+

Introduction to UDP and TCP

End-to-end process mapping, reliable transmission and end-to-end congestion control

Chapter structure

1. Refresher about reliable transmission and the end-to-end principle
2. Intro to UDP

Supplement the Internet Service Model

- IP Service Model
  - Host-to-host (End-to-end)
  - Packets may not arrive in order
  - Packets may not arrive on time or not arrive at all
  - Packets can be corrupted
  - Packets can be duplicated by the network
- Applications need a reliable process-to-process channel
  - As though we were writing a file within an operating system (Unix, Windows…)
    - In order
    - No duplicates
    - No errors
    - With delivery guarantees

+ Properties of transport protocols

- Guarantees message delivery
- Delivers messages in the same order they were sent
- Delivers at most one copy of each message
- Supports arbitrarily large messages
- Supports synchronization between the sender and the receiver
- Allows the receiver to apply flow control to the sender
- Supports multiple application processes on each host

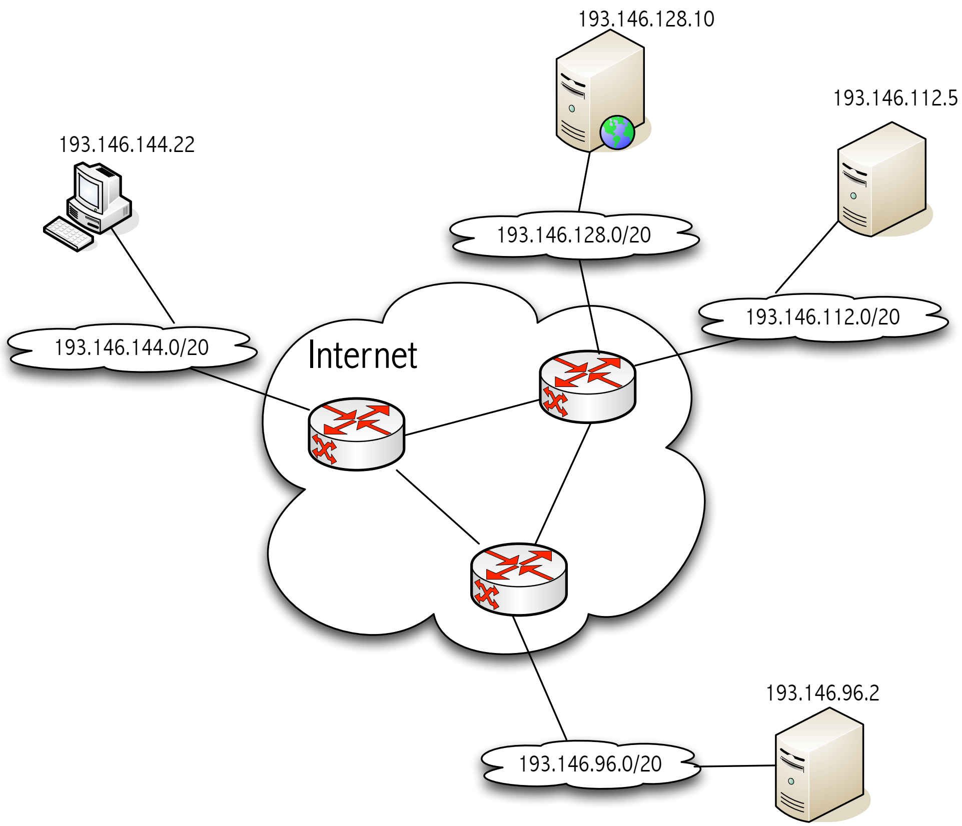
UDP: User Datagram Protocol

End-to-end unreliable packet delivery (Process-to-process)

Simple Demultiplexer (UDP)

- Extends host-to-host delivery service of the underlying network into a process-to-process communication service

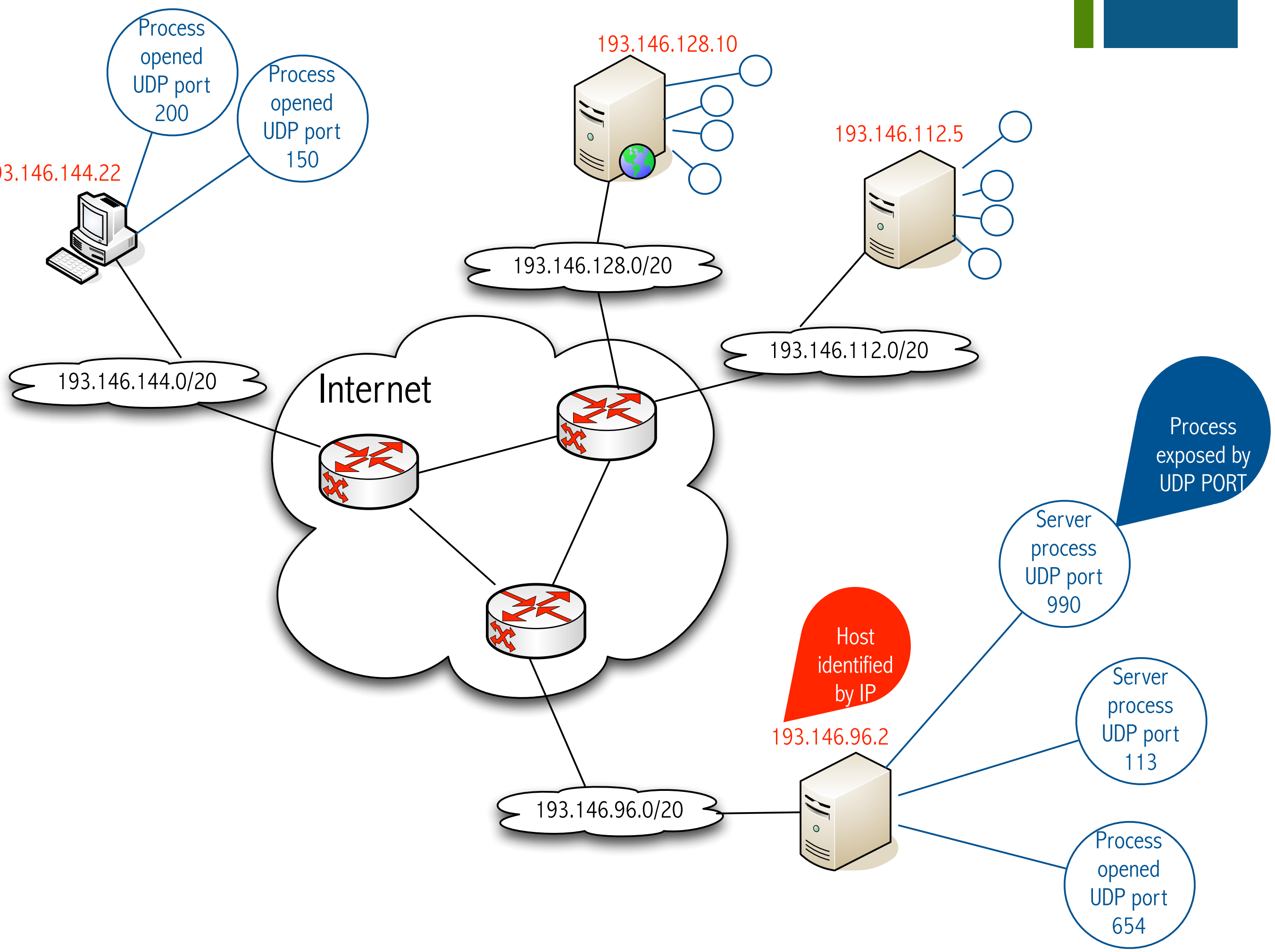
- Adds a level of demultiplexing which allows multiple application processes on each host to share the network



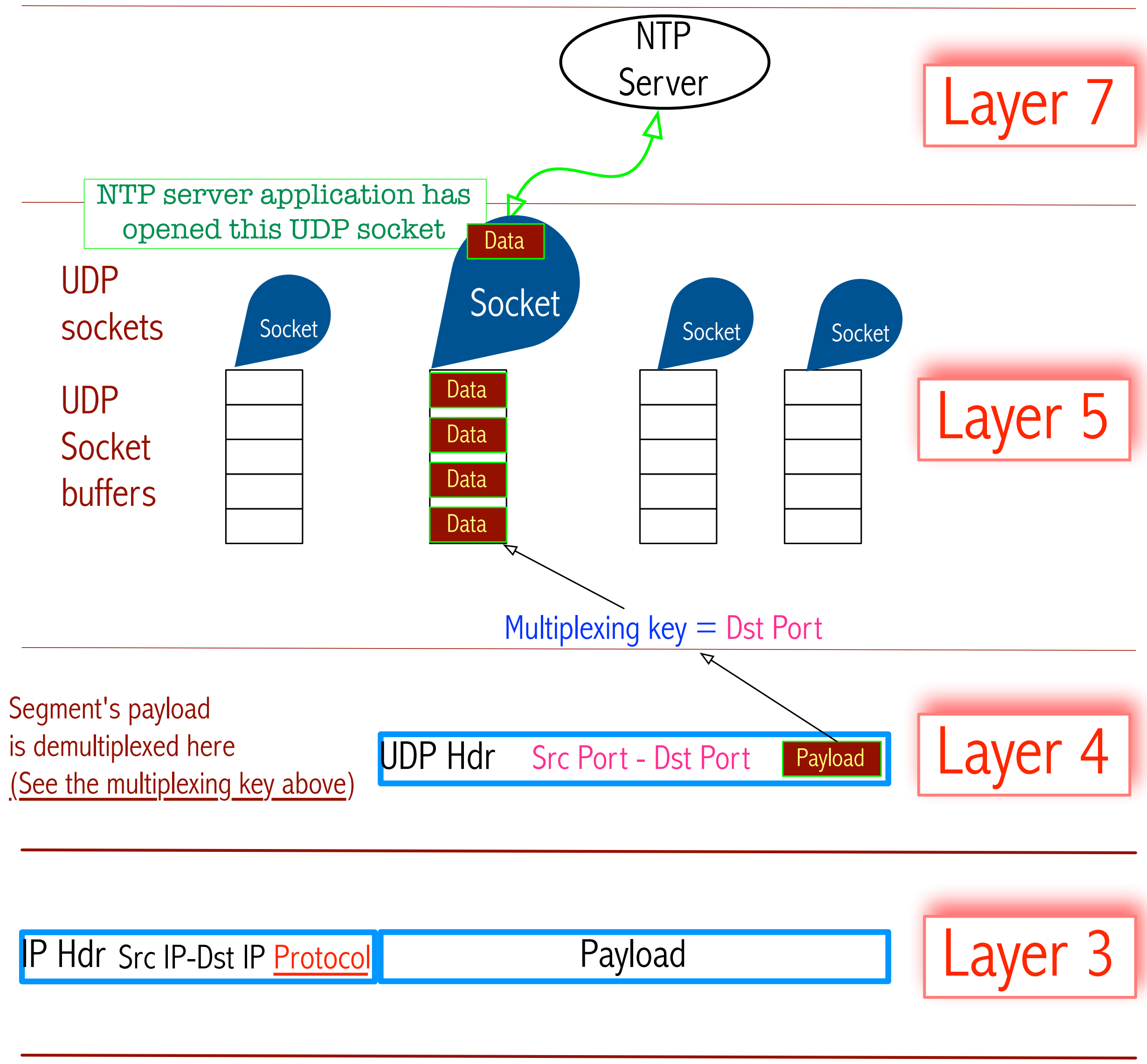
Simple Demultiplexer (UDP)

- Extends host-to-host delivery service of the underlying network into a process-to-process communication service

- Adds a level of demultiplexing which allows multiple application processes on each host to share the network

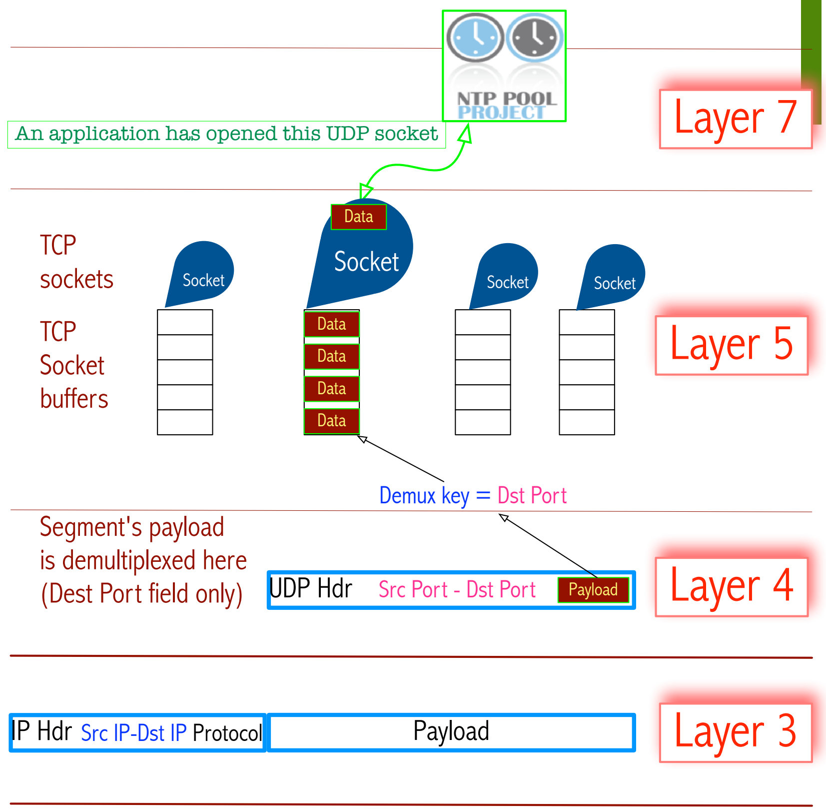


UDP: The Simple Demultiplexer



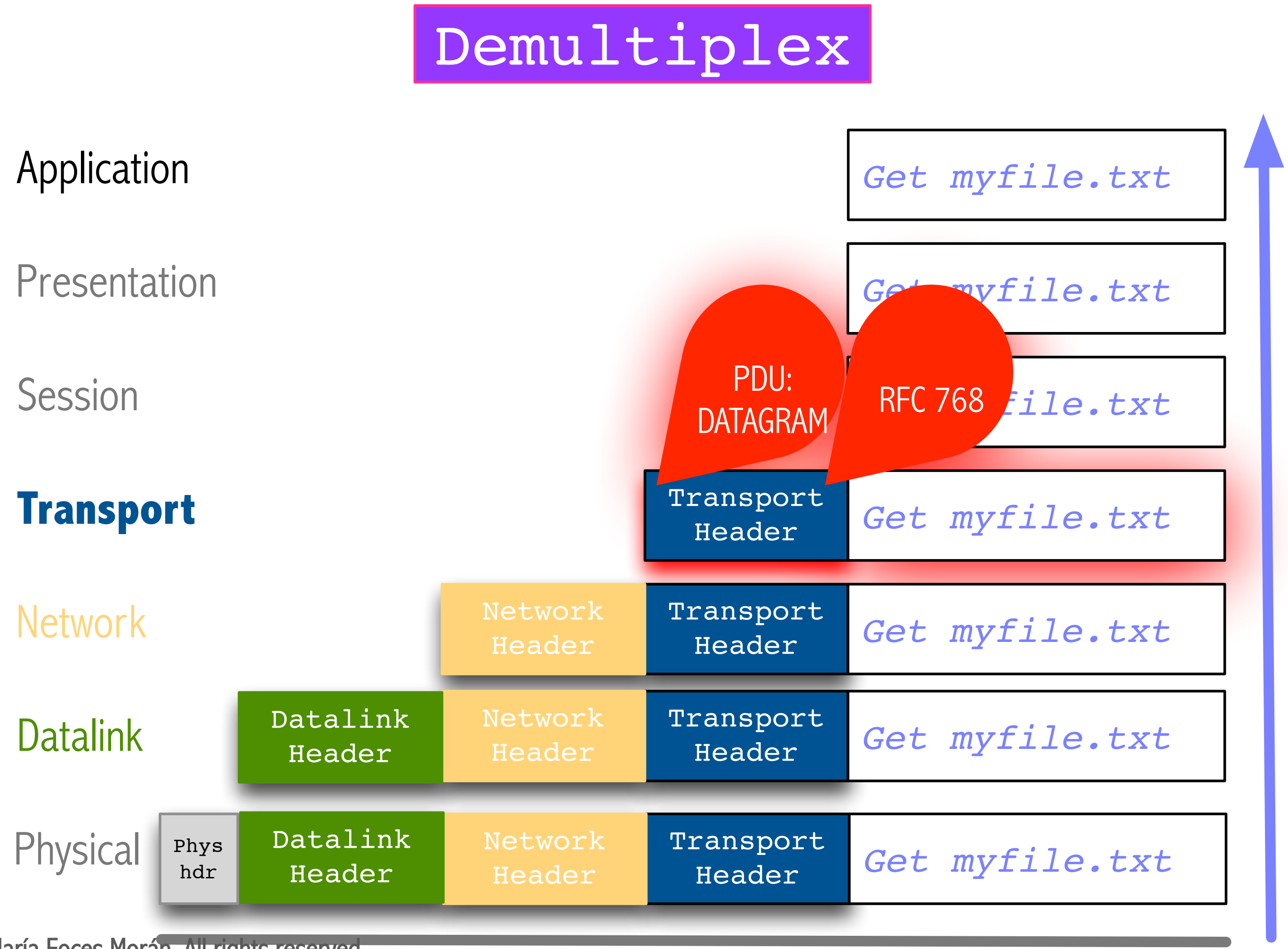
Little's Law
Datagram socket queue

- Little's Law
  - The theorem states that the mean number of customers in a queueing system is the product of their mean waiting time and their mean arrival rate
    - N is the mean number of customers in the queue (Queue length)
      - Datagram's data in our case
    - λ is the arrival rate in customers/sec
      - Datagrams/sec
    - T is the mean residence time
      - How long any datagram stays in the queue since it arrives until it is consumed by the application
  - The receiver may be overrun by the transmitter
    - This is a form of information loss



$$N = \lambda \cdot T$$

Simple Demultiplexer (UDP)



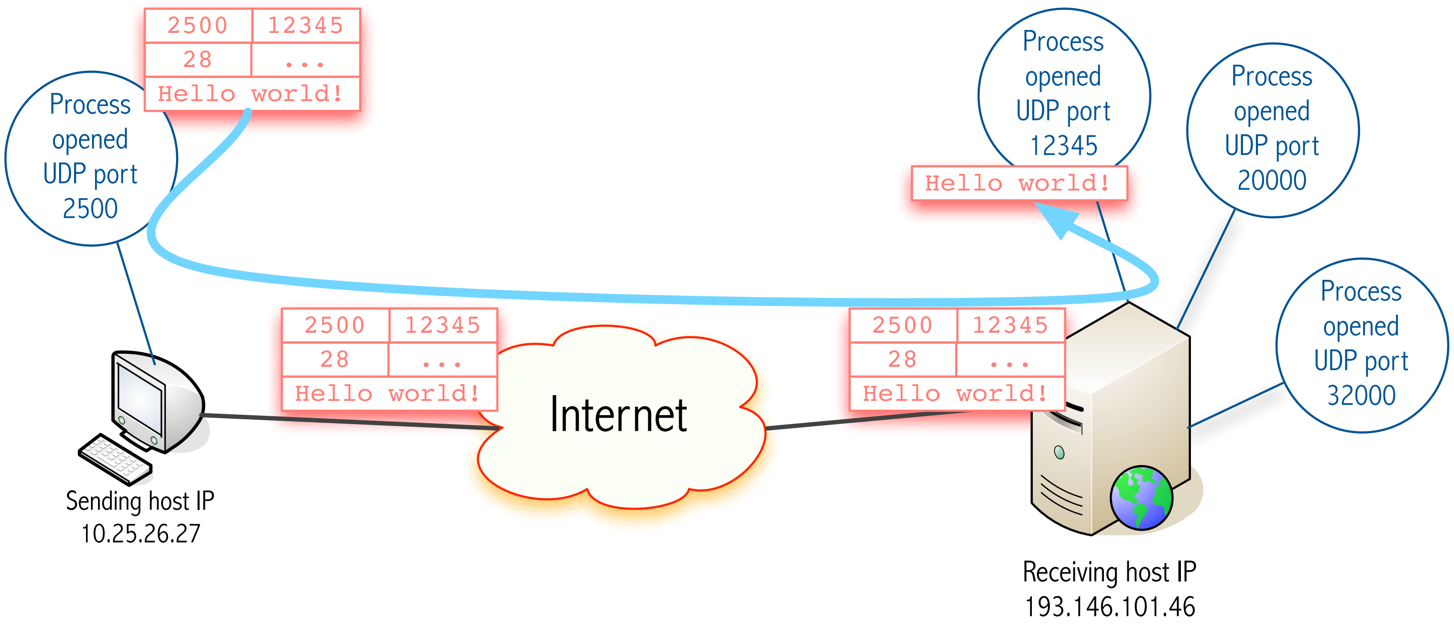
UDP Protocol Data Unit: Datagram RFC 768

- Source port:
  The port number opened by the sending process within the sending host
- Destination port:
  Port number opened by the receiving process within the receiving host
- Length: Number of bytes occupied by the whole datagram (Header + payload)
- Checksum: Internet checksum of the IP pseudoheader + datagram

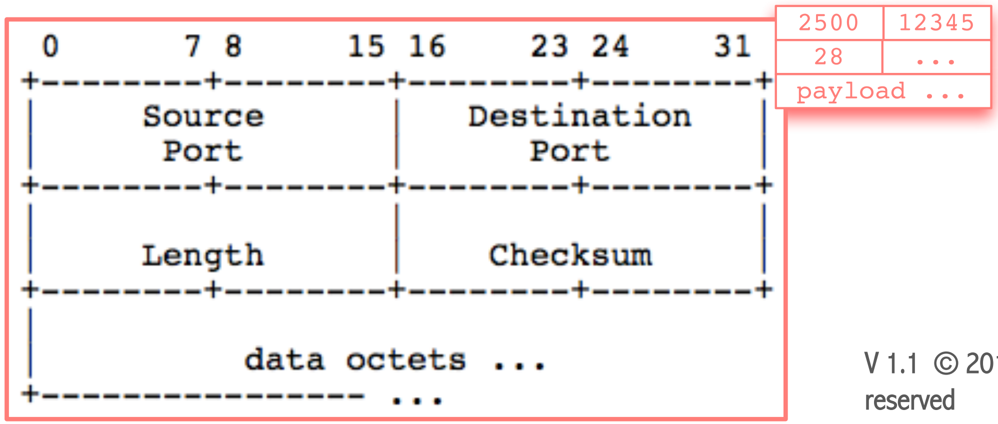
```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum     |
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...
```

UDP Segment (RFC 768 by Jon Postel, august 1980)

UDP demultiplexing example



UDP datagram



I used the following references in the composition of the present work (In order of importance):

1. Peterson and Davie, Computer Networks, MKP Elsevier 2012