توازن قابلیت حفاظت بیشتری دارد.

Internet Checksums

- حاصل‌جمع، بر روی **حساب ریاضی مکمل یک** تعریف می‌شود. در محاسبه باید **حامل‌ها** نیز جمع شوند.

سپس حاصل را منفی می‌کنیم.

"The Checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words…" - RFC791

Internet Checksums (2)

00 01 F2 03 F4 F5 F6 F7 00 00

ارسال:

۱) داده‌ها را به صورت کلمه‌های ۱۶ بیتی (۴ رقم hexadecimal) مرتب کنید.

```
  0001
  f203
  f4f5
  f6f7
+(0000)
------
```

۲) صفر را در جایگاه checksum قرار دهید و جمع بزنید.

۳) برای رسیدن به ۱۶ بیت باید تمام اعداد حامل موجود را از عدد به دست آمده جدا کرده و با خود عدد جمع بزنید.

۴) حاصل را مکمل منفی کنید.

Internet Checksums (3)

```
  0001
  f203
  f4f5
  f6f7
+ 220d
------
```

دریافت:

۱) داده‌ها را به صورت کلمه‌های ۱۶ بیتی مرتب کنید.

۲) Checksum غیرصفر را به عدد اضافه کنید و با آن جمع بزنید.

۳) برای رسیدن به ۱۶ بیت باید تمام اعداد حامل موجود را از عدد به دست آمده جدا کرده و با خود عدد جمع بزنید .

۴) حاصل را منفی کنید. «اگر حاصل صفر باشد به این معناست که خطایی در ارسال پیام رخ نداده است و اگر غیرصفر باشد به معنای وجود خطاست»

Internet Checksums (3)

00 01 F2 03 F4 F5 F6 F7 00 00

Partial sum	0001 F203 F204
Partial sum	F204 F4F5 1E6F9
Carry	E6F9 1 E6FA
Partial sum	E6FA F6F7 1DDF1
Carry	DDF1 1 DDF2
Ones complement of the result	220D

(a) Checksum calculation by sender

Partial sum	0001 F203 F204
Partial sum	F204 F4F5 1E6F9
Carry	E6F9 1 E6FA
Partial sum	E6FA F6F7 1DDF1
Carry	DDF1 1 DDF2
Partial sum	DDF2 220D FFFF

(b) Checksum verification by receiver

Internet Checksums (4)

- checksum تا چه اندازه می‌تواند خوب کار کند؟
  - فاصله (distance) این کد چقدر است؟ ۲
  - چند خطا را می‌تواند تشخیص و تصحیح کند؟
    یک خطا را شناسایی می‌کند ولی نمی‌تواند خطایی را تصحیح کند.
- در مورد خطاهای بزرگ‌تر موفق عمل می‌کند؟
  خطاهای انفجاری تا ۱۶ بیت.

به دلیل اینکه این روش بر روی کلمه کد عمل می‌کند و نه بیت‌ها، می‌تواند **خطاهایی را که دچار تغییر بیت توازن نمی‌شوند ولی checksum را تغییر می‌دهند** را تشخیص دهد.

Internet Checksums (5)

0	0	0	1	F	2	0	3	F	4	F	5	F	6	F	7	0	0	0	0

0000	0000	0000	0001	1111	0010	0000	0011	1111	0100	1111	0101	1111	0110	1111	0111	0000	0000	0000	0000

- مثال قبلی را در نظر بگیرید. بیت توازن زوج برای رشته بیت برابر صفر است. حال فرض کنید خطای انفجاری به صورت زیر رخ داده است و ۱۲ بیت را تحت تاثیر قرار داده است:

0000	0000	0000	0001	1111	1111	0010	0000	1111	0100	1111	0101	1111	0110	1111	0111	0000	0000	0000	0000

- با رخ دادن خطای انفجاری در بیت‌های نشان داده شده، بیت توازن همچنان صفر است، اما checksum که برابر 220D بود، با رخ دادن خطا عوض می‌شود. گیرنده برای چک کردن خطای پیش آمده از checksum بدون خطا که 220D است، استفاده می‌کند.

Internet Checksums (6)

دریافت:

0	0	0	1	F	F	2	0	F	4	F	5	F	6	F	7	2	2	0	D
0000	0000	0000	0001	1111	1111	0010	0000	1111	0100	1111	0101	1111	0110	1111	0111	0010	0010	0000	1101

```
   0001
   FF20     Error
   F4F5
   F6F7
+  220D     Checksum of correct word
----------
  30D1A
    ↓
   0D1A
+     3
----------
  0D1D  ⟶  F2E2
```

حاصل غیر صفر، معادل وجود خطا

بنابراین خطای انفجاری تشخیص داده شد.

Cyclic Redundancy Check (CRC)

Cyclic Redundancy Check (CRC)

- قابلیت حفاظت بیشتر و قوی‌تر

– برای n بیت داده، k تا check bit را به گونه‌ای می‌سازد که n+k بیت به دست آمده به صورت دسته‌هایی با تعداد زوج بر C که خود یک مولد است بخش‌پذیر باشد.

- مثال عددی:

n = 302 , k = one digit , C = 3

3021 mod(21÷3)=0

3020 mod(20÷3)=2

CRCs (2)

- دست آوردهای این روش:
  - بر پایه ریاضیات میدان‌های متناهی کار می‌کند که در آن اعداد بیانگر چند جمله‌ای‌ها هستند.
  - مثال e.g. 10011010 is $X^7 + X^4 + X^3 + X^1$

- معنای آن‌چه در بالا گفته شد چیست؟
  - ما با مقادیر دوتایی یا همان باینری کار می‌کنیم و از محاسبات ریاضی مبنای دو بهره می‌گیریم.

CRCs (3)

- **فرآیند ارسال:**

۱) n بیت داده را با k تا صفر بسط می‌دهد.

۲) با استفاده از مقدار مولد C تقسیم‌بندی را انجام می‌دهد.

۳) باقیمانده را نگه می‌دارد و خارج قسمت را نادیده می‌گیرد.

۴) k تا check bit را به وسیله باقیمانده تنظیم می‌کند.

- **فرآیند دریافت:**

۱) برای رسیدن به باقیمانده صفر تقسیم‌بندی و چک می‌کند.

CRCs (4)

Data Bits:

1101011111

Check Bits:

$X^4 + X^1 + 1$

C = 10011

K = 4

```
1 0 0 1 1 | 1 1 0 1 0 1 1 1 1 1 0 0 0 0
          - 1 0 0 1 1
            1 0 0 1 1
          -   1 0 0 1 1
                1 1 1 1 0
              - 1 0 0 1 1
                  1 1 0 1 0
                - 1 0 0 1 1
                      1 0 0 1 0
                    - 1 0 0 1 1
                          0 0 1 0      باقیمانده
1 0 0 1 1 1 1 0 1 0 1 1 1 1 1 0 0 1 0  فریم ارسالی
```

CRCs (5)

- قابلیت محافظت این روش وابسته به مقدار مولد است.

Standard CRC-32 is 100000100110000010001110110110111

- خاصیت‌های این روش:
  - HD=4، تا **سه بیت** خطا را می‌تواند تشخیص دهد.
  - همچنین **تعداد خطاهای فرد** را نیز تشخیص می‌دهد.
  - خطاهای انفجاری تا k بیت را تشخیص می‌دهد.
  - نسبت به خطاهای سیستماتیک آسیب‌ناپذیر است مانند checksums.

تشخیص خطا در عمل

- **CRC ها** به صورت گسترده در انواع **لینک‌ها** استفاده می‌شوند:
  - Ethernet , 802.11 , ADSL , Cable …

- **Checksum ها** در **اینترنت** استفاده می‌شوند:
  - IP , TCP , UDP … (اما ضعیف هستند)

- **بیت توازن**
  - بسیار کم استفاده می‌شوند.