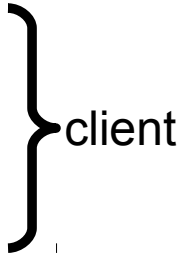


# **L19: Time & Ordering**

Nickolai Zeldovich  
6.033 Spring 2012

# Simple time sync protocol

```
sync(server):  
    tsrv = getTime(server)  
    local_time = tsrv
```



```
getTime():  
    return local_time
```



# Estimating network latency

sync(server):

$t\_begin = local\_time$

$tsrv = getTime(server)$

$t\_end = local\_time$

$delay = (t\_end - t\_begin) / 2$

$offset = (t\_end - delay) - tsrv$

$local\_time = local\_time - offset$

# Slew time

sync(server):

t\_begin = local\_time

tsrv = getTime(server)

t\_end = local\_time

delay = (t\_end - t\_begin) / 2

offset = (t\_end - delay) - tsrv

freq = base +  $\epsilon$  \* sign(offset)

sleep(freq \* abs(offset) /  $\epsilon$ )

freq = base

} temporarily  
speed up /  
slow down  
local clock

timer\_intr(): # on every oscillator tick..

local\_time = local\_time + 1/freq

# Adjust local frequency estimate

```
sync_freq(server):
```

```
    tc0 = local_time
```

```
    ts0 = getTime(server)
```

```
    sleep(N)
```

```
    tc1 = local_time
```

```
    ts1 = getTime(server)
```

```
    ratio = (tc1-tc0) / (ts1-ts0)
```

```
    freq = freq * ratio
```

} set local  
frequency  
to match  
server

# Summary

- NTP can synchronize time across the Internet
  - Be careful w/ assumptions, when using time
- Optimistic concurrency: concurrent changes
- Vector timestamps help detect concurrent changes