NEURON

for computer simulations of neurons and neural networks

The NEURON simulation environment

Networks and more

Michael Rudolph

Advanced Course in
Computational Neuroscience
Obidos, Portugal, 2004

References

web:

- `http://www.neuron.yale.edu/`
- `http://neuron.duke.edu/`

papers:

- Hines, M.L. and Carnevale, N.T.
  Discrete event simulation in the NEURON environment.
  Neurocomputing **58-60** (2004), 1117-1122.
- Carnevale, N.T. and Hines, M.L.
  Efficient discrete event simulation of spiking neurons using NEURON.
  Abstract (2003).

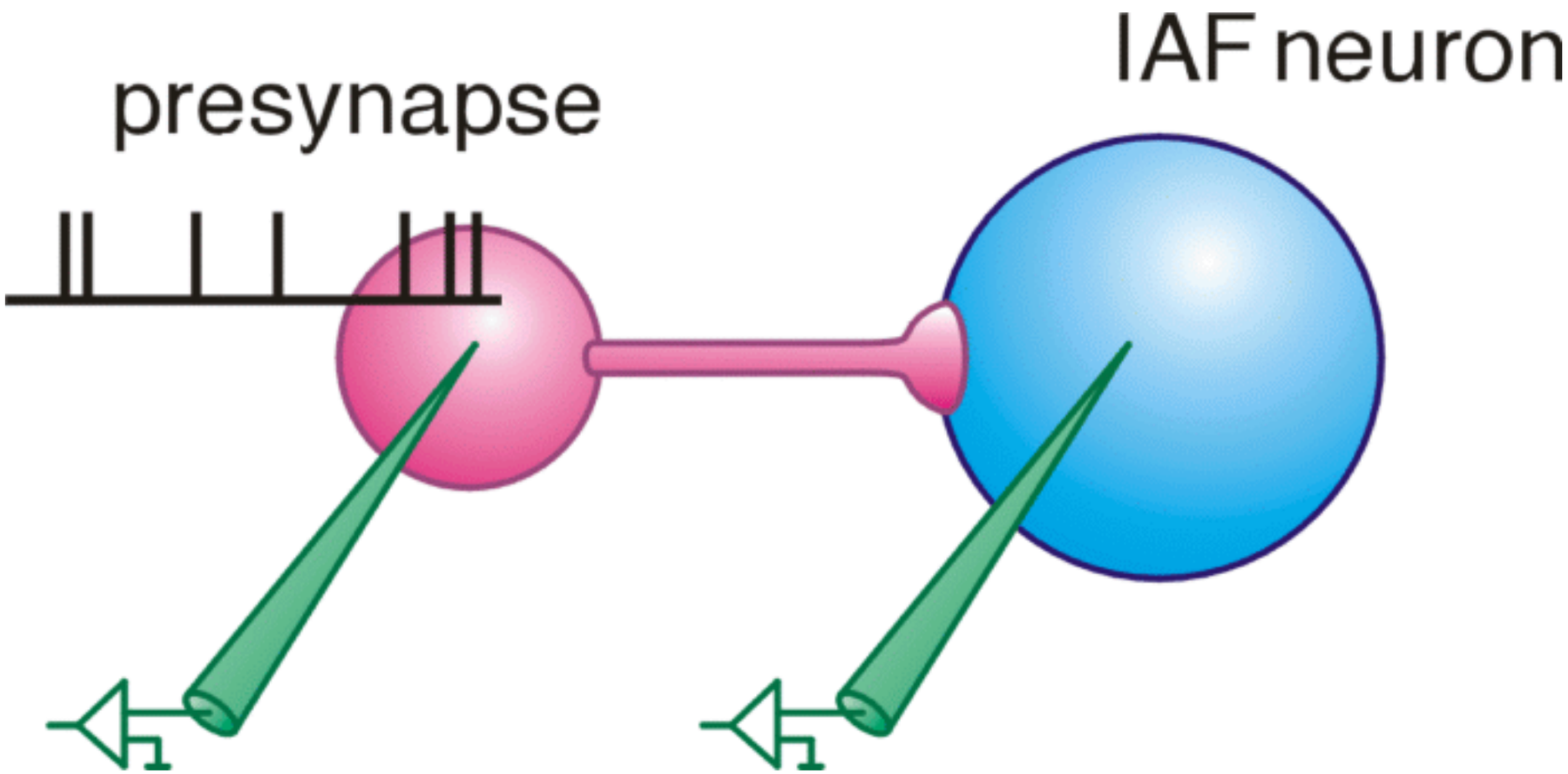
Outline

1. A very (very!) simple example
   (introducing NetCon, NetStim and templates)

2. Event-based approach to network modelling

1 A very (very!) simple example

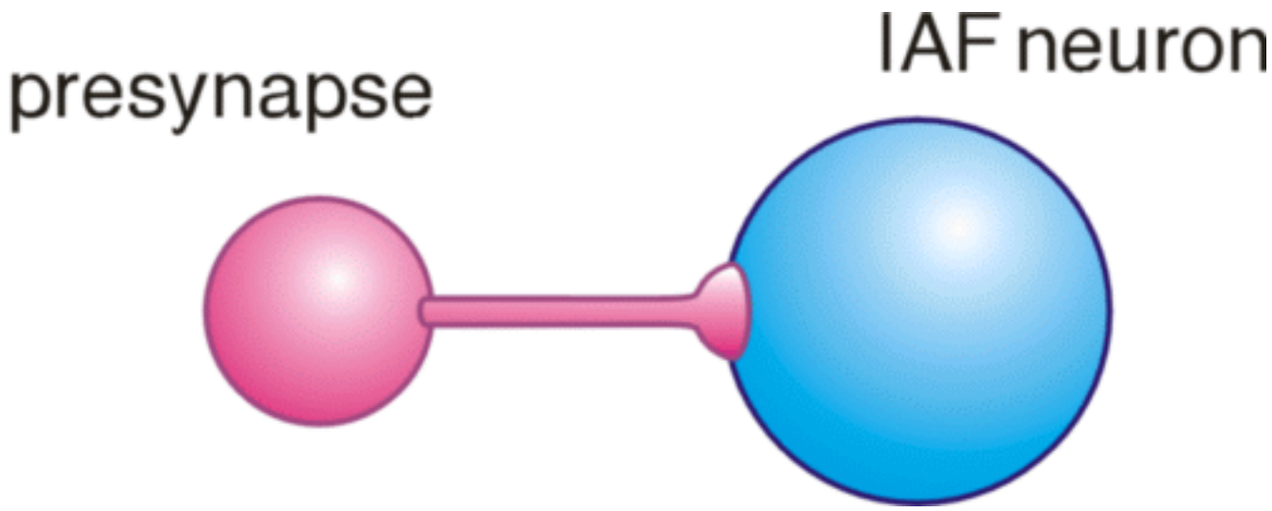
Goal: "Neural network" with one (and more) IAF neuron receiving random synaptic input



- single compartment with IAF point process `IntFire1` provided in NEURON
- "network input": single pulsed input using the `NetStim` point process provided in NEURON

❶ A very (very!) simple example

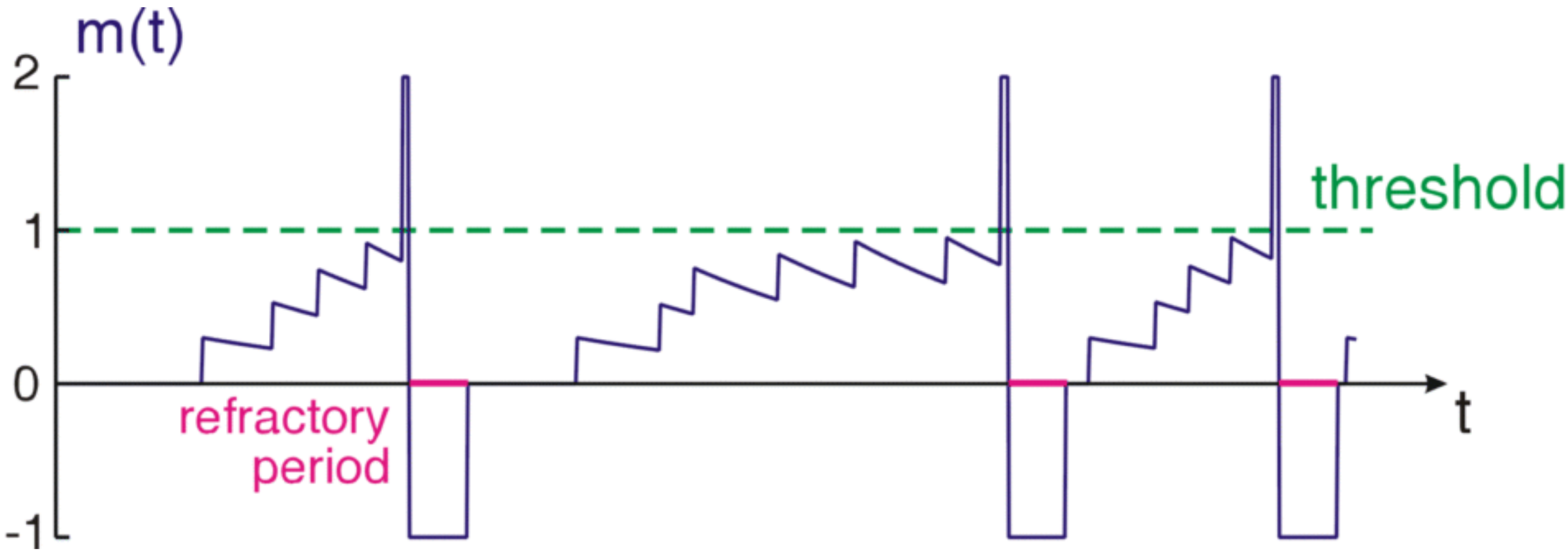
① Creating cell




```
objref IAF                                  create point process

create soma                                 create compartment

soma {
   IAF = new IntFire1(0.5)
   IAF.tau = 20                             create IAF cell
   IAF.refrac = 5
}
```

1 A very (very!) simple example

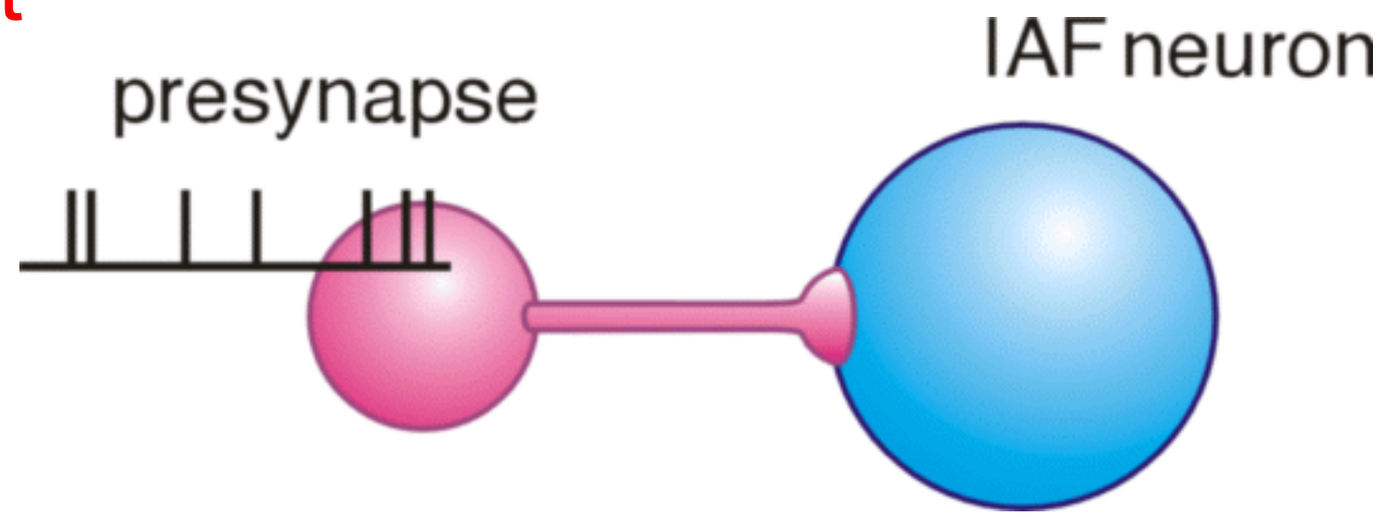
The `IntFire1()` object




```
c = new IntFire1(x)
c.tau
c.refrac
c.m
c.M
```

❶ A very (very!) simple example

② Adding network input




```
objref StimTrigger, NetInput
```
create stimulation object

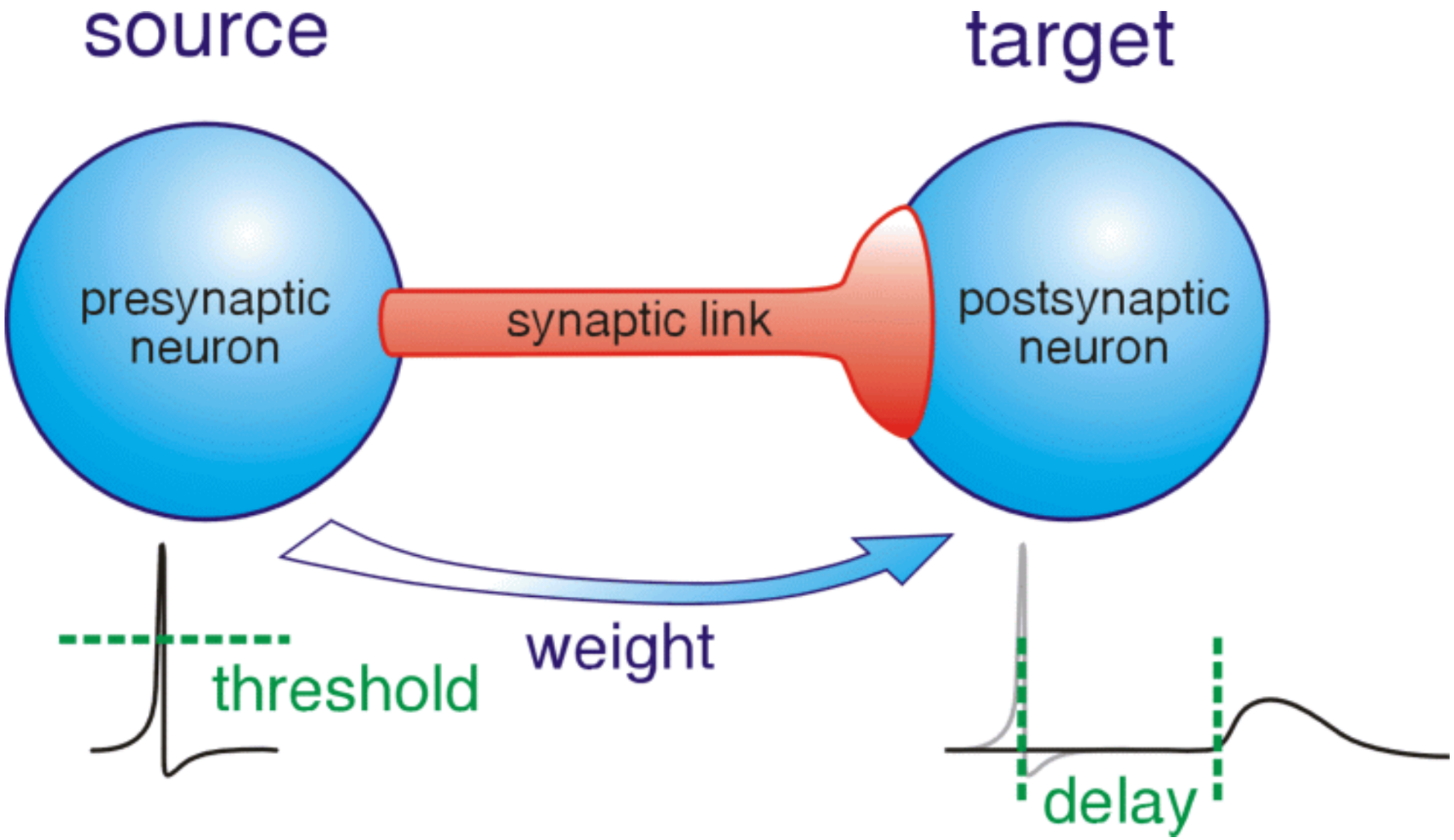
```
create Presynapse
```
create input compartment

```
Presynapse {
  StimTrigger = new NetStim(0.5)
  StimTrigger.start = 10
  StimTrigger.interval = 5
  StimTrigger.number = 50
  StimTrigger.noise = 0
```
create stimulation trigger

```
  NetInput = new NetCon(StimTrigger, IAF, 0.5, 0, 0.3)
}
```
connect stimulation trigger to network cell

1 A very (very!) simple example

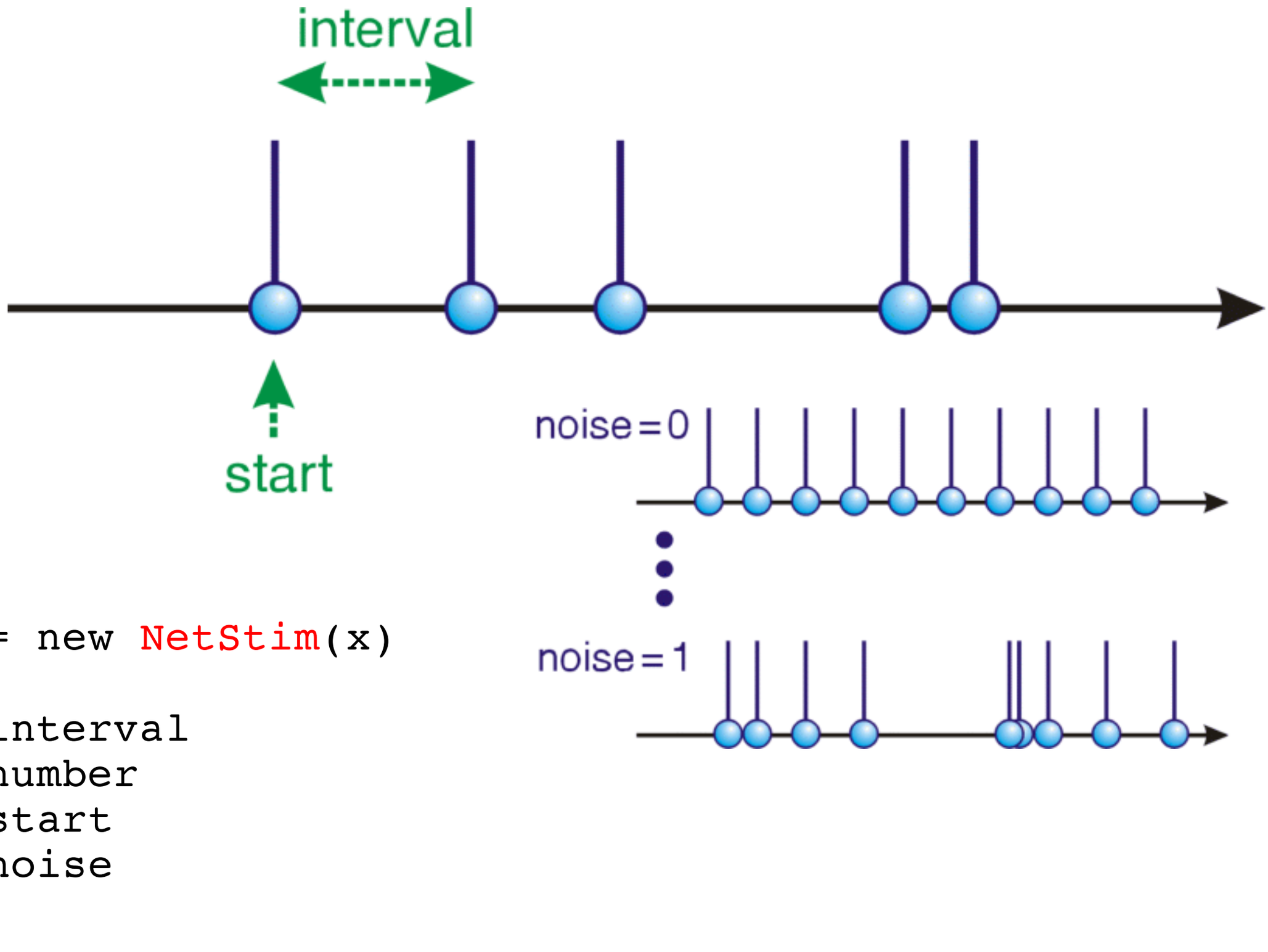
The `NetCon()` object




```
section netcon = new NetCon(&v(x), target, threshold, delay, weight)
netcon = new NetCon(source, target, threshold, delay, weight)
section netcon = new NetCon(&v(x), target)
netcon = new NetCon(source, target)
```

1 A very (very!) simple example

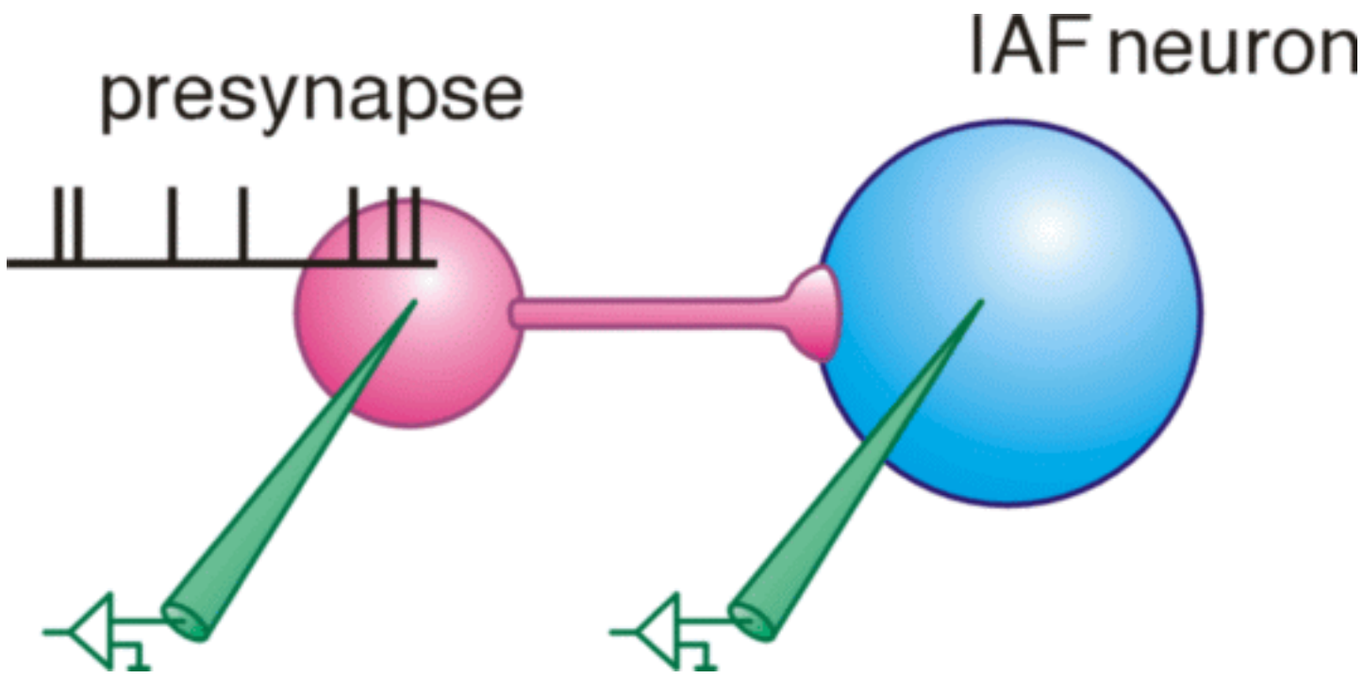
The `NetStim()` object

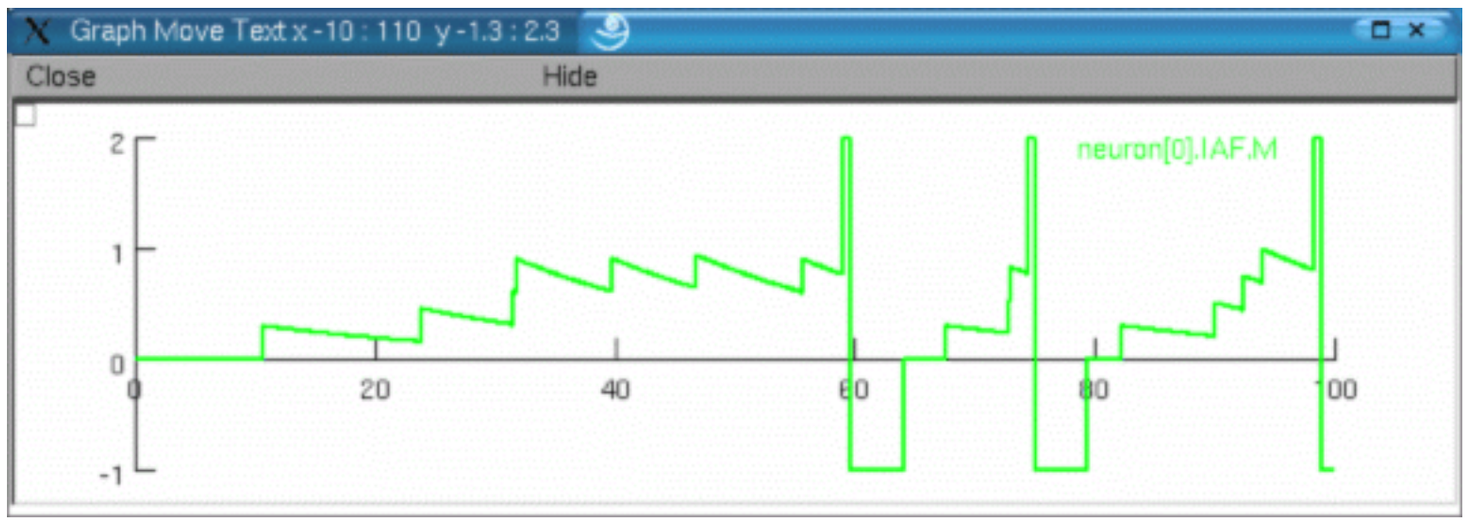



```
s = new NetStim(x)

s.interval
s.number
s.start
s.noise
```

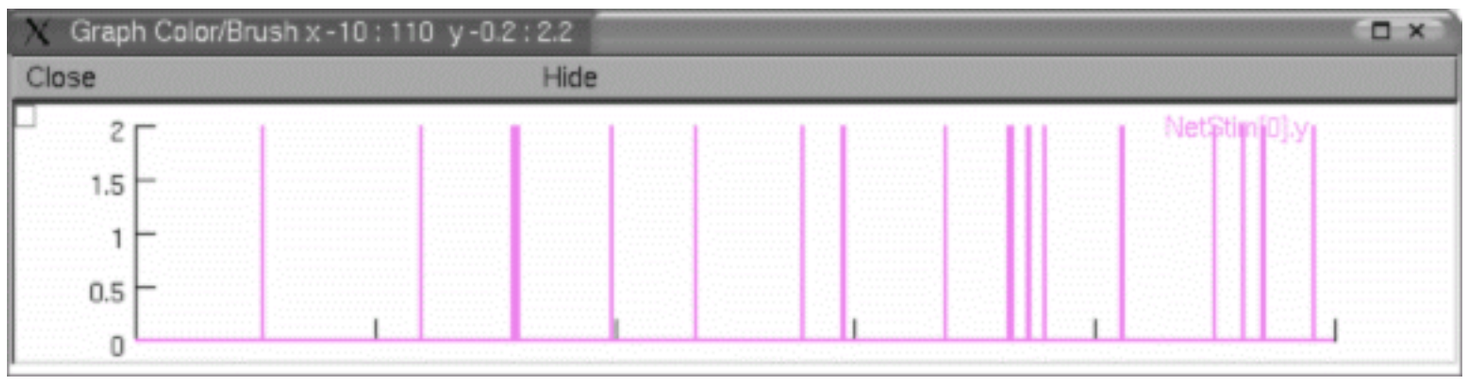
❶ A very (very!) simple example

③ Simulation





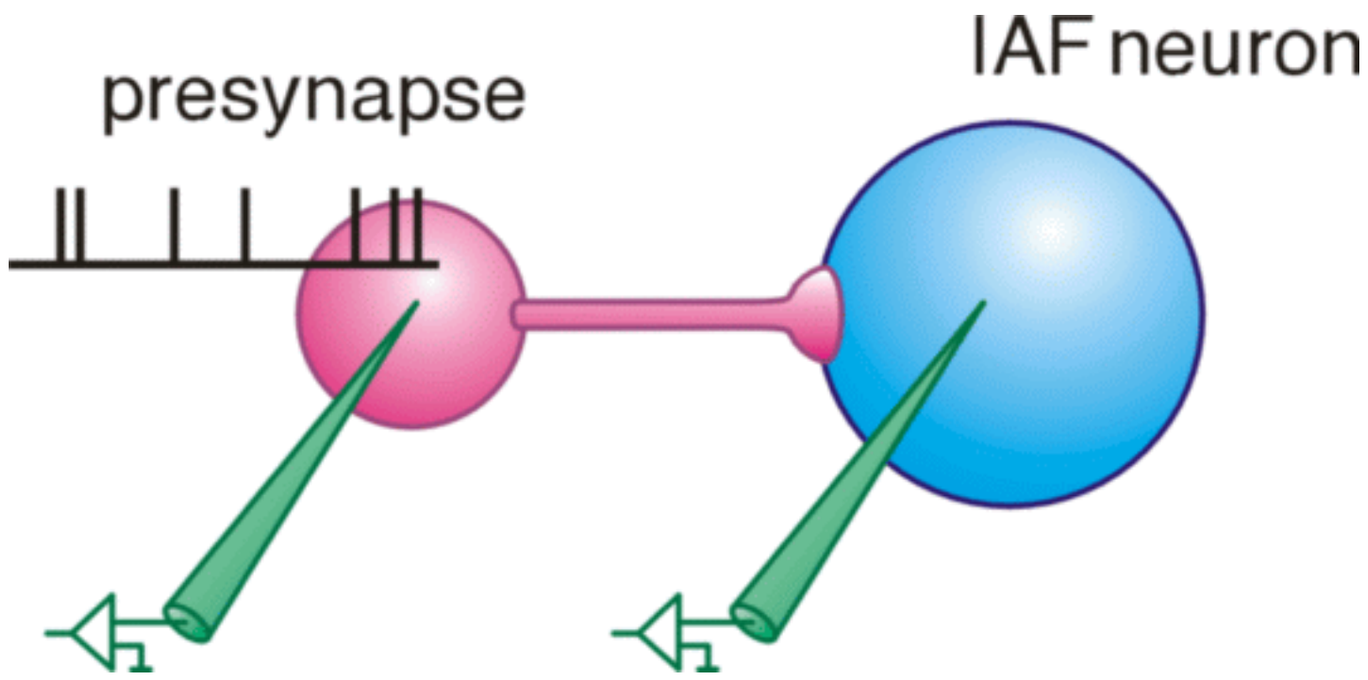
state variable
of IAF neuron



presynaptic
activity

❶ A very (very!) simple example

④ Incorporation of network topology




```
objref IAF

create soma

soma {
  pt3dclear()
  pt3dadd(0, 0, 0, 10)
  pt3dadd(10, 0, 0, 10)

  IAF = new IntFire1(0.5)
  IAF.tau = 20
  IAF.refrac = 5
}
```

set position of cell

❶ A very (very!) simple example

⑤ Using templates

```
begintemplate <NAME>

  public <VARIABLE NAME>
  <BODY>
  proc init() {
    <INITIALIZATION>
  }
  proc <PROCEDURE NAME> {
    <PROCEDURE>
  }

endtemplate <NAME>
```

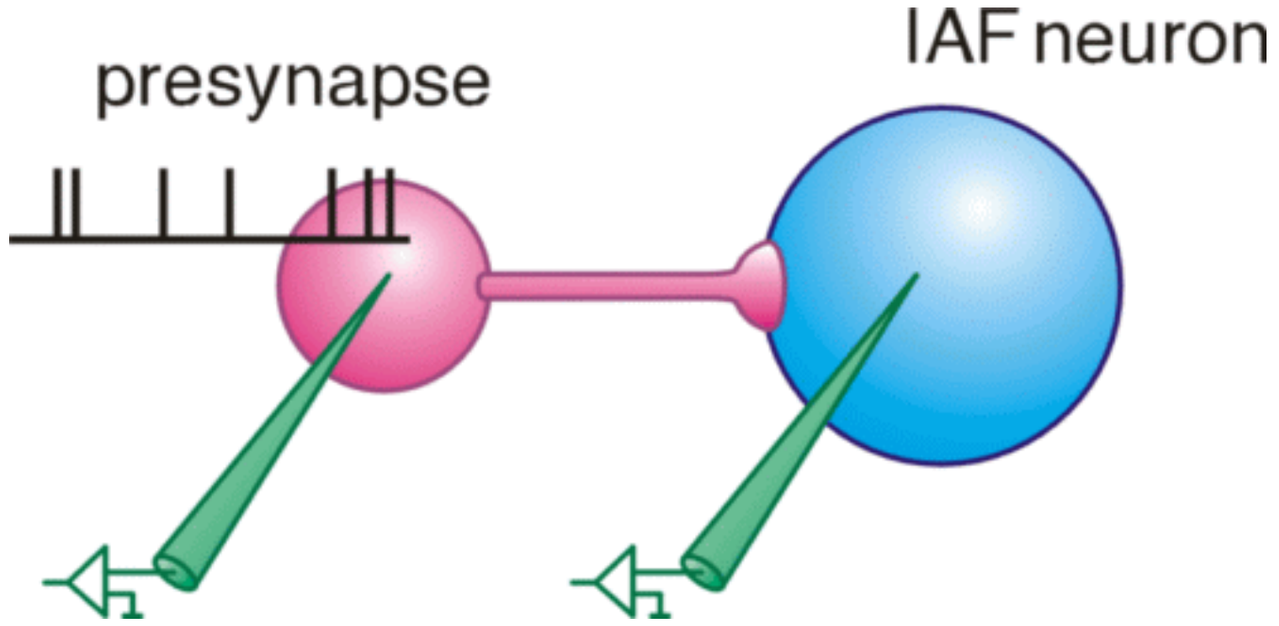
definition

```
objref neuron[NumberCells]

l = 0
for l=0, NumberCells {
  neuron[l] = new <NAME>()
  neuron[l].<VARIABLE NAME> = <VALUE>
}
```

usage

1 A very (very!) simple example

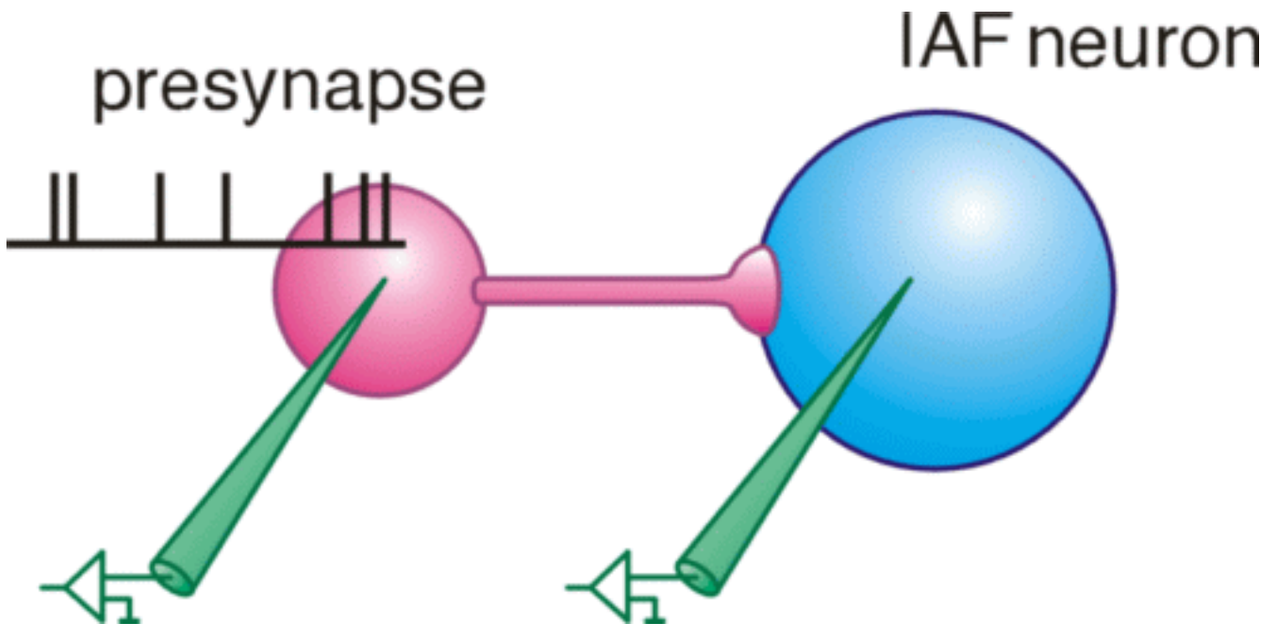
Definition of cell template




```
begintemplate simpleIAFneuron

  public soma
  public IAF
  public x, y, z
  public addConnection

  objref IAF
  create soma

  proc init() {
    x = $1
    y = $2
    z = $3
    soma {
      pt3dclear()
      pt3dadd(x, y, z, 10)
      pt3dadd(x+10, y, z, 10)

      IAF = new IntFire1(0.5)
      IAF.tau = 10
      IAF.refrac = 5
    }
  }

  proc addConnection() {
    soma {
      pt3dadd(x+5, y, z, 1)
      pt3dadd($1+5, $2, $3, 1)
    }
  }

endtemplate simpleIAFneuron
```

1 A very (very!) simple example

Usage of cell template




```
objref neuron[NumberCells]
```

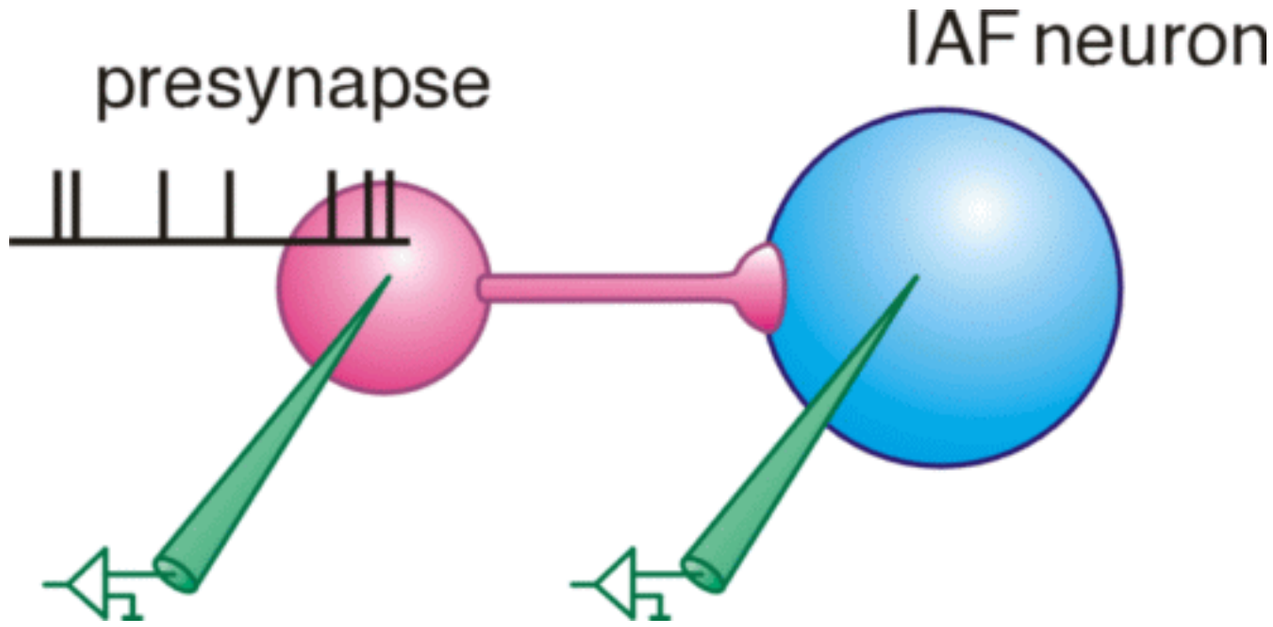
define objects for cells

```
l = 0
for i=0, NETDIM_X-1 {
  for j=0, NETDIM_Y-1 {
    for k=0, NETDIM_Z-1 {
      neuron[l] = new simpleIAFneuron(x+i*Dx, y+j*Dy, z+k*Dz)
      neuron[l].IAF.tau = IAF_TAU
      neuron[l].IAF.refrac = IAF_REFRAC
      l = l + 1
    }
  }
}
```

create objects for cells

1 A very (very!) simple example

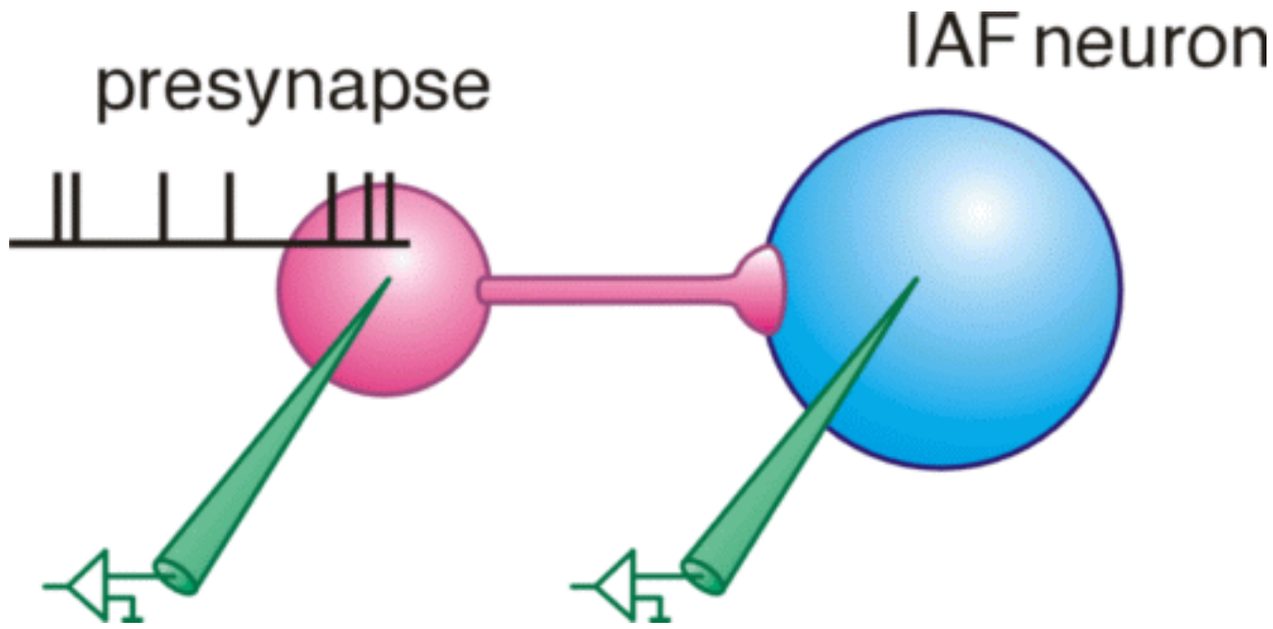
Connect cell objects




```
NumberConnections = 0
NumberPossibleConnections = 0
objref prob
prob = new Random()

objref CellConnection[MAXNUMBER_CONNECTION]

for i = 0, NumberCells-1 {
  for j = 0, NumberCells-1 {
    NumberPossibleConnections = NumberPossibleConnections + 1
    aprob = prob.uniform(0, 1)
    if ((aprob <= CONNECTION_PROB) && (i != j)) {
      adist = <EUCLIDIC DISTANCE>
      CellConnection[NumberConnections] = new NetCon(neuron[i].IAF,
                          neuron[j].IAF, CONNECTION_THRESHOLD,
                          CONNECTION_DELAY*adist, CONNECTION_WEIGHT)
      NumberConnections = NumberConnections + 1
    }
  }
}
```

1 A very (very!) simple example

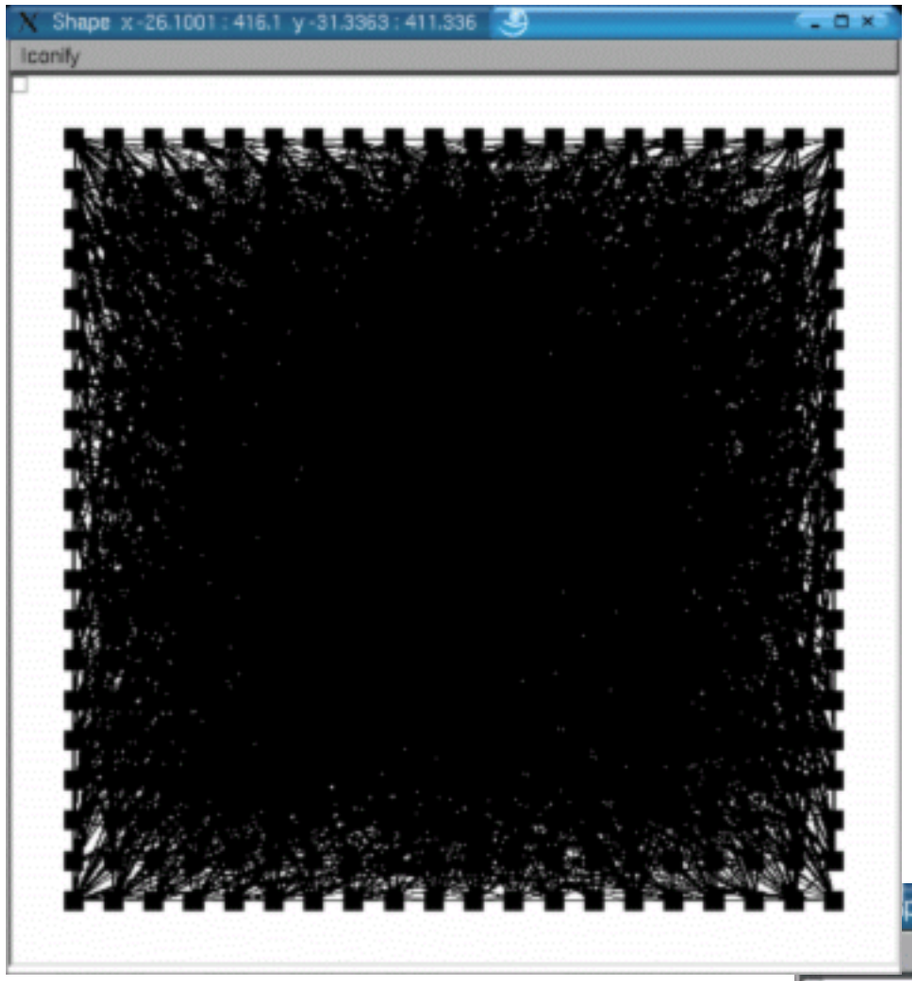
Feed synaptic input into network




```
objref StimTrigger, NetInput

create Presynapse

Presynapse {
  pt3dclear()
  pt3dadd(NETIN_SP_X, NETIN_SP_Y, NETIN_SP_Z, 10)
  pt3dadd(NETIN_SP_X+10, NETIN_SP_Y, NETIN_SP_Z, 10)

  StimTrigger = new NetStim(0.5)
  StimTrigger.start = NETIN_SP_START
  StimTrigger.interval = NETIN_SP_INTERVAL
  StimTrigger.number = NETIN_SP_NUMBER
  StimTrigger.noise = NETIN_SP_NOISE

  NetInput = new NetCon(StimTrigger,
        neuron[NETIN_SP_TARGET].IAF,
        0.5, 0, NETIN_SP_WEIGHT)
}
```

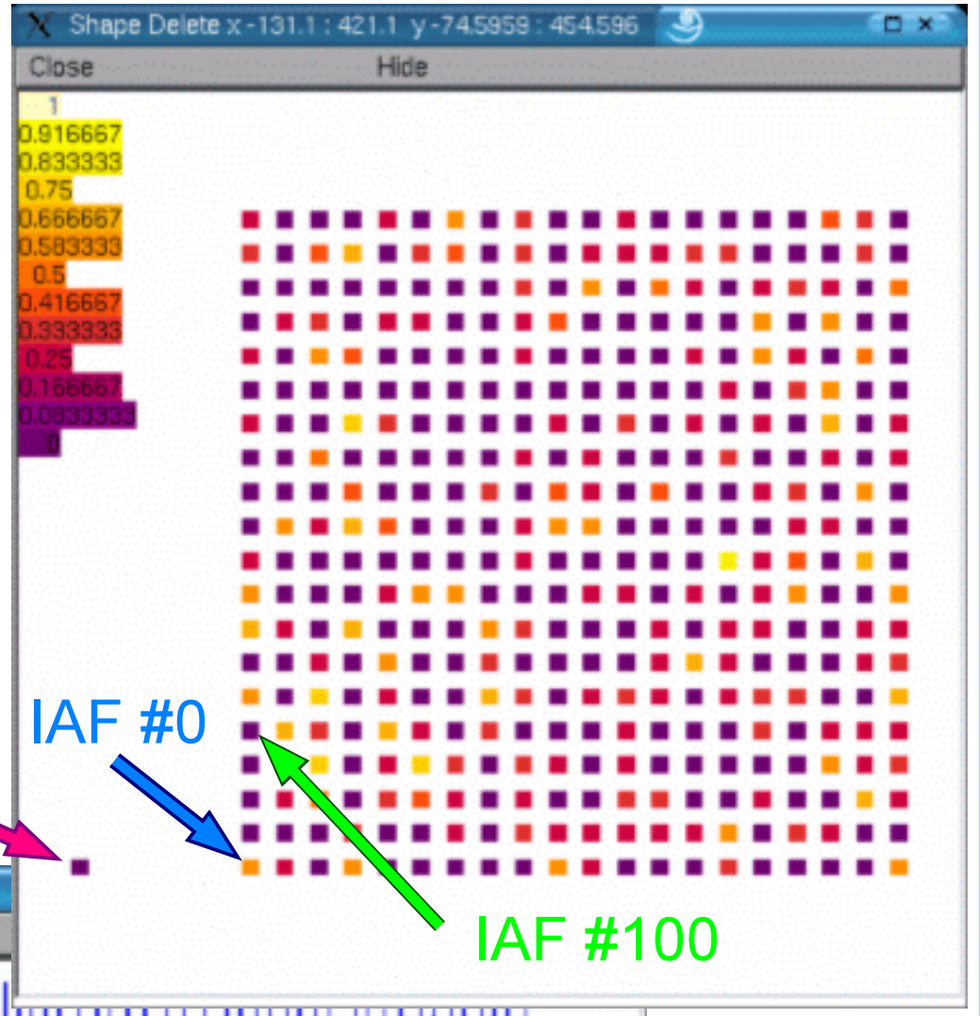
❶ A very (very!) simple example

Extension: Neural network of many (?) IAF neurons



shape plot

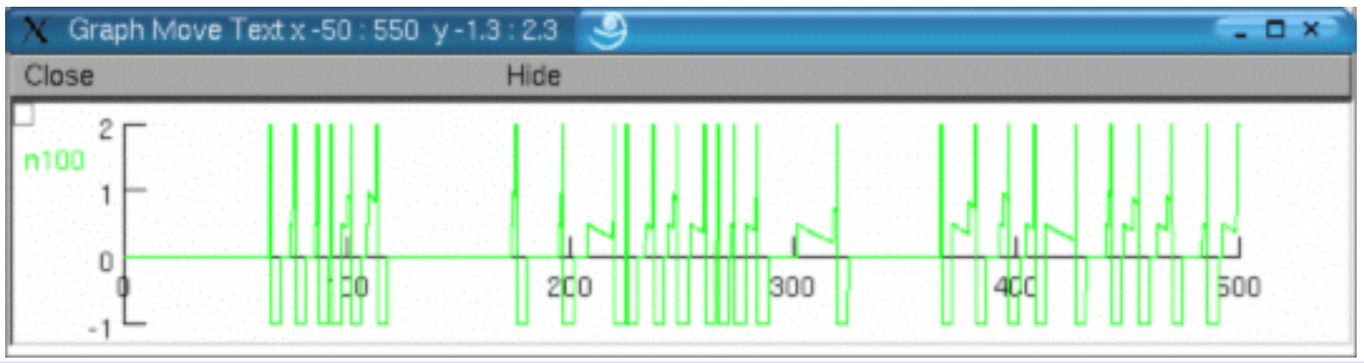
state of all IAF neurons



“presynapse”

state of IAF neuron #1

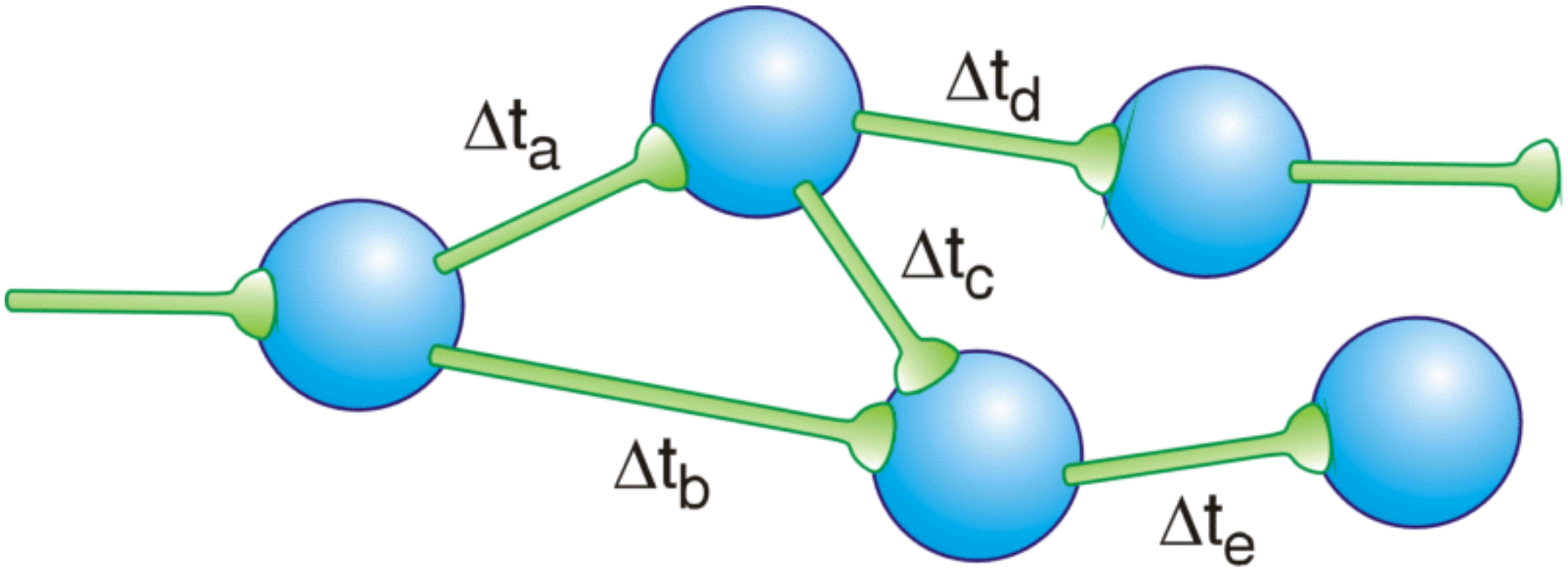
state of IAF neuron #100

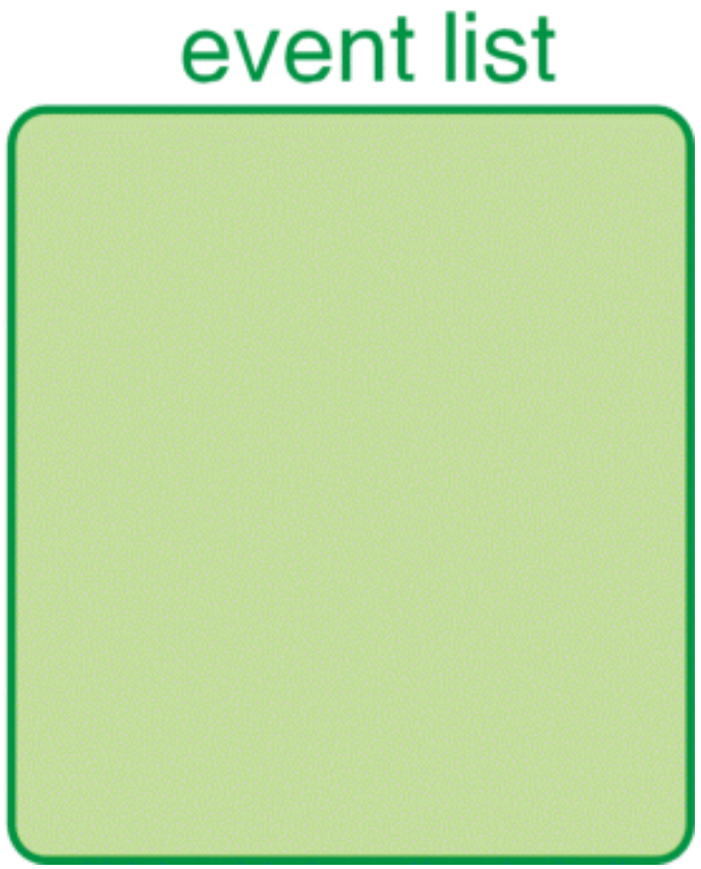


Outline

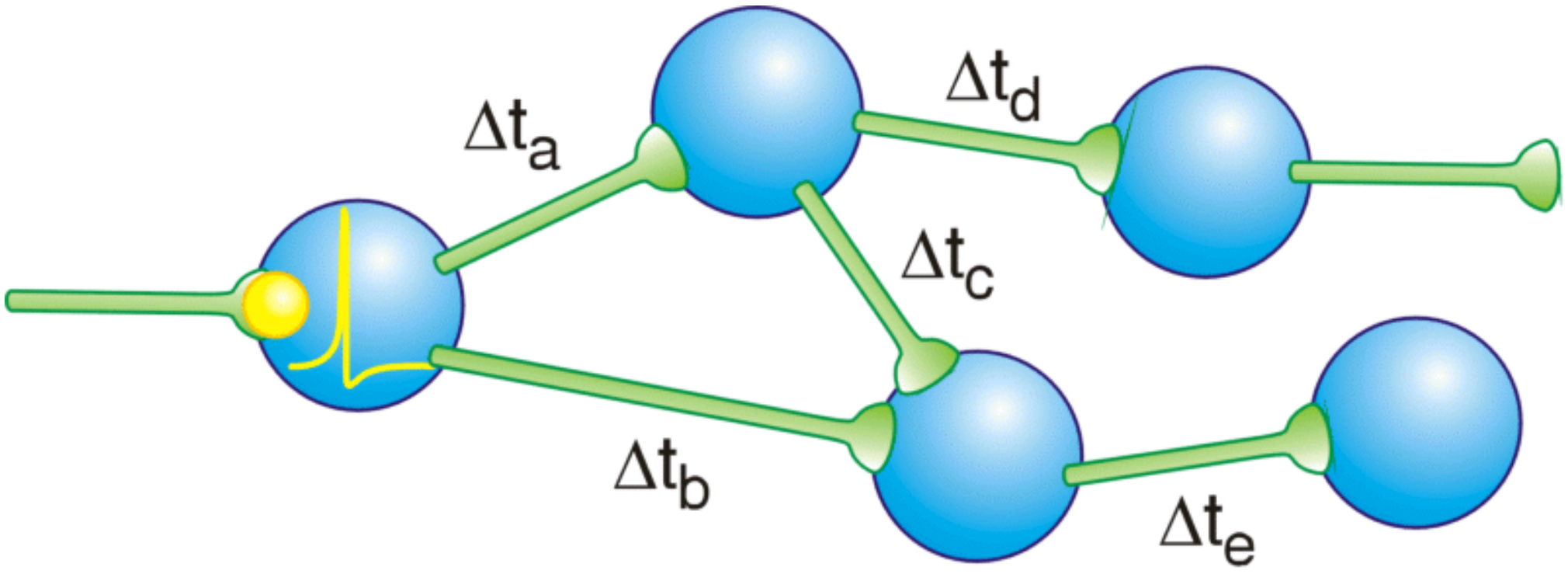
1. A very (very!) simple example
   (introducing NetCon, NetStim and templates)
2. Event-based approach to network modelling

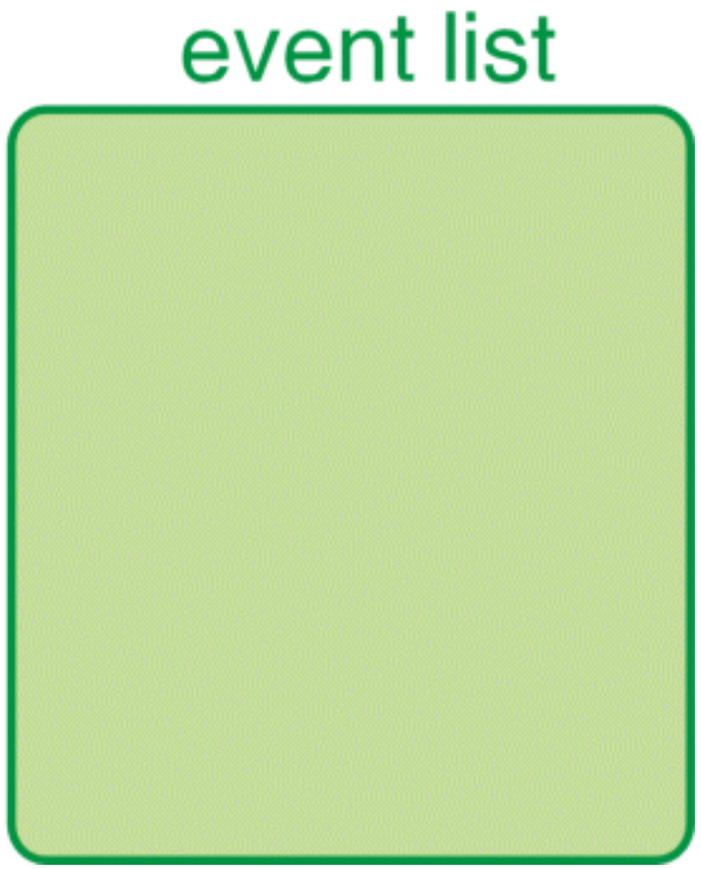
② Event-based approach to network modelling



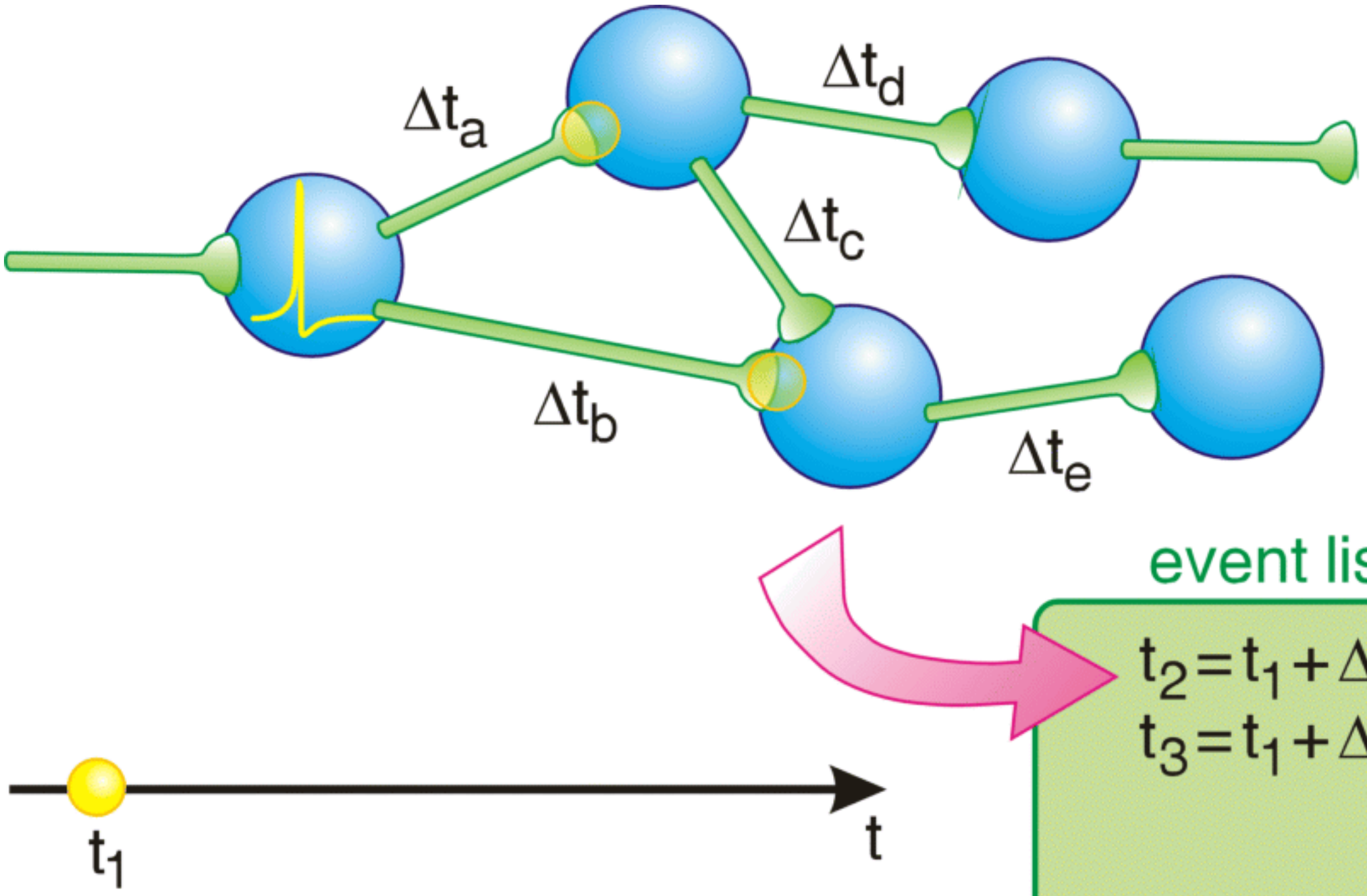


② Event-based approach to network modelling

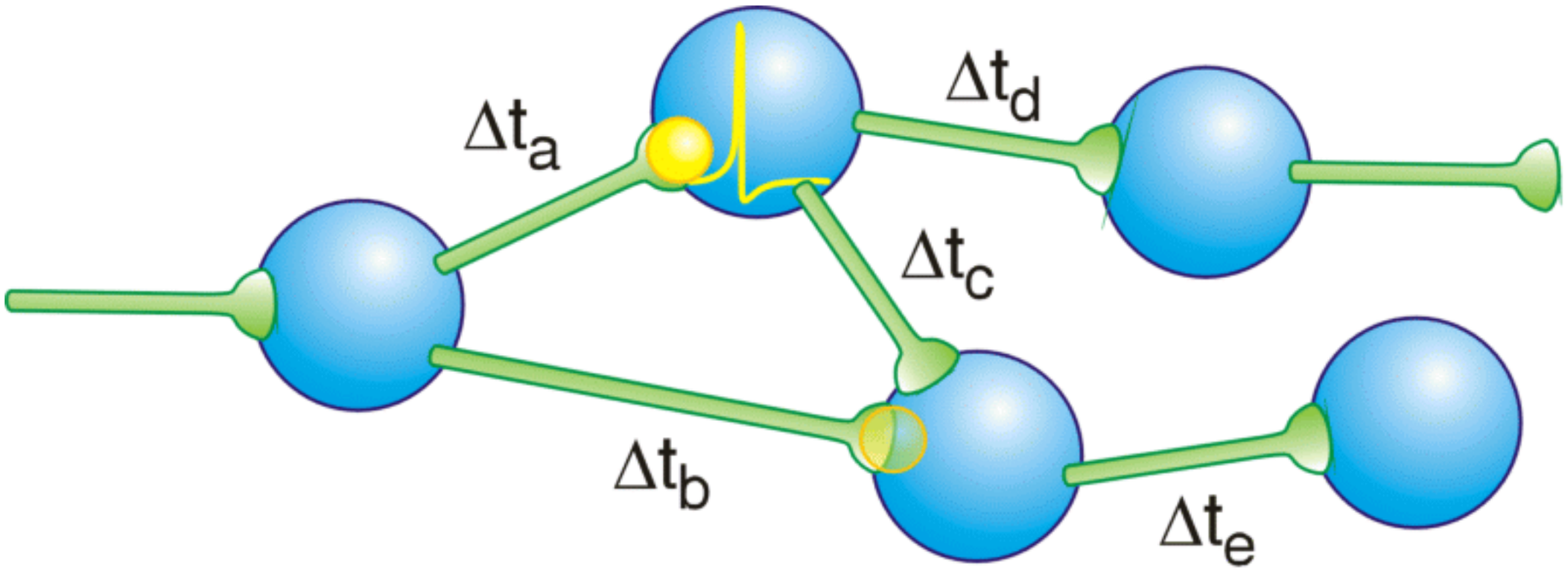


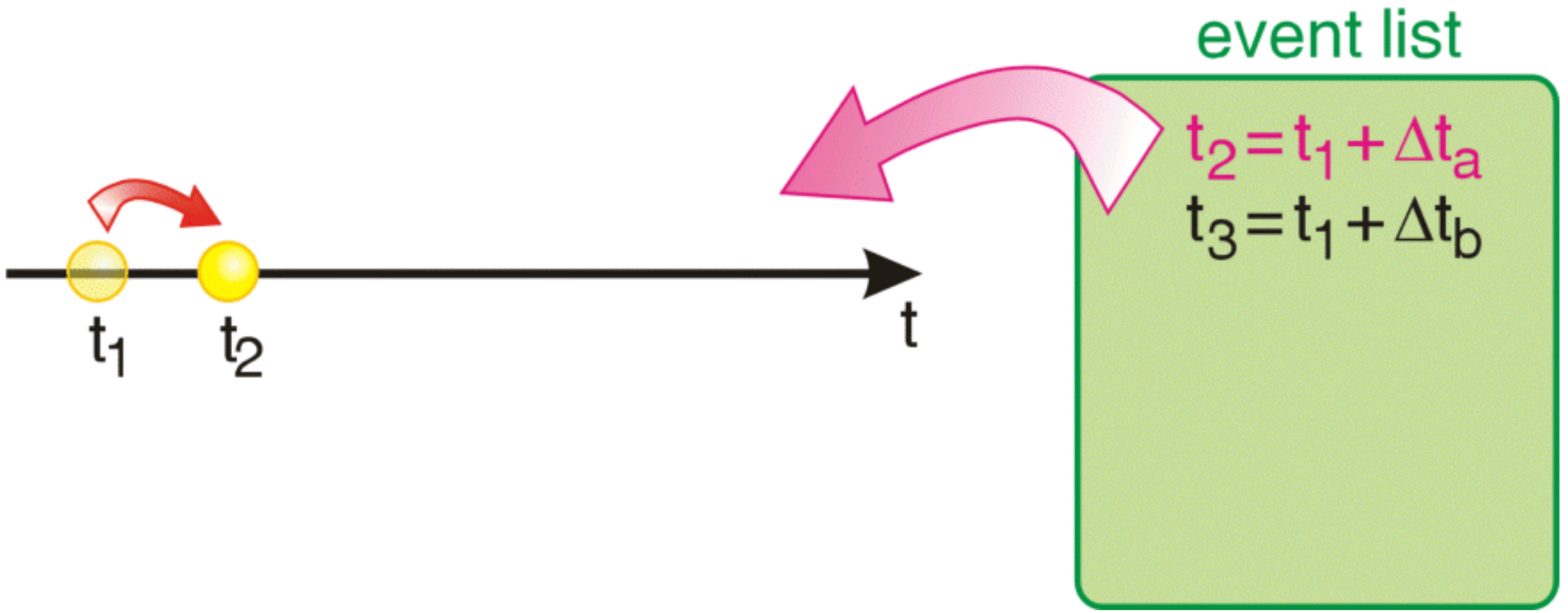


② Event-based approach to network modelling

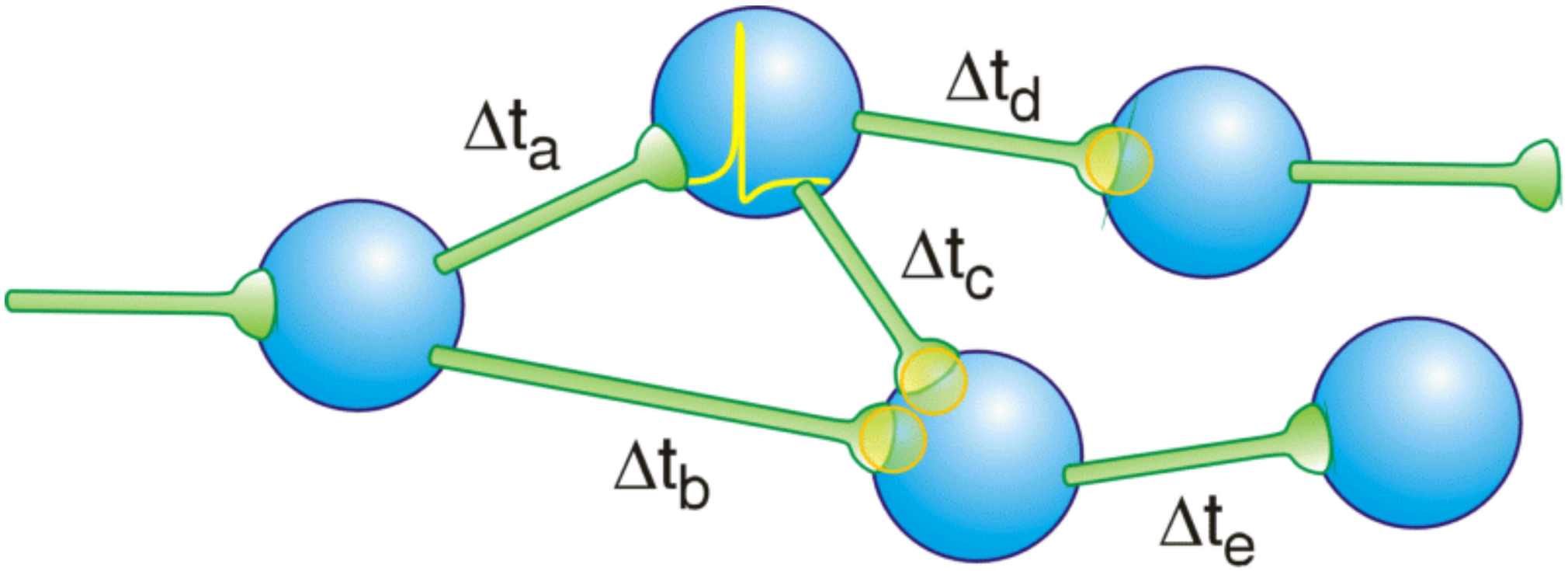


② Event-based approach to network modelling





② Event-based approach to network modelling



event list

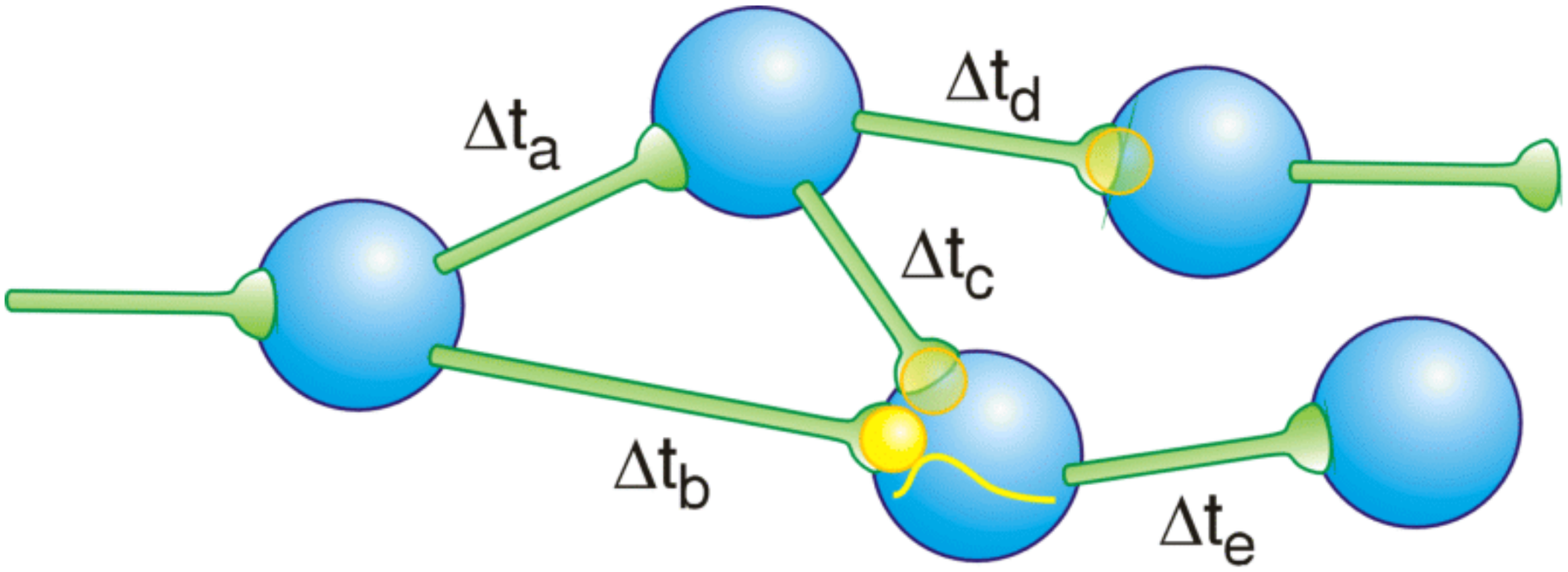
$t_2 = t_1 + \Delta t_a$

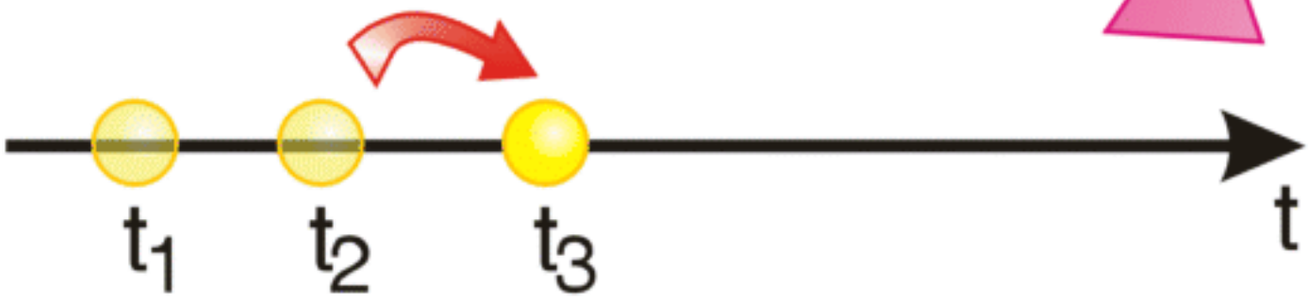
$t_3 = t_1 + \Delta t_b$

$t_4 = t_2 + \Delta t_c$

$t_5 = t_2 + \Delta t_d$

② Event-based approach to network modelling





event list

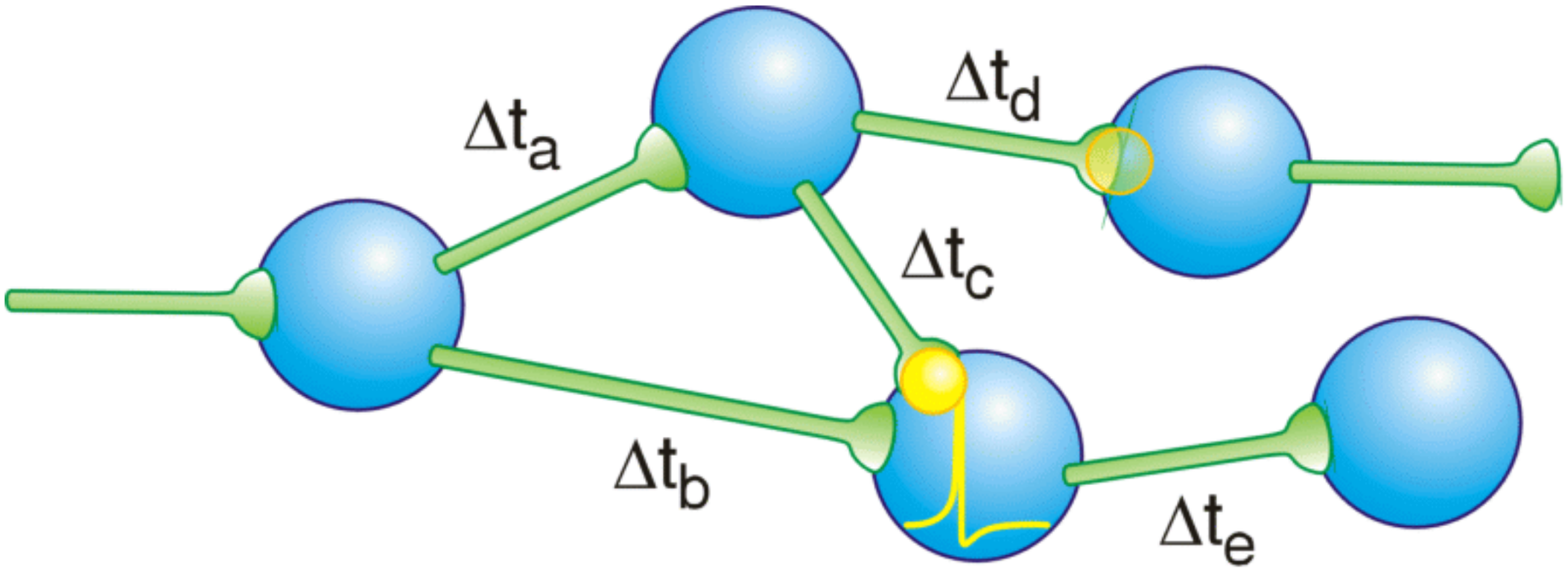
~~$t_2 = t_1 + \Delta t_a$~~

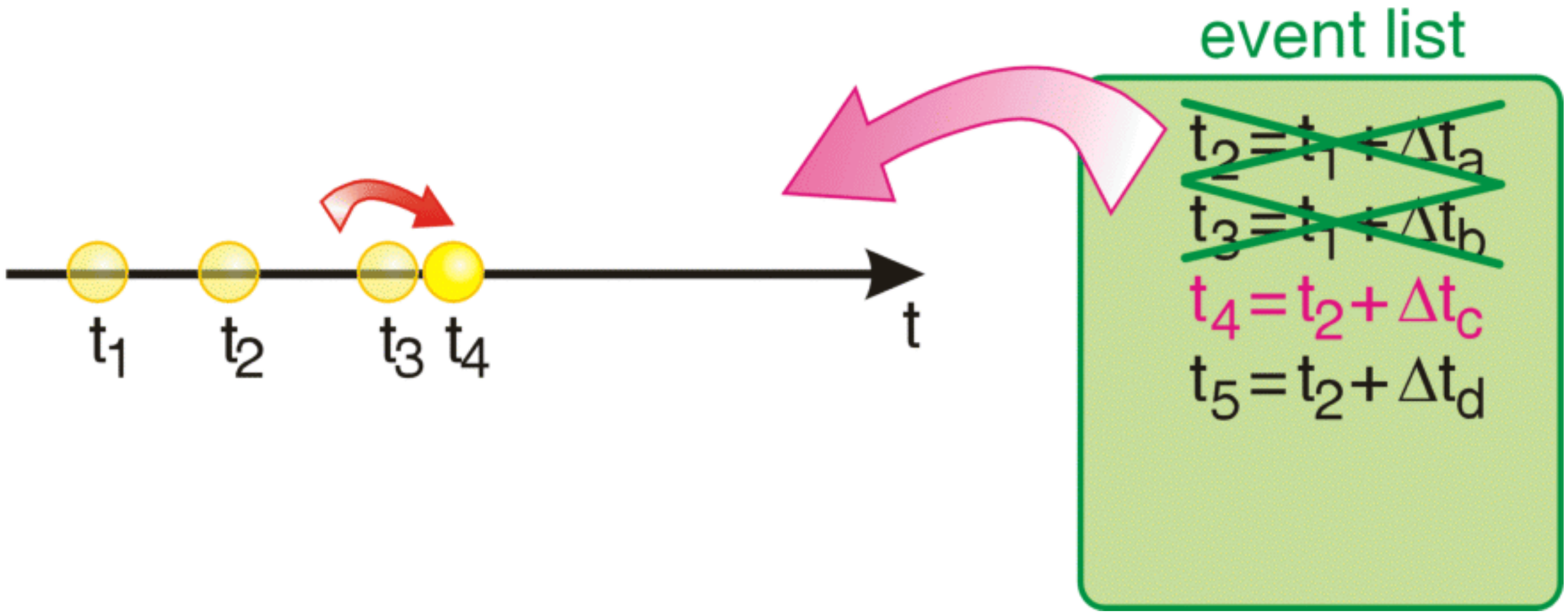
$t_3 = t_1 + \Delta t_b$

$t_4 = t_2 + \Delta t_c$

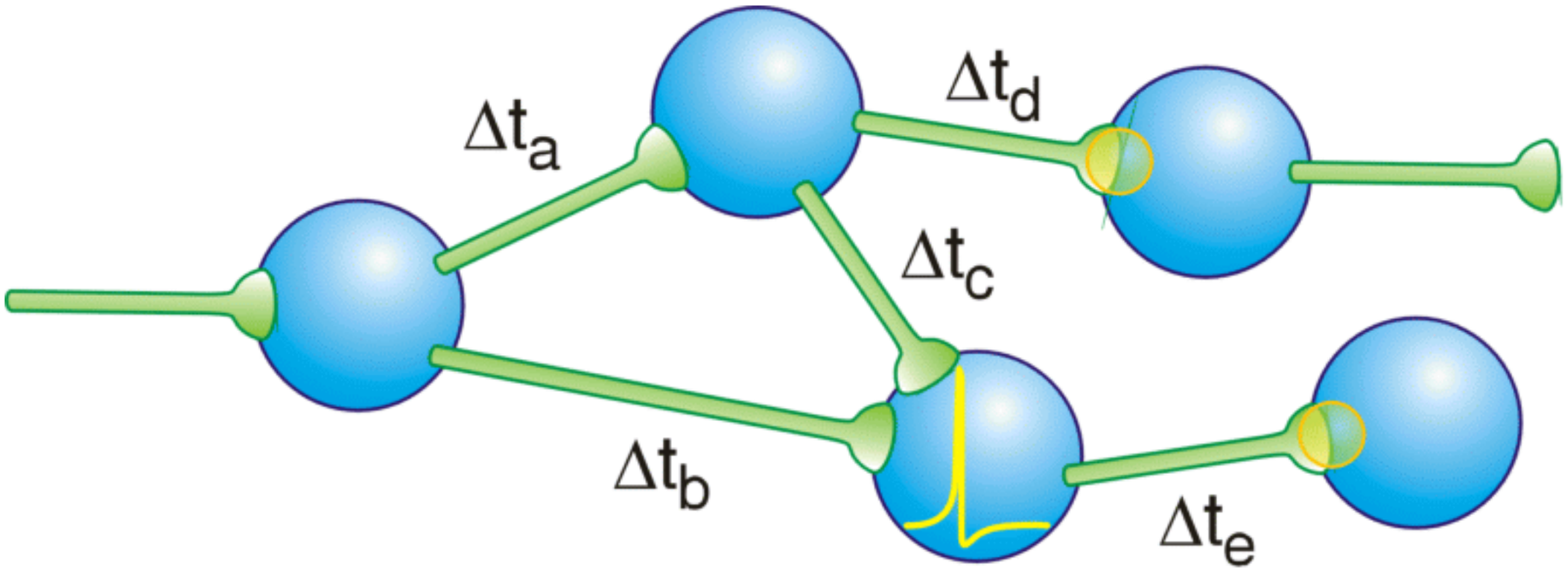
$t_5 = t_2 + \Delta t_d$

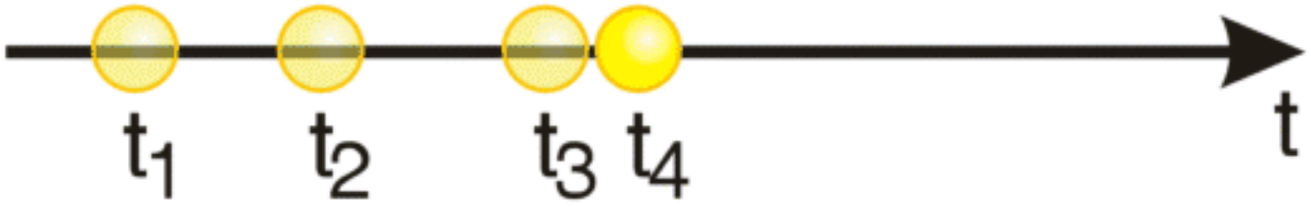
② Event-based approach to network modelling





② Event-based approach to network modelling





event list

$t_2 = t_1 + \Delta t_a$

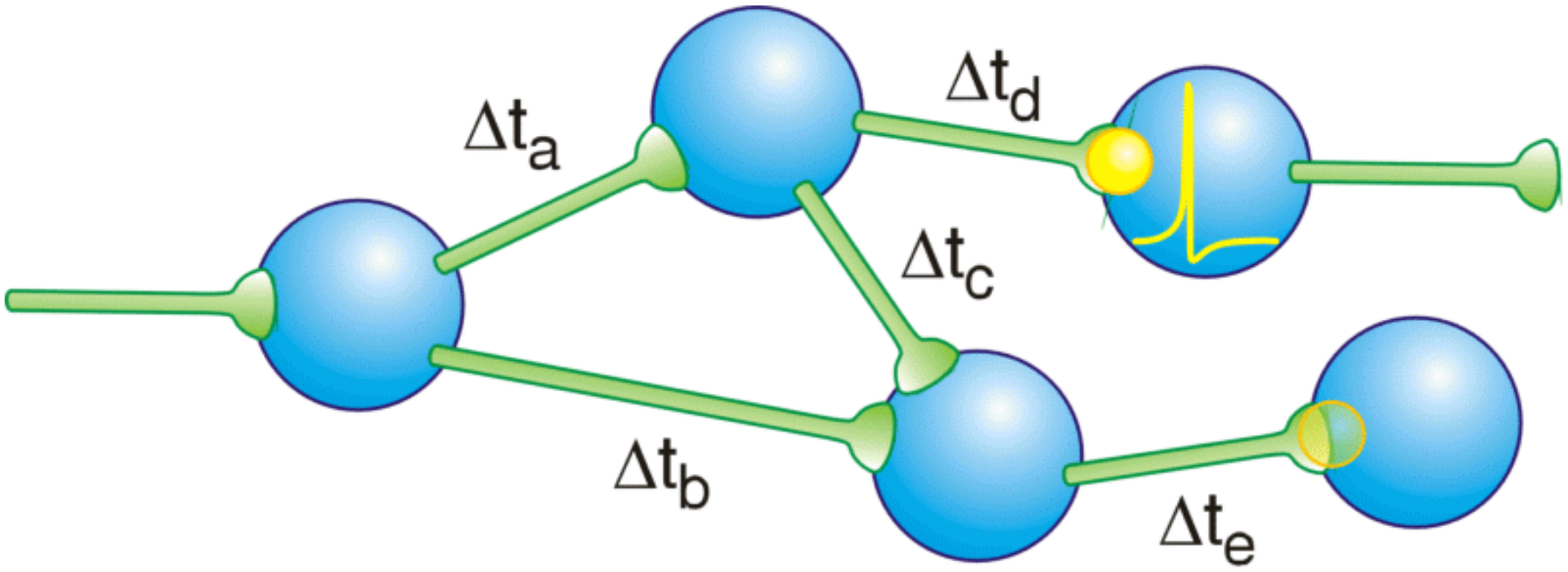
$t_3 = t_1 + \Delta t_b$

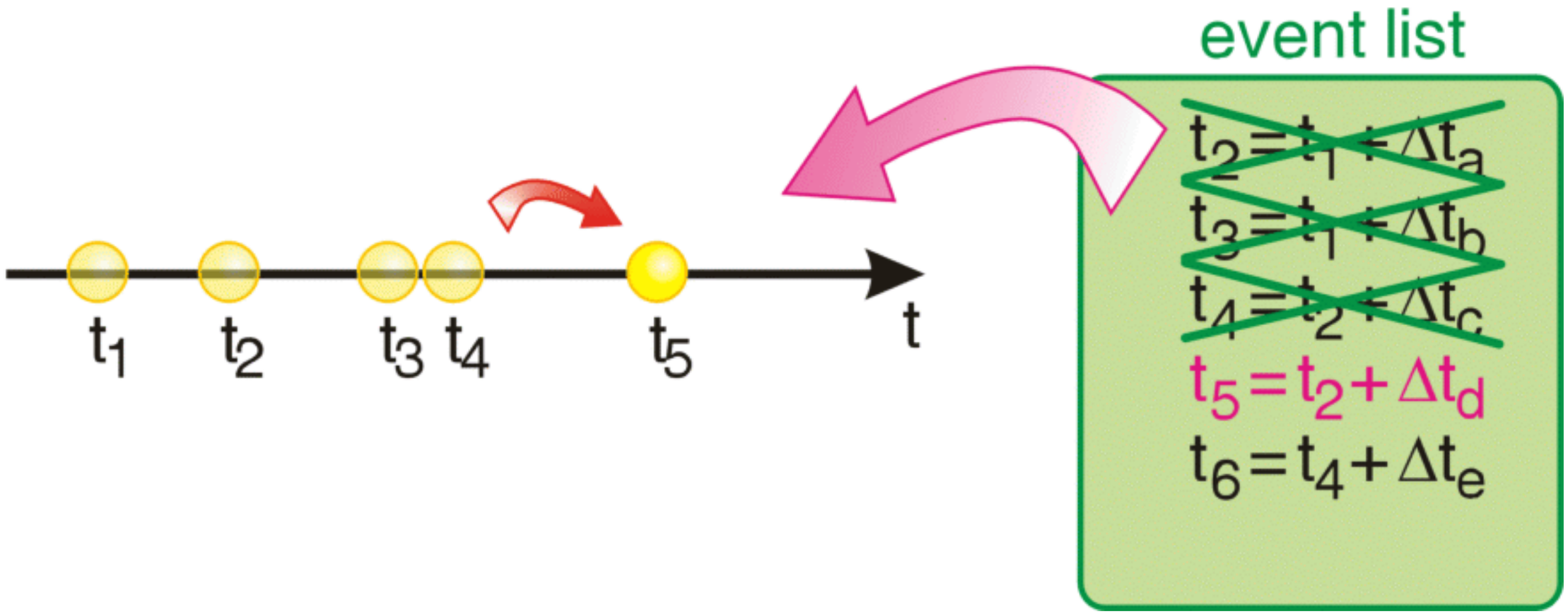
$t_4 = t_2 + \Delta t_c$

$t_5 = t_2 + \Delta t_d$

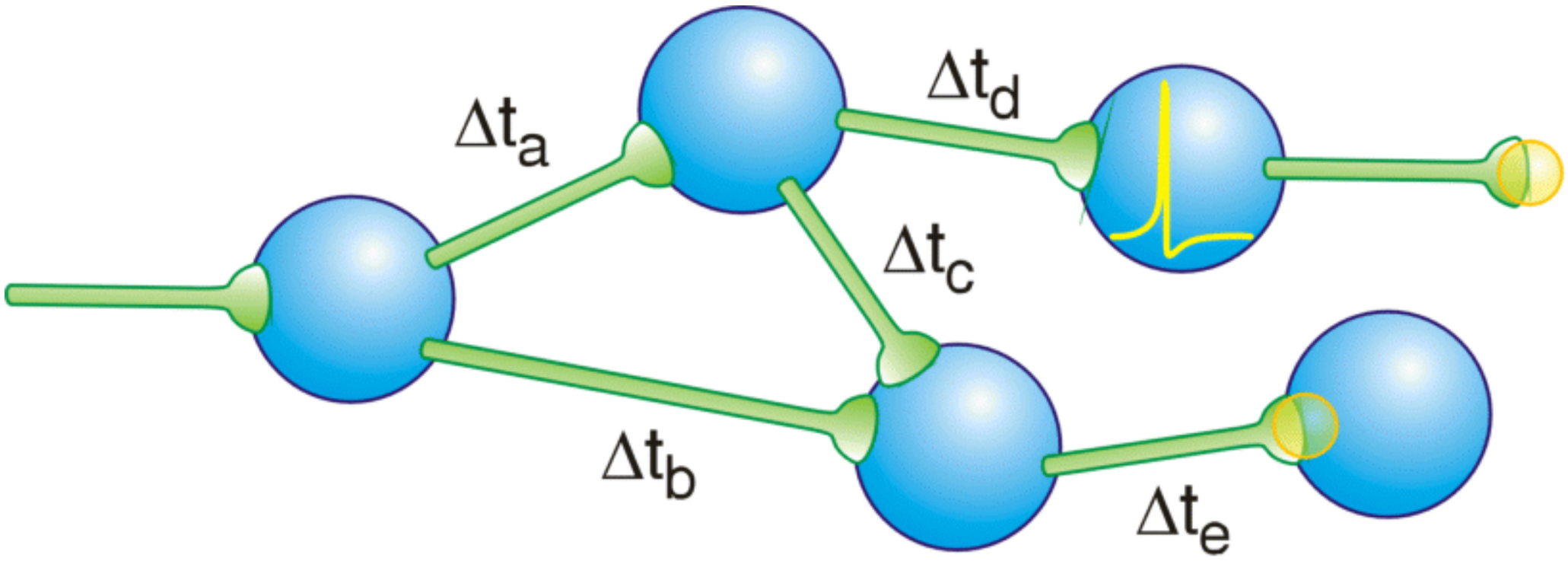
$t_6 = t_4 + \Delta t_e$

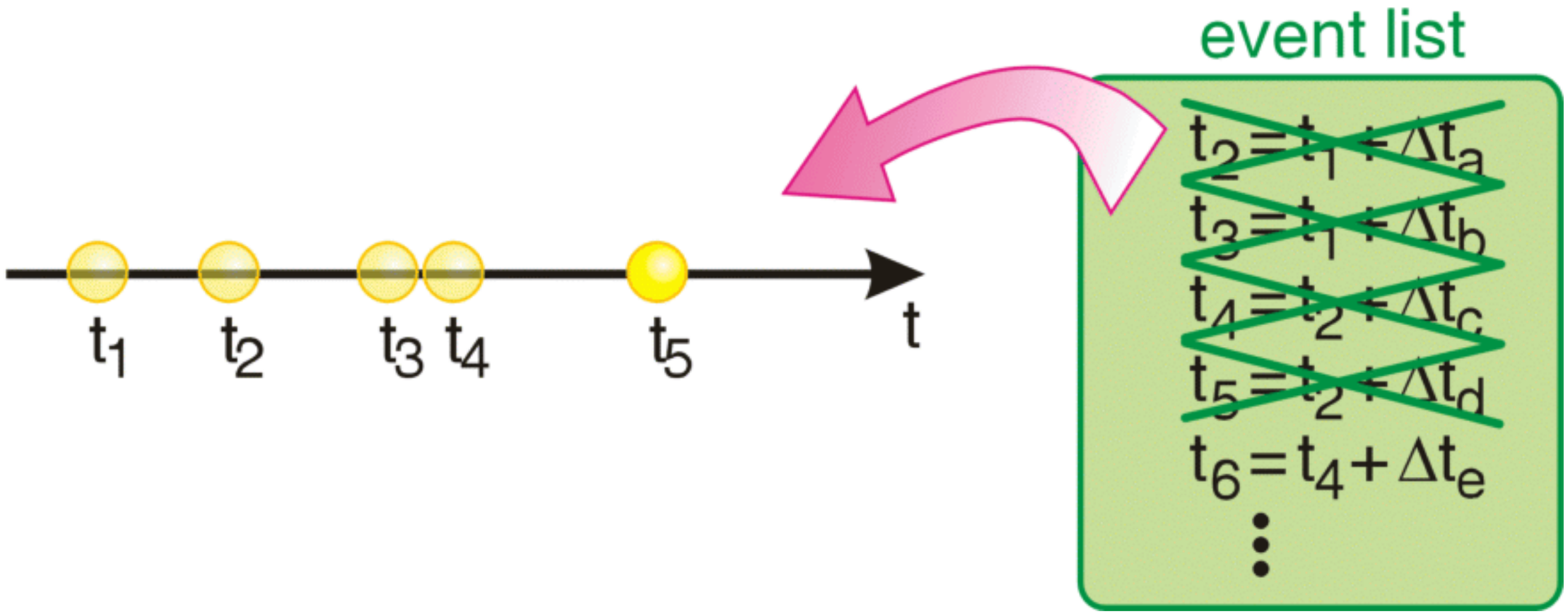
② Event-based approach to network modelling



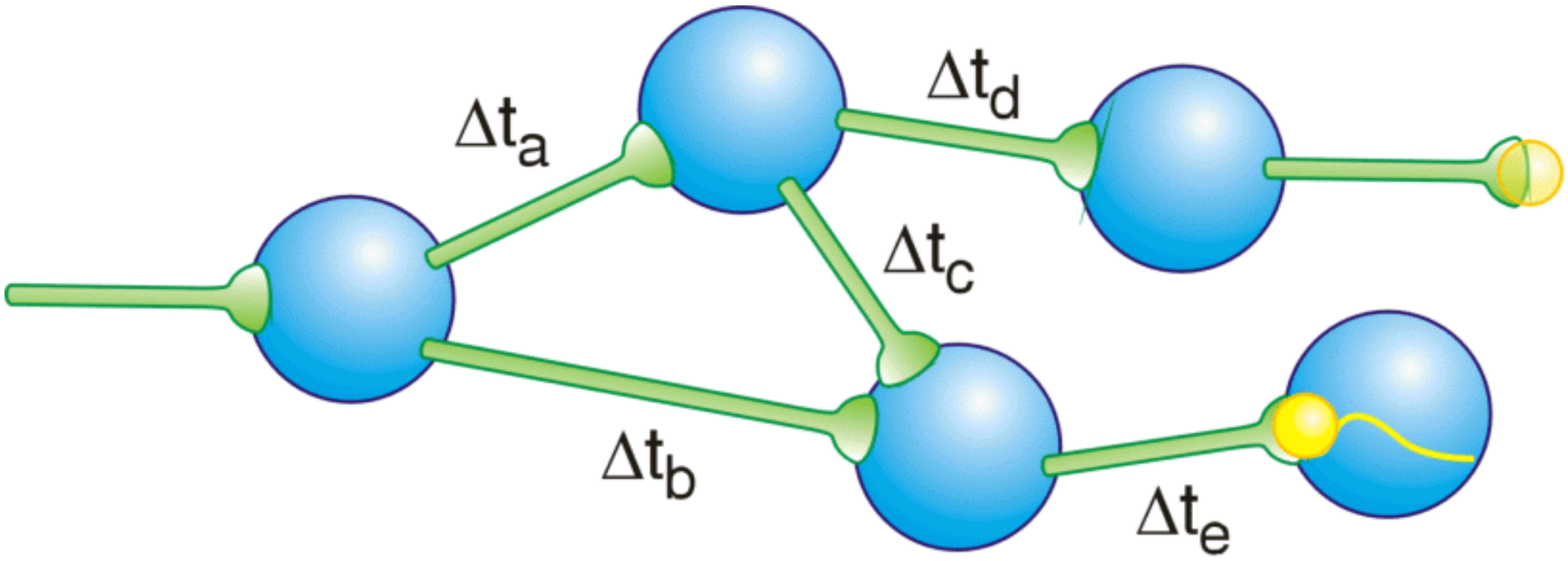


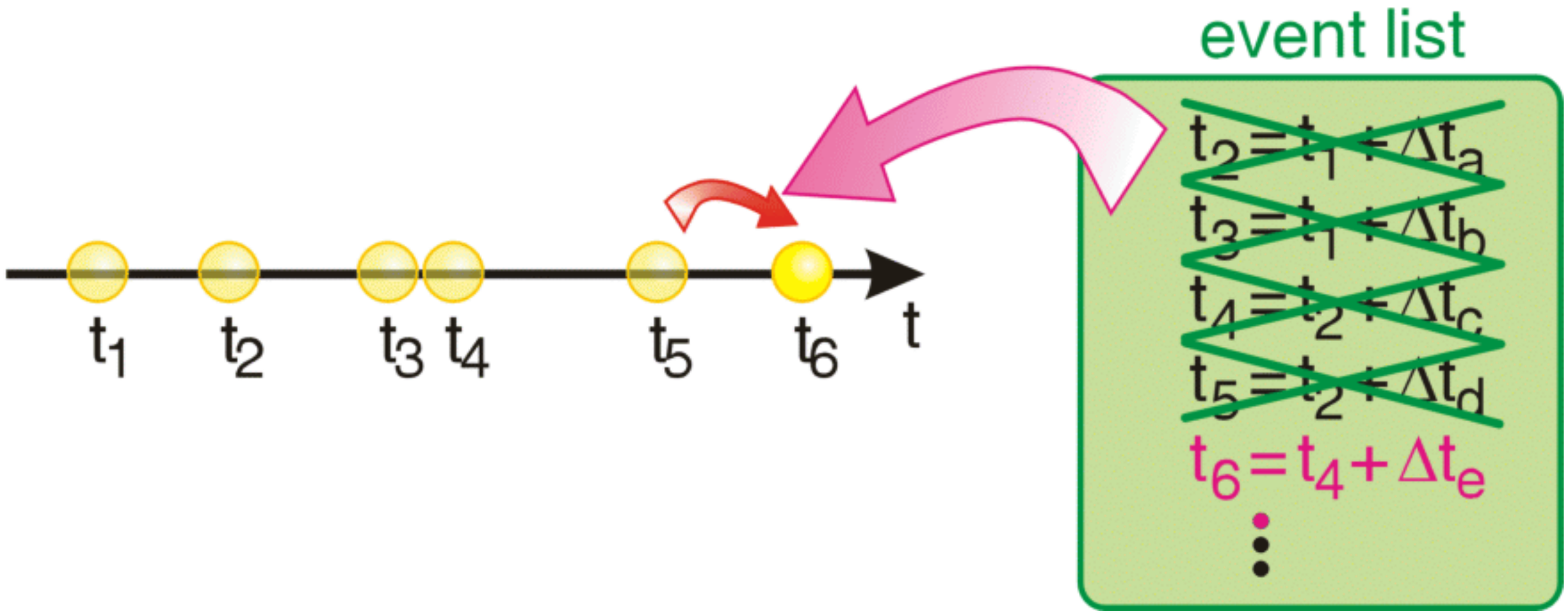
② Event-based approach to network modelling



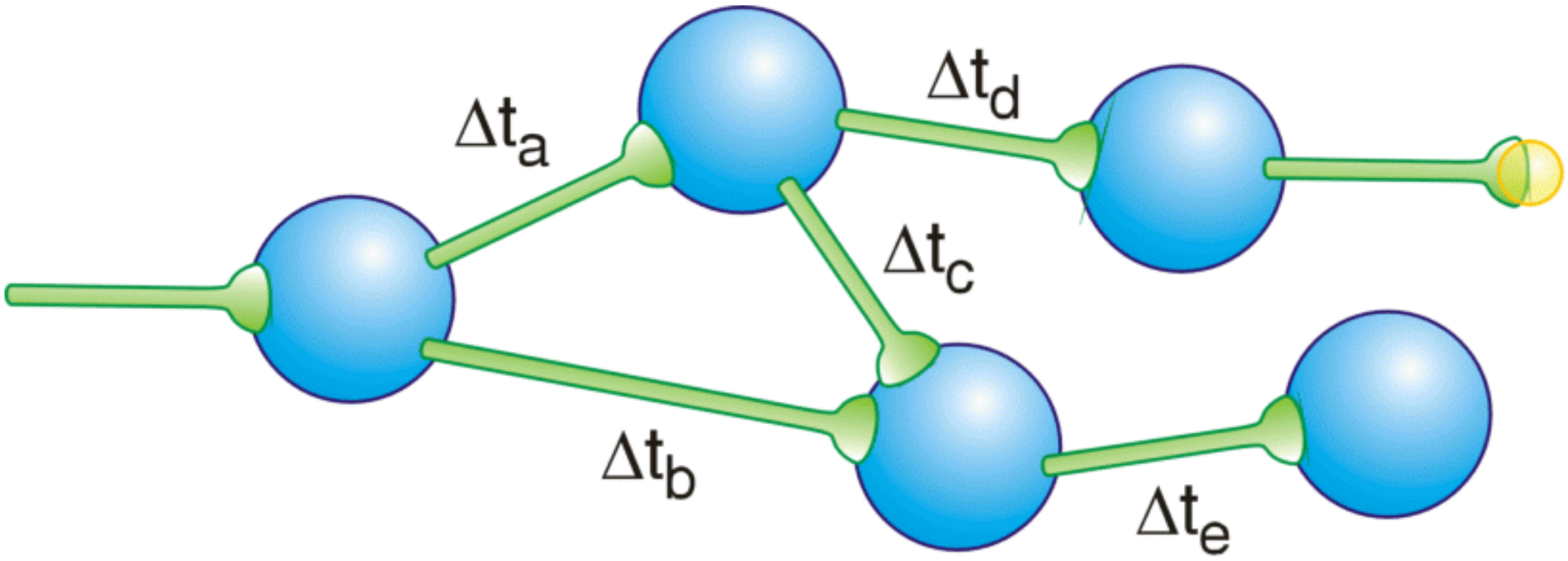


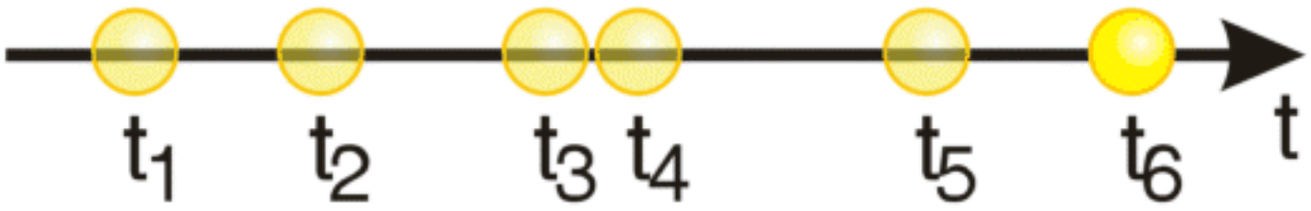
② Event-based approach to network modelling





② Event-based approach to network modelling

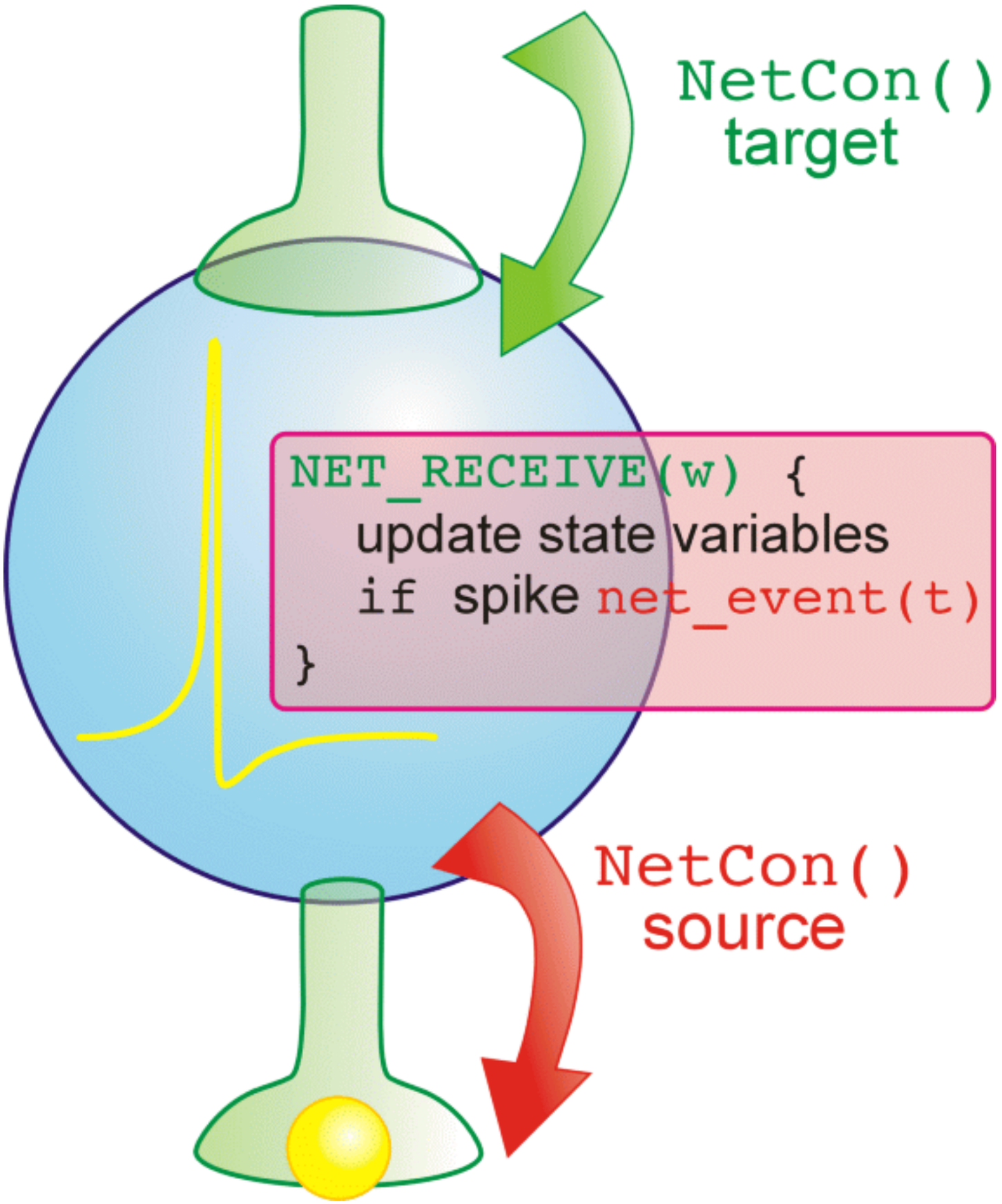




event list

$t_2 = t_1 + \Delta t_a$
$t_3 = t_1 + \Delta t_b$
$t_4 = t_2 + \Delta t_c$
$t_5 = t_2 + \Delta t_d$
$t_6 = t_4 + \Delta t_e$
⋮

❷ Event-based approach to network modelling



Example: **`IntFire1`** point process

```
NEURON {
  POINT_PROCESS IntFire1
  RANGE tau, m
}

PARAMETER {
  tau = 10 (ms)
}

ASSIGNED {
  m
  t0 (ms)
}

INITIAL {
  m = 0
  t0 = 0
}
```

```
NET_RECEIVE (w) {
  m = m*exp( -(t -t0)/tau)
  m = m + w
  t0 = t

  if (m >= 1) {
    net_event(t)
    m = 0
  }
}
```

❷ Event-based approach to network modelling

Example: **`IntFire1`** point process

```
NET_RECEIVE (w) {
    m = m*exp( -(t -t0)/tau)
    m = m + w
    t0 = t

    if (m >= 1) {
        net_event(t)
        m = 0
    }
}
```

executed only when `NetCon` delivers a new event; no BREAKPOINT or SOLVE block to be executed at every `dt`

calculate present state variable `m` analytically

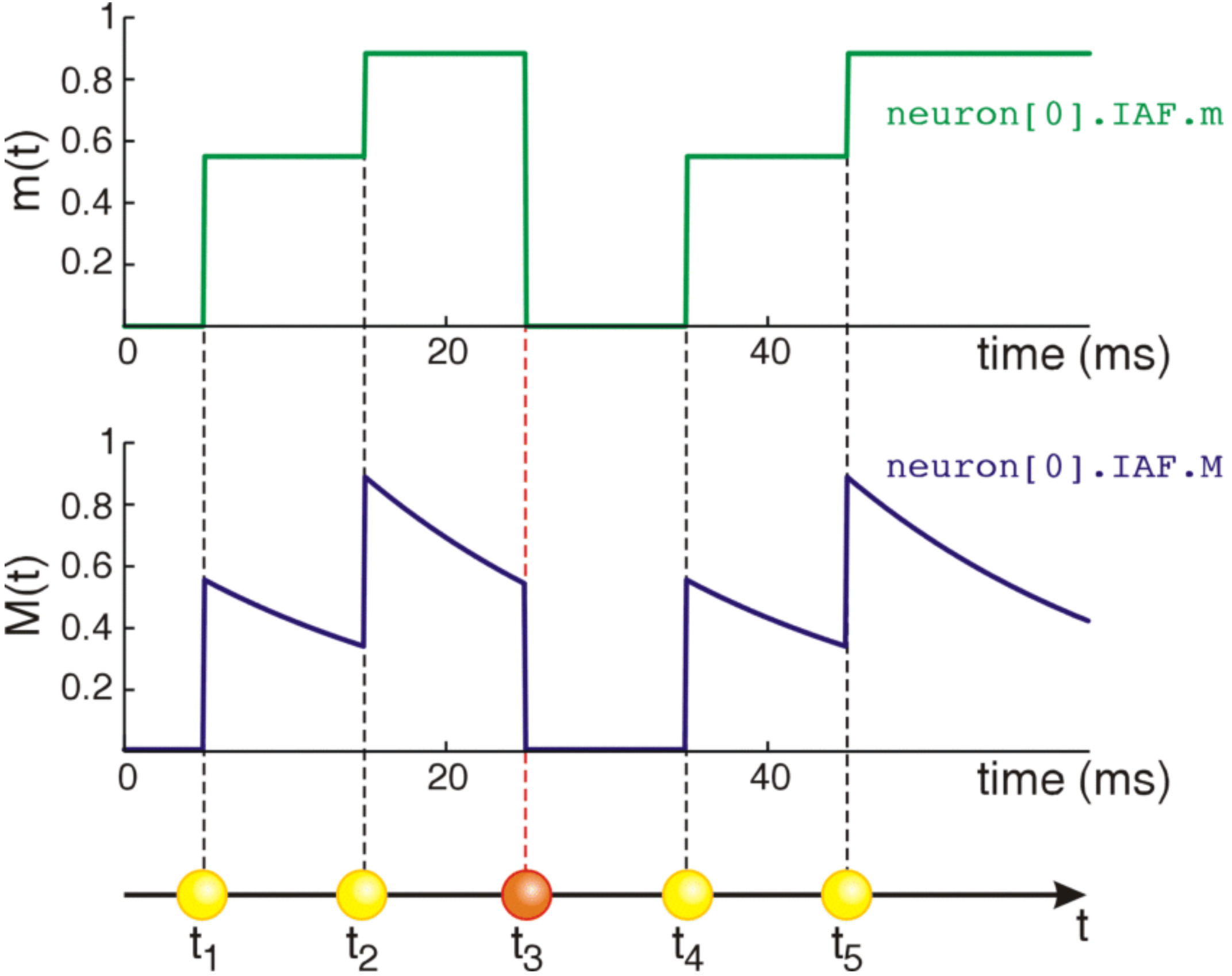
increment `m` by weight of event

check for threshold crossing

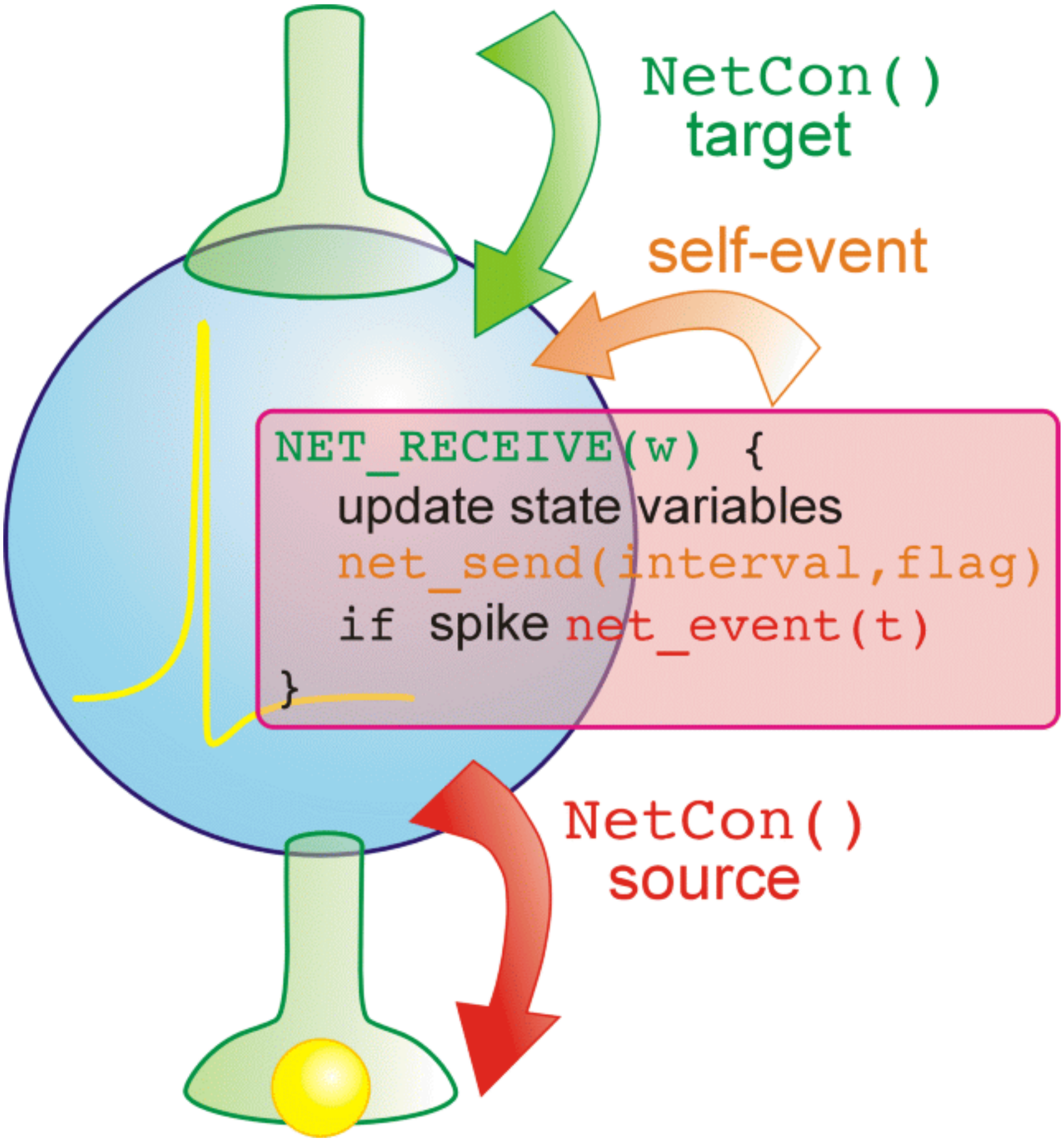
reset `m` to 0

Notify all `NetCon` objects for which this point process is a source that it fired a spike at time `t`

② Event-based approach to network modelling



② Event-based approach to network modelling



Example: **IntFire1** point process

```
NET_RECEIVE (w) {
   if (refractory == 0) {
      m = m*exp( -(t -t0)/tau)
      m = m + w
      t0 = t

      if (m >= 1) {
         net_event(t)
         refractory = 1
         net_send(refrac, refractory)
         m = 2
      }
   } else if (flag == 1) {
      refractory = 0
      m = 0
      t0 = t
   }
}
```

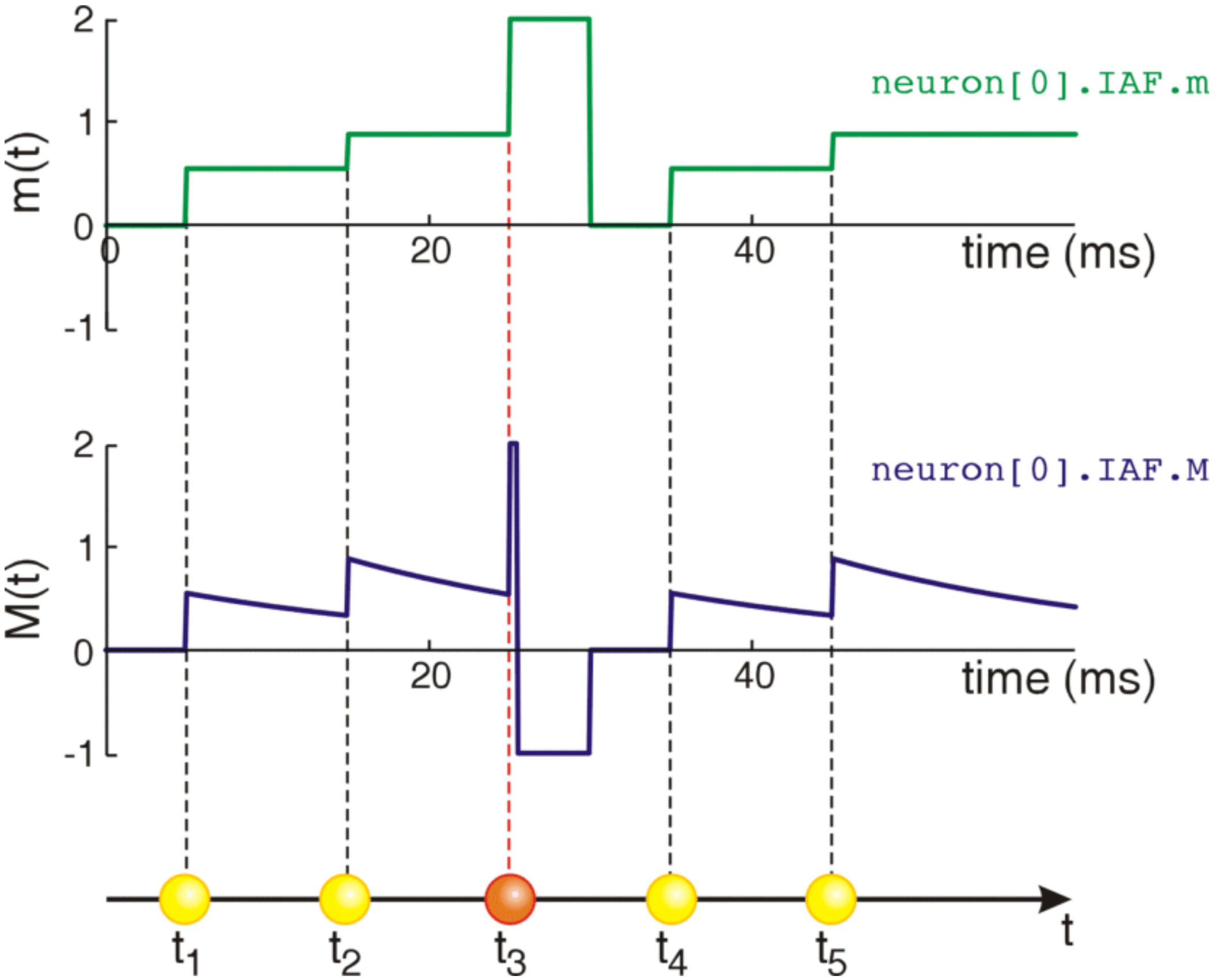
accept external events

issue a self-event that will arrive after `refrac` ms, tagged with `flag == 1`

ignore external event; accept “internal” event (self-event)

detect and respond to self-event

② Event-based approach to network modelling



❷ Event-based approach to network modelling

Example: **`MyOwnIAF`** point process

```
NEURON {
  ARTIFICIAL_CELL MyOwnIAF
  RANGE tau, m
}
```

define new point process

```
PARAMETER {
  tau = 10 (ms)
}

ASSIGNED {
  m
  t0 (ms)
}

INITIAL {
  m = 0
  t0 = 0
}
```

usage on hoc level:

```
objref IAF
IAF = new MyOwnIAF(0.5)
```

analytic cell model

```
NET_RECEIVE (w) {
  m = f[t-t0, tau]
  m = m + g[m,w]
  t0 = t

  if (m >= 1) {
    net_event(t)
    m = 0
  }
}
```