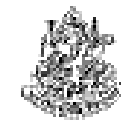
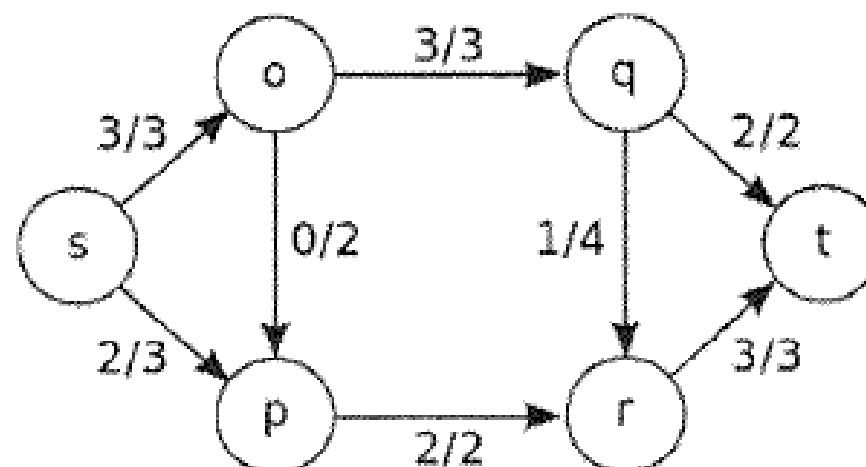


# Graph set06



## Network Flow และ Maximum Flow

- เนื่องจากกราฟถูกมองเป็นเน็ตเวิร์คที่เชื่อมต่อกันไป
  - โหนดแต่ละโหนดอาจมองได้ว่าเป็นเครื่องคอม
  - เส้นเชื่อมที่มีค่าน้ำหนักคือขนาดข้อมูลที่ส่งข้ามไปมากันได้
  - เราต้องการทราบว่าเราสามารถส่งข้อมูลจาก source ไปยัง terminal ได้สูงสุดเท่าใดในคราวเดียว (พูดง่าย ๆ ก็คือคิดจะส่งช่วยกันหลายทาง แต่วามันก็อาจจะขัดขวางกันเอง และดูยากกว่าจะใช้เส้นทางไหนบ้างและต้องการแบนด์วิธเท่าใด)
  - ปัญหาแบบนี้เราเรียกว่า Maximum Flow



ภาพจาก wikipedia

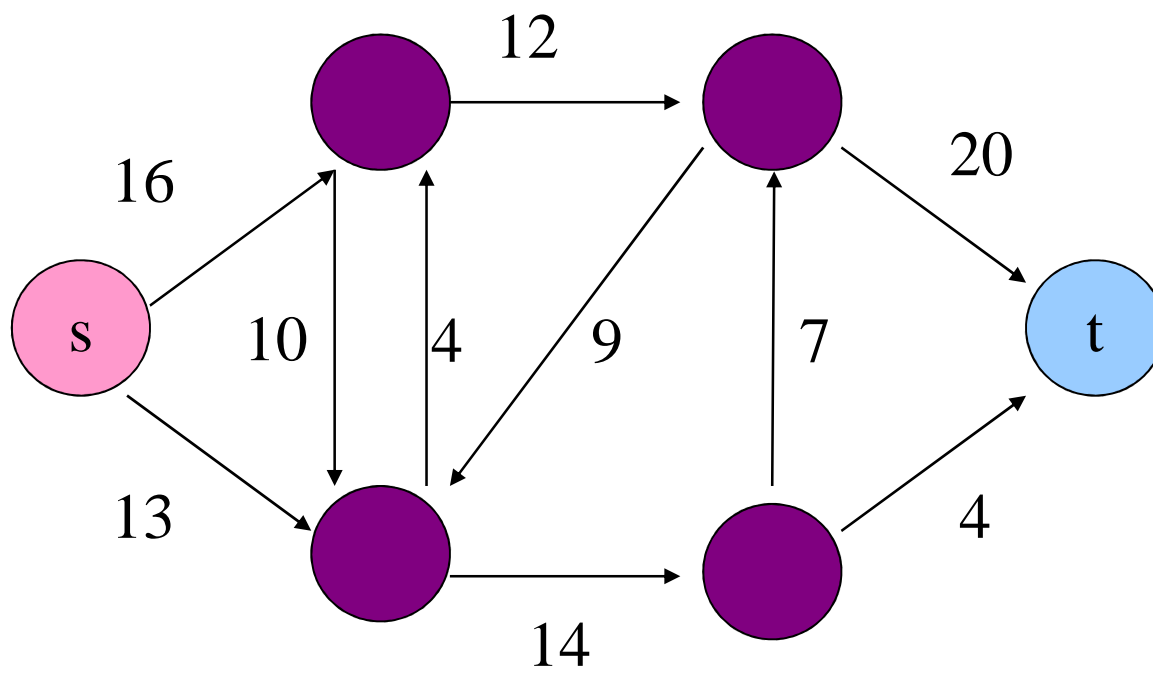


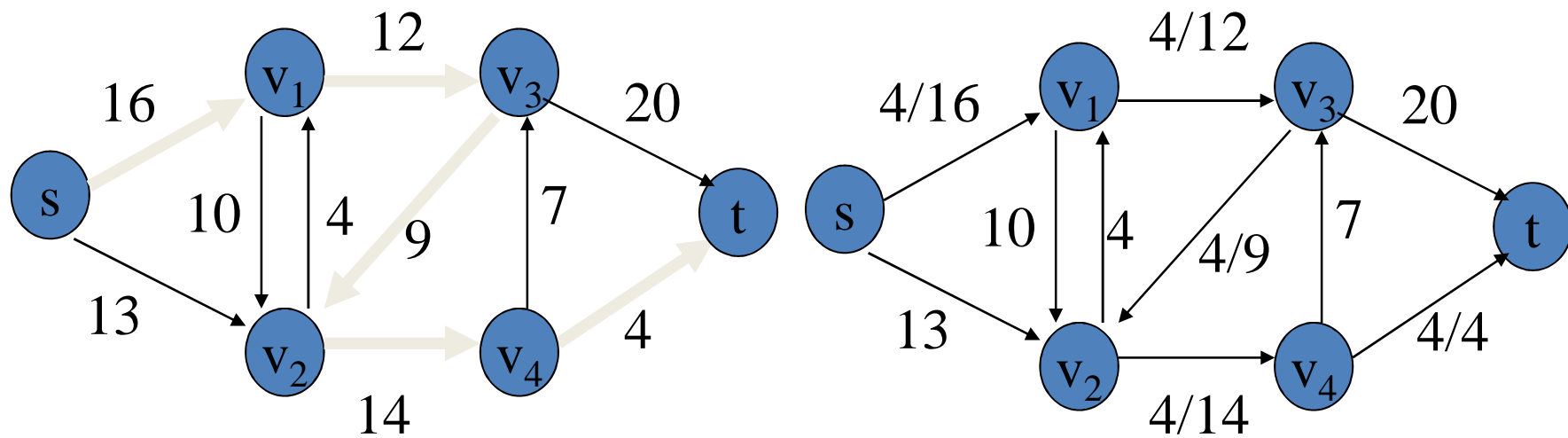
## Ford-Fulkerson's Method

- เป็นอัลกอริทึมสำหรับการหา Maximum Flow ที่หลักการเข้าใจง่าย
- คือจากความจุของเส้นเชื่อม ถ้าเราพบว่ายังมีทาง 'ปล่อยของ' จาก ต้นทาง (source) ไปปลายทาง (terminal) ก็ให้ปล่อยเพิ่มไปเรื่อย ๆ จนเต็ม
- ปัญหาคือจะรู้ได้ไงว่าจะปล่อยของเพิ่มได้หรือเปล่า
  - ถ้าปล่อยได้แสดงว่ามีเส้นทางที่ยังมีความจุเหลืออยู่จากต้นทางถึงปลายทาง
  - ดังนั้นก็ให้ค่อย ๆ หาเส้นทางซ้ำ ๆ ไปเรื่อย ๆ จนกว่าจะเต็มความจุทุกเส้นทาง
- ปัญหาอีกอย่างก็คือทางที่เลือกไปก่อนหน้านี้อาจไม่ค่อยดี เราจึงต้องคอยปรับใหม่ ซึ่งขั้นตอนนี้จะยุ่ง ๆ ในตอนแรกว่ามันเป็นไปได้ยังไง

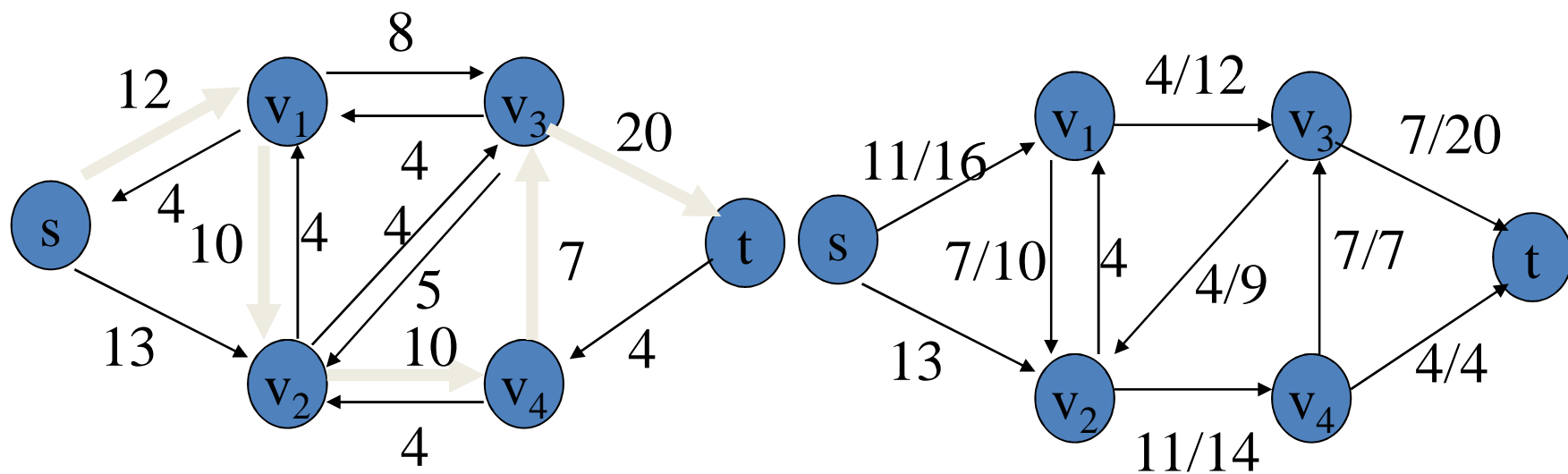
# Ford-Fulkerson algorithm:

```
FORD-FULKERSON( $G, s, t$ )  
  for each edge  $(u, v) \in E[G]$   
    do  $f[u, v] \leftarrow 0$   
        $f[v, u] \leftarrow 0$   
  while there exists a path  $p$  from  $s$  to  $t$  in the  
    residual network  $G_f$   
    do  $c_f(p) \leftarrow \min\{c_f(u, v) : (u, v) \text{ is in } p\}$   
       for each edge  $(u, v)$  in  $p$   
         do  $f[u, v] \leftarrow f[u, v] + c_f(p)$   
             $f[v, u] \leftarrow -f[u, v]$ 
```

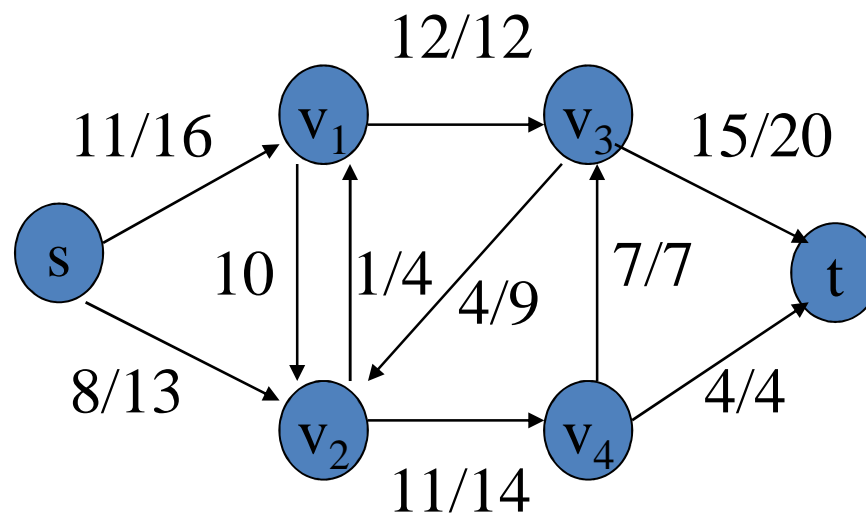
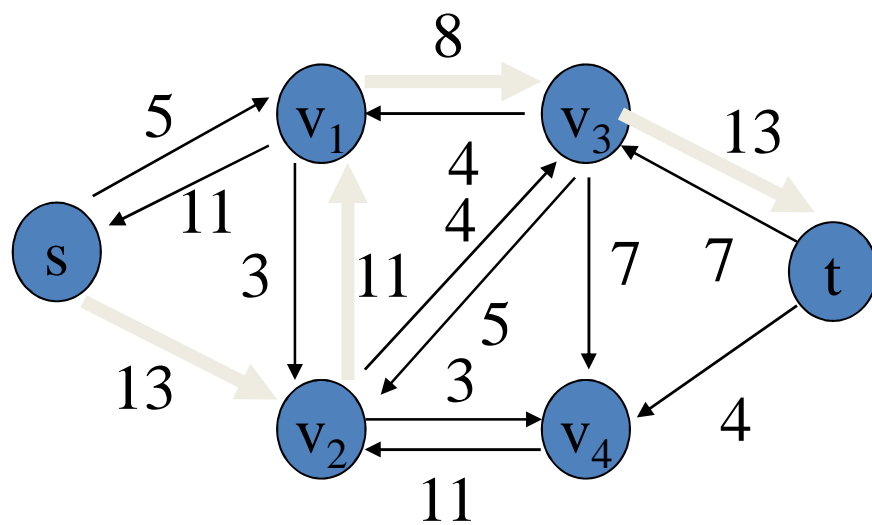




(a)

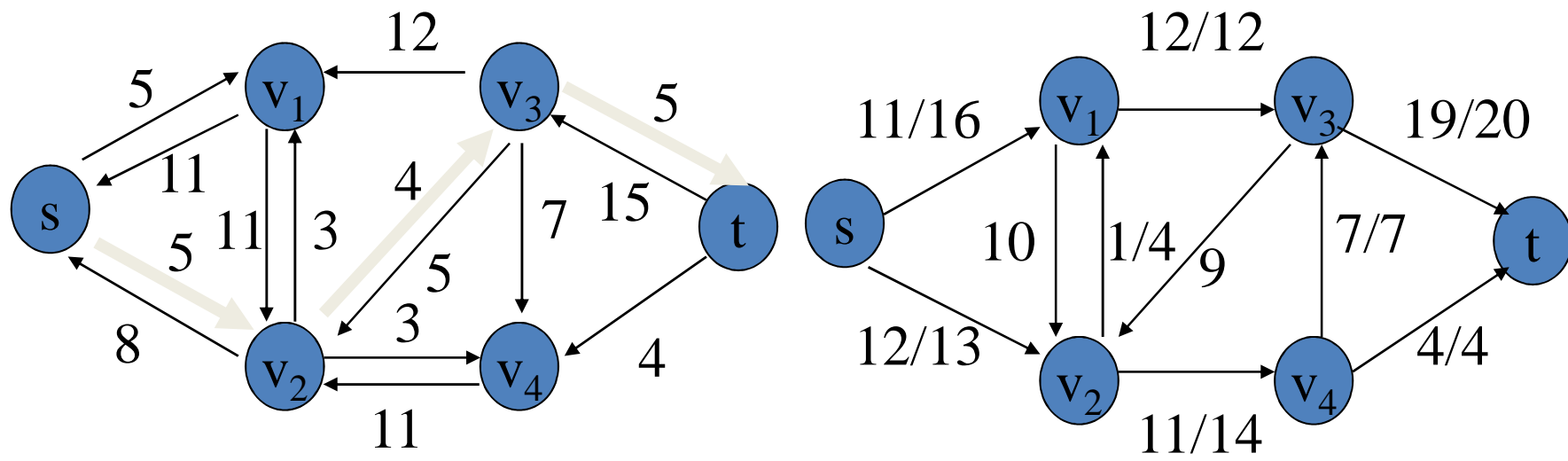


(b)



(c)

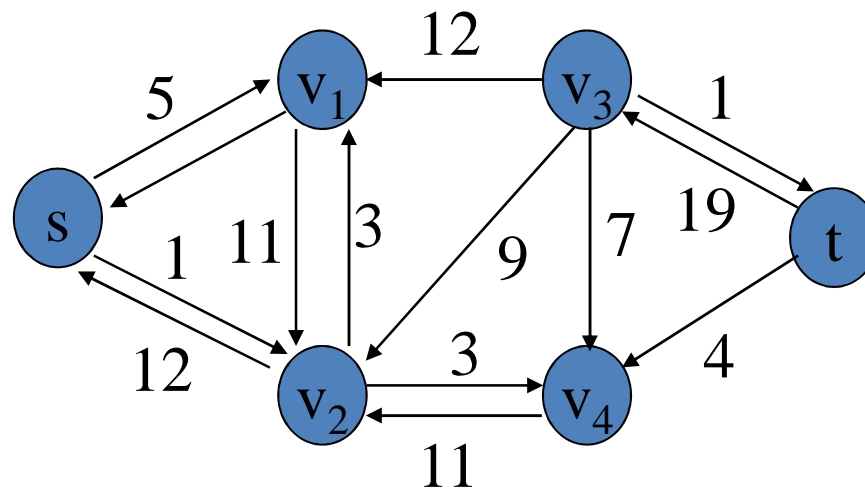




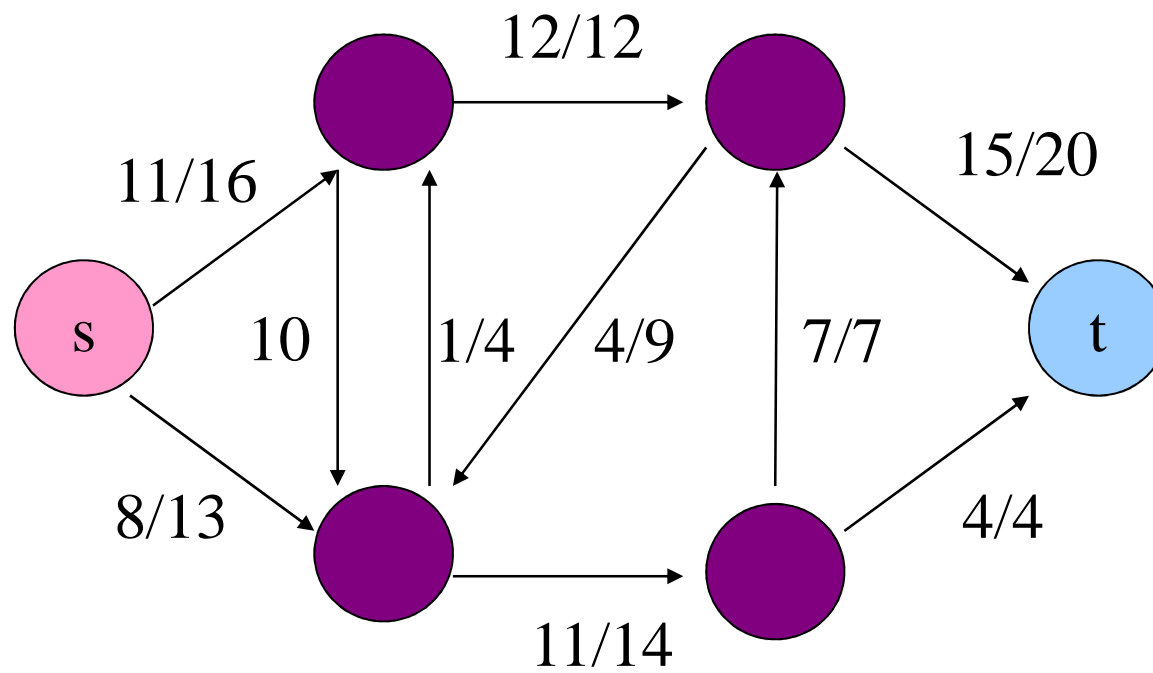
(d)

# Residual network test

No augmenting path



(e)



A flow  $f$  in  $G$  with value  $|f| = 19$  .

# Problems

- Internet Bandwidth (~8/820.html)
- **10594 – Data Flow (Hard)**