

Bezvadu sensoru tīklu virtuālās mašīnas

Ģirts Strazdiņš

Kurss “Bezvadu sensoru tīkli” [B]

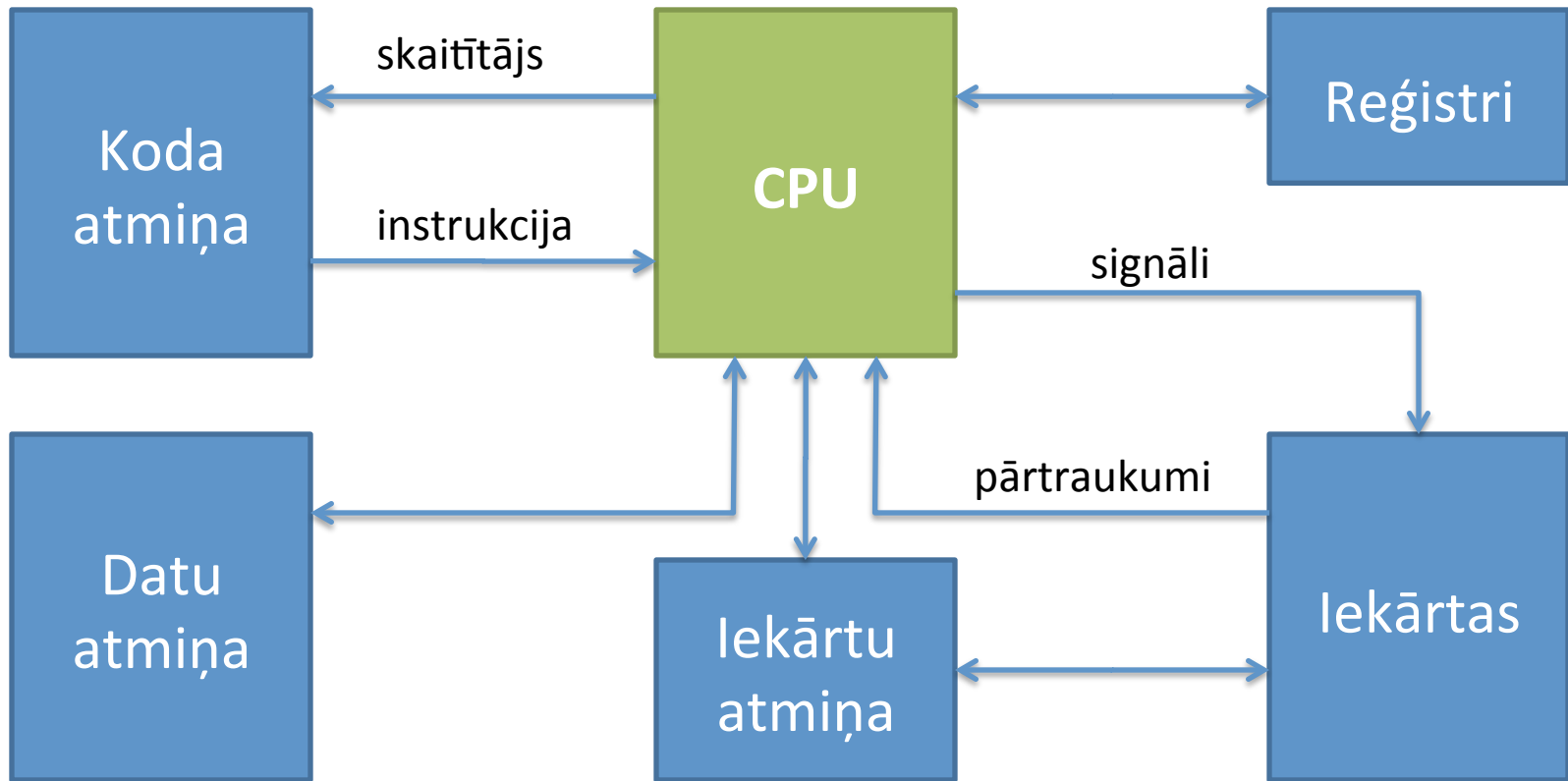
Datorikas fakultāte

Latvijas Universitāte

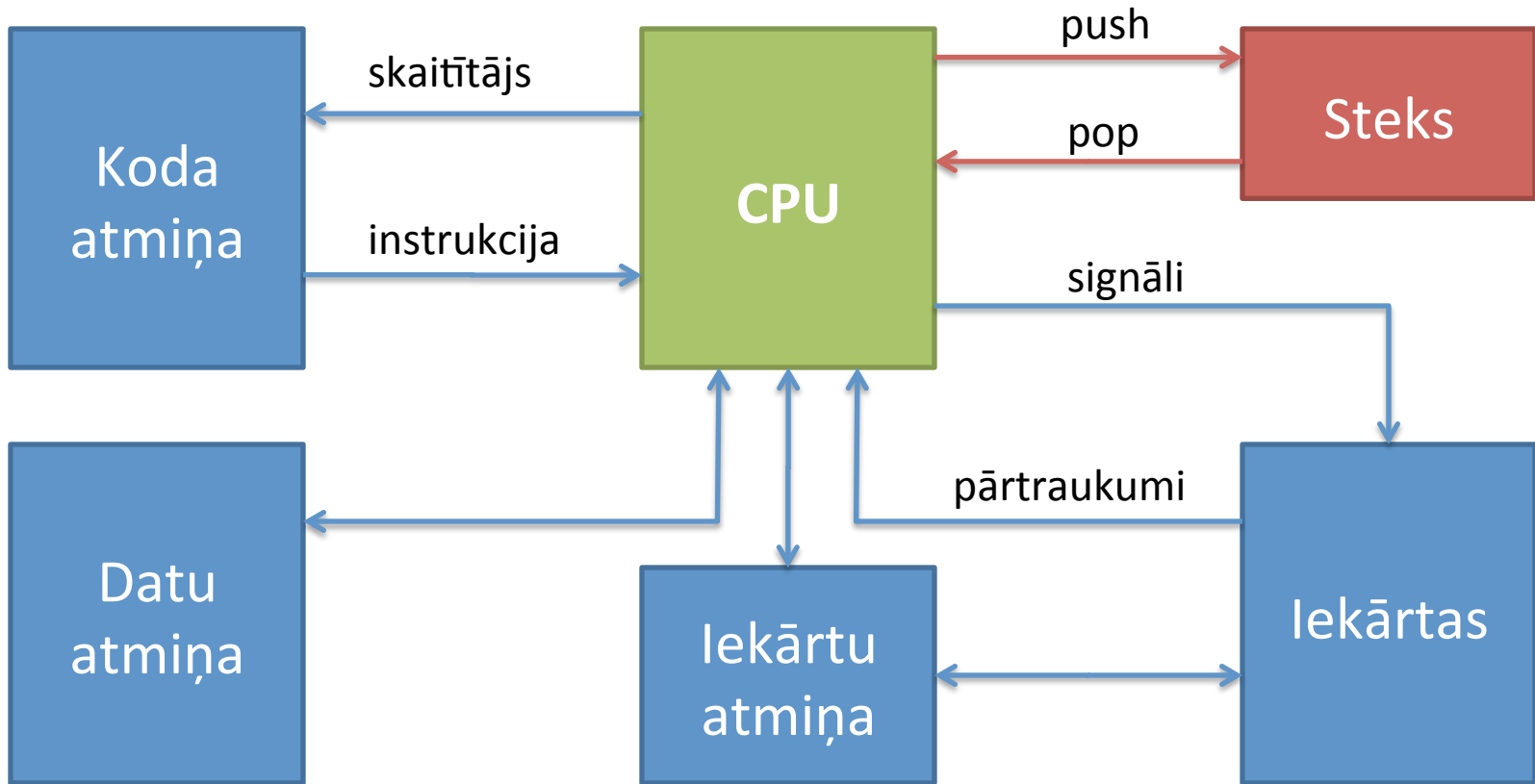
30.nov.2011.

Kas ir reāla mašīna (dators)?

Kas ir reāla mašīna (dators)?



Steka mašīna



Kā steks aizvieto reģistrus

- Ielādēt 2+3 atmiņā ar adresi 100:

Reģistru pieeja:

```
set r1, 2
add r1, 3
store r1, [100]
```

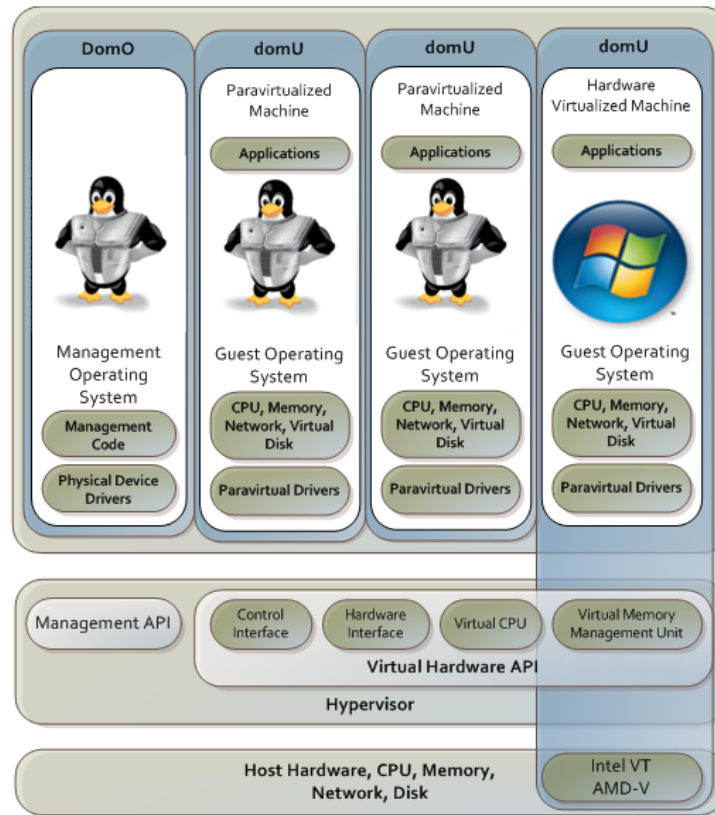
Steka pieeja:

```
push 2
push 3
add
store [100]
```

Kas labāks?

- Reģistriem: mazāk instrukciju
- Stekam: īsākas instrukcijas
- Kopsummā steka mašīnām programmas īsākas

Kur izmanto virtuālās mašīnas?

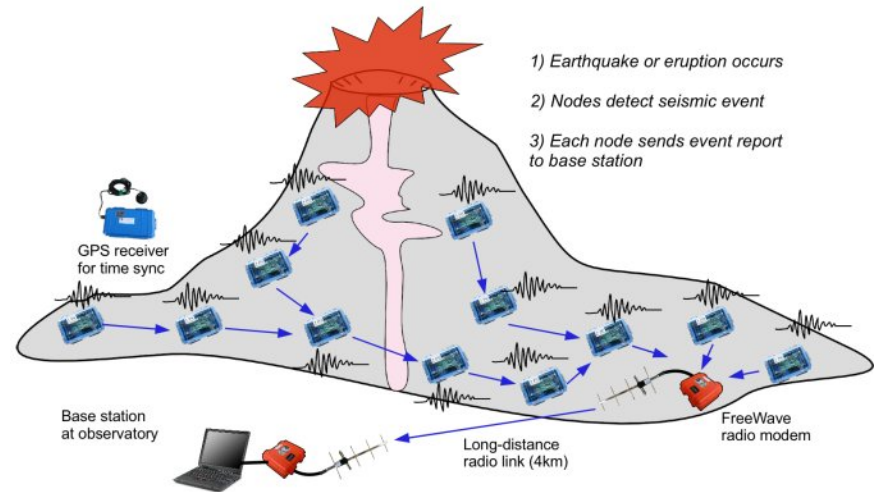


1 reāls dators = n virtuāli datori

Kāds BST labums no VM?

BST motivācija

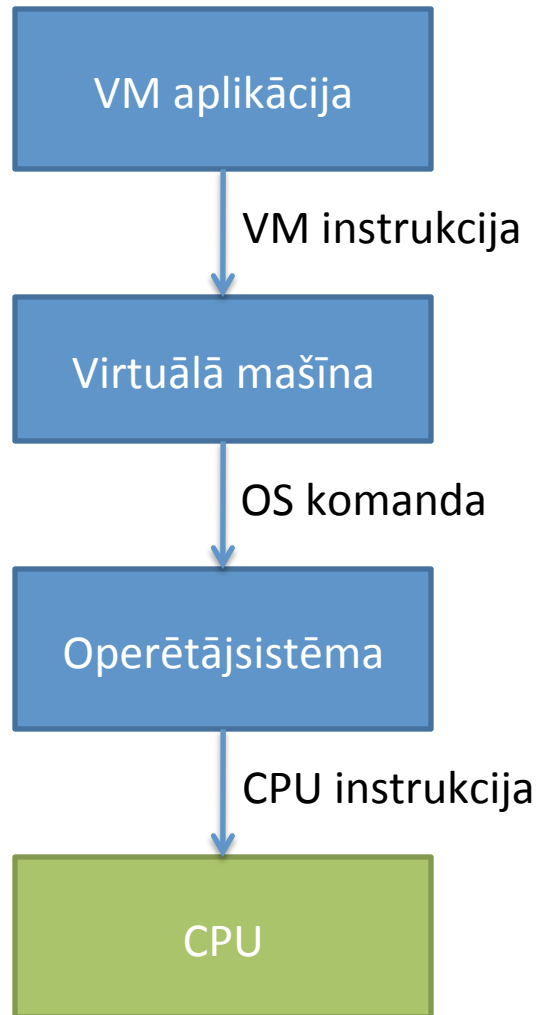
- Motes ir jā(pār)programmē
- Gribas:
 - Ātri
 - Droši
 - Plašas iespējas



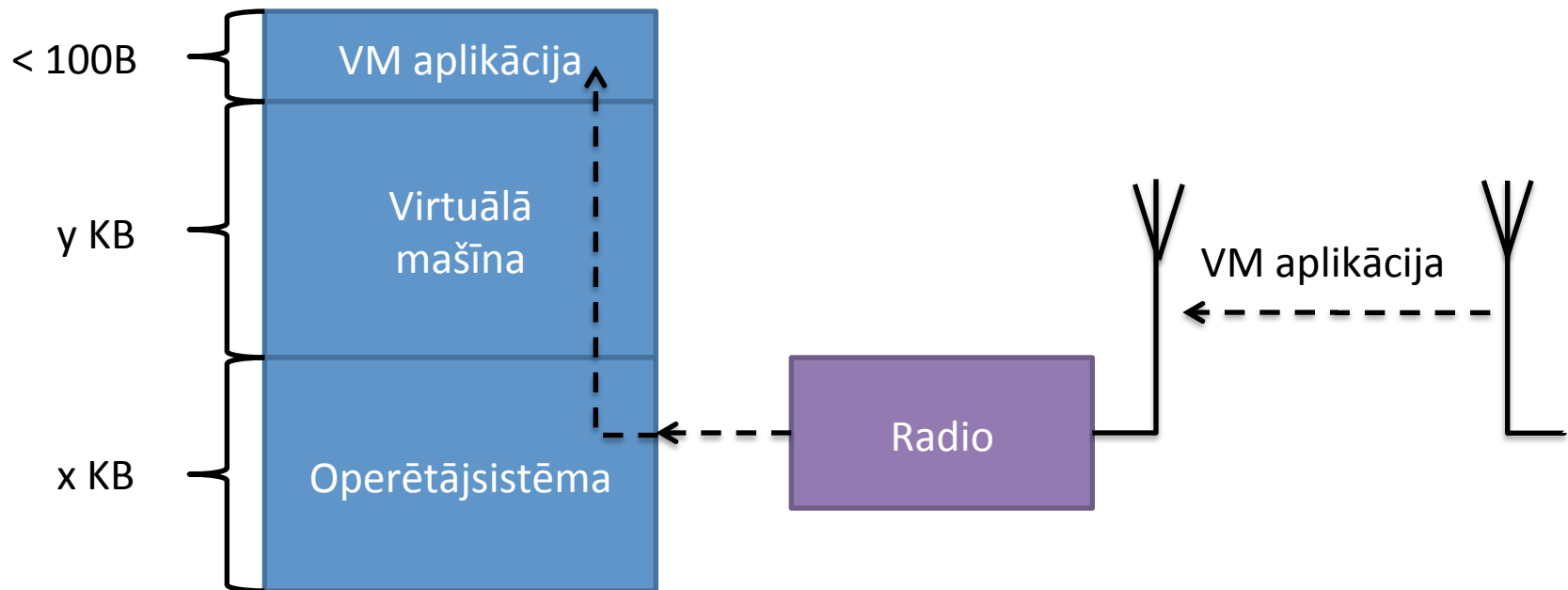
<http://fiji.eecs.harvard.edu/Volcano>

- Risinājums: samazināt koda izmēru

Kā darbojas VM uz motes?



Motes pārprogrammēšana



- VM aplikācija izmēros stipri mazāka par OS un VM kodu

Mate virtuālā mašīna

- UC Berkeley, 2002.g
- Komunikācijas centrēta VM sensoru tīkliem
- Mazas programmas
- Ērta programmu “virusāla” izplatīšana

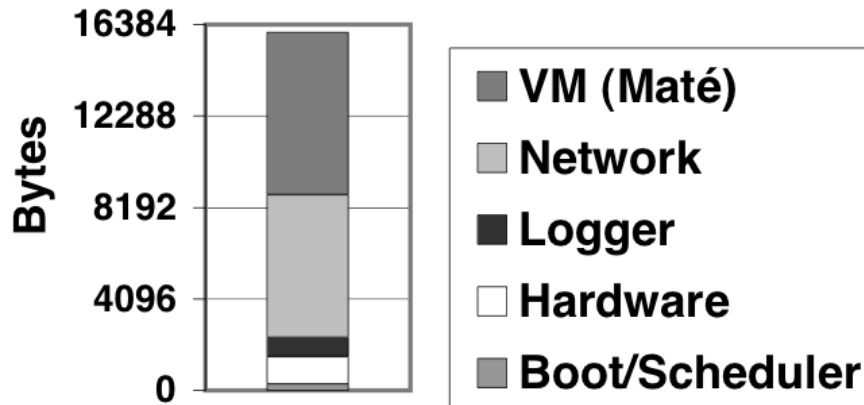


Mate tēja

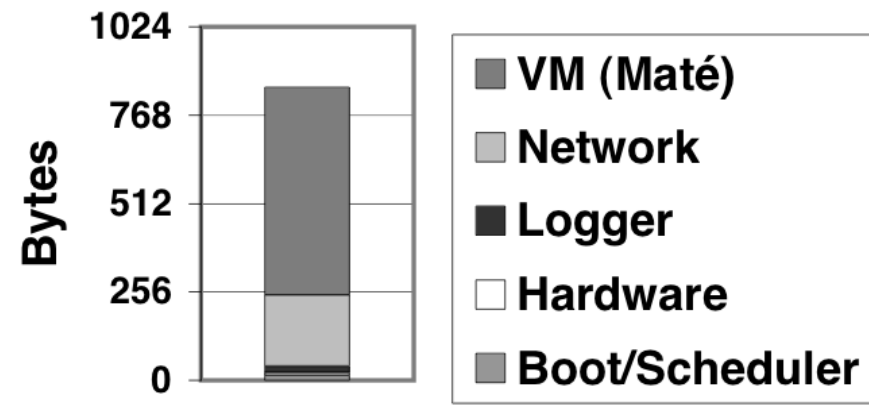
<http://www.sciencedaily.com/releases/2007/10/071023163949.htm>

Mate aplikācijas izmēri

Maté Code Footprint

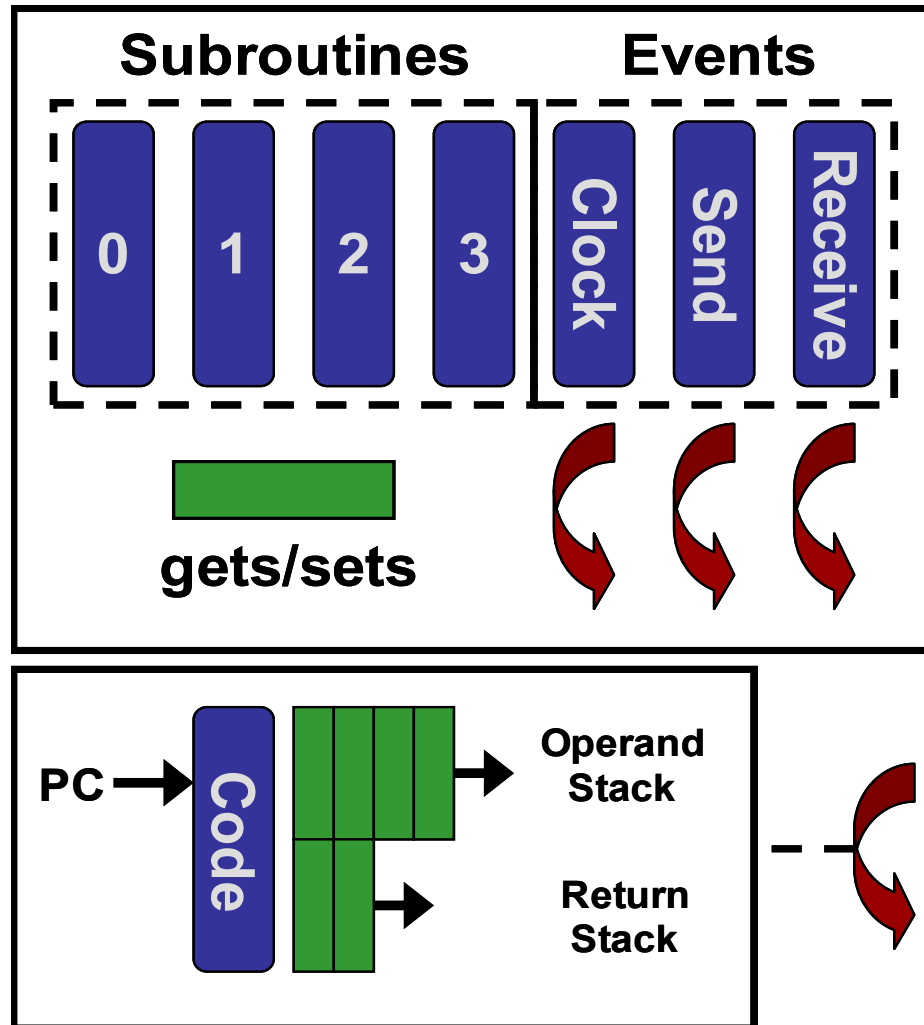


Maté Data Footprint



Component	Code (bytes)	Data (bytes)
VM (Maté)	7286	603
Network	6410	206
Logger	844	18
Hardware	1232	8
Boot/Scheduler	272	14
Total	16044	849

Mate izpildes konteksti



Mate instrukciju kopa

basic	00iiiiiii	i = instruction
s-class	01iiixxx	i = instruction, x = argument
x-class	1ixxxxxx	i = instruction, x = argument

- basic: aritmētika, LEDi, halt
- s-class: atmiņa
- x-class: push const, branch if <=

Mate CounterToLeds aplikācija

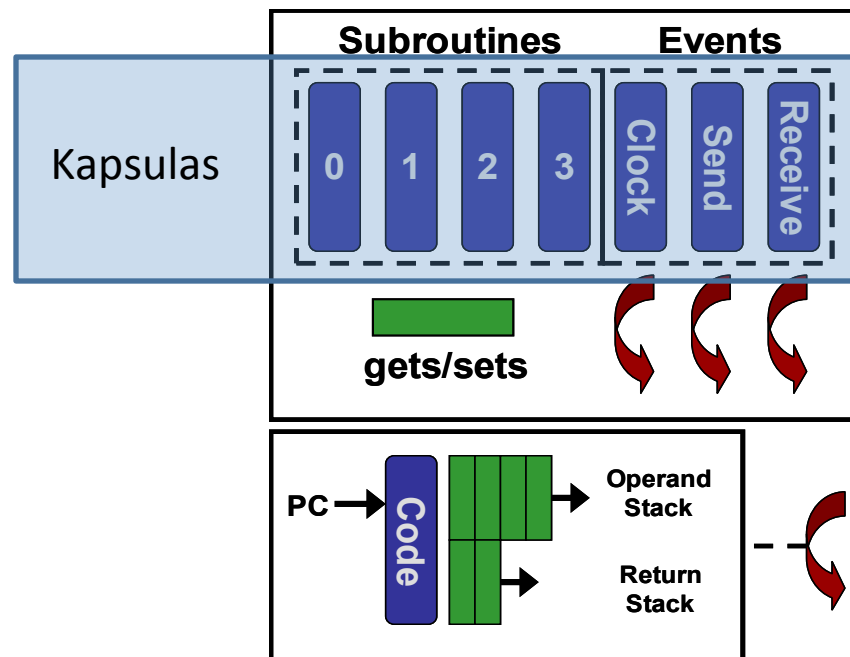
```
pushc 1      # Push one onto operand stack
add          # Add the one to the stored counter
copy        # Copy the new counter value
pushc 7
and          # Take bottom three bits of copy
putled      # Set the LEDs to these three bits
halt
```


Mate SenseAndSend

```
pushc 1    # Light is sensor 1
sense      # Push light reading on stack
pushm      # Push message buffer on stack
clear      # Clear message buffer
add        # Append reading to buffer
send       # Send message using built-in
halt       # ad-hoc routing system
```

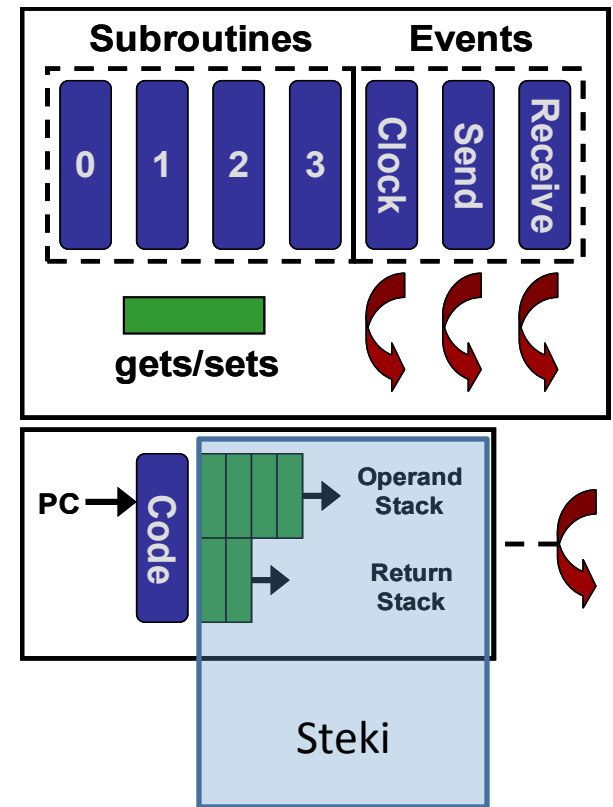
Mate koda kapsulas

- Katra kapsula viena konteksta apstrādei
- Max 24 instrukcijas kapsulā (24B)
- Ja vajag vairāk – apakšprogrammas
- Katrai kapsulai 32bitu versijas numurs



Mate steki

- Katram kontekstam 2 steki:
 - Operandu steks (16B), instrukciju parametri
 - Izsaukumu steks (8B), apakšprogrammām
- Clock konteksts steku saglabā starp izsaukumiem



Mate apakšprogrammas

```
pushc 3    // izsauc apakšprogrammu #3  
call
```

- Atgriešanās adresi ieliek izsaukumu stekā

Mate statistiskais mainīgais

- Tikai viens globālais mainīgais
- Kopīgs visiem kontekstiem
- Instrukcijas:
 - `sets`: vērtību no steka ieraksta mainīgajā
 - `gets`: otrādi

Konteksta kapsulas izpilde

- Iestājas notikums
- Tiek palaista attiecīgā kapsula
- Katra instrukcija kā TinyOS uzdevums
- Konteksti var izpildīties paralēli
- Apstājas pie `halt`

- Kāds notikums izsauc Send kontekstu?

Mate SendOnSenseChange

```
pushc 1      # Push one on the operand stack
sense        # Read sensor 1 (light)
copy         # Copy the sensor reading
gets         # Get previous sent reading

inv          # Invert previous reading
add          # Current - previous sent value
pushc 32
add

blez 17      # If curr < (prev-32) jump to send
copy         # Copy the sensor reading
inv          # Invert the current
gets         # Get the previous reading

add          # Previous - current
pushc 32
add
blez 17      # If (curr+32) > prev jump to send

halt
copy         # PC 17 -- jump-to point from above
sets        # Set shared var to current reading
pushm       # Push a message onto operand stack

clear        # Clear out the message payload
add         # Add the reading to message payload
send        # Send the message
halt
```

Mate koda izmēri

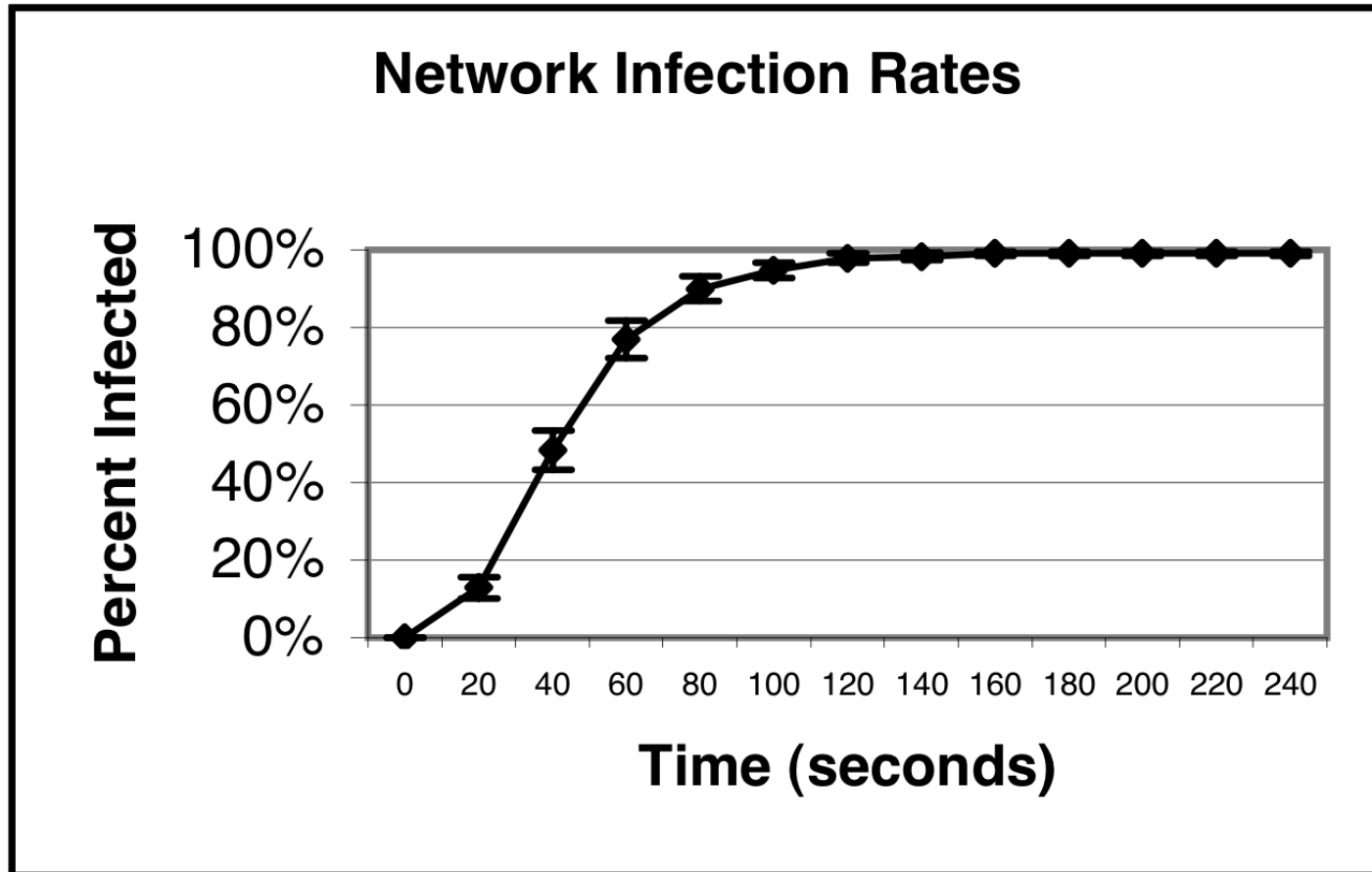
Application	Binary		Maté	
	Size(bytes)	Install Time	Capsules	Instructions
<code>sens_to_rfm</code>	5394	$\approx 79s$	1	6
<code>gdi-comm</code>	7130	$\approx 104s$	1	19
<code>bless-test</code>	7381	$\approx 108s$	7	108

Mate veikspēja

Operation	Maté Clock Cycles	Native Clock Cycles	Cost
Simple: and	469	14	33.5:1
Downcall: rand	435	45	9.5:1
Quick Split: sense	1342	396	3.4:1
Long Split: sendr	685 + $\approx 20,000$	$\approx 20,000$	1.03:1

- Veiktspēja cieš
- Enerģijas patēriņu tas maz ietekmē

Mate infektoloģija



42 motes, 3 x 14 režģis

Mate kopsavilkums

- Viena no pirmajām BST VM
- Maza izmēra aplikācijas kods (<100B)
- Ērta pārprogrammēšana
- Veiktspēja zema, bet energo efektivitāti nebojā

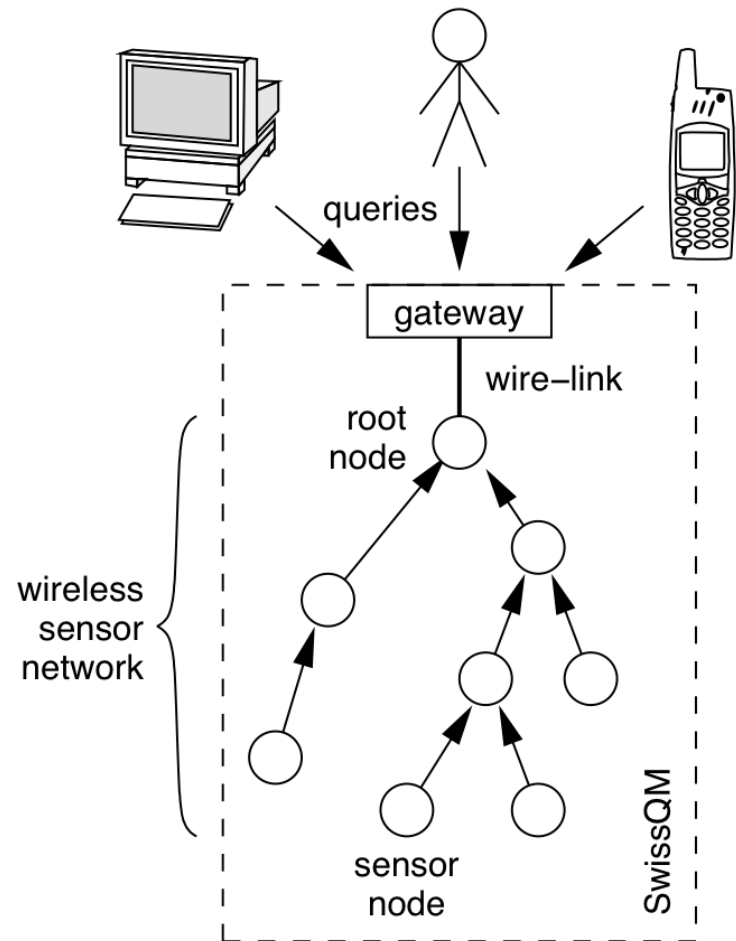
SwissQM virtuālā mašīna

- ETH Zurich, 2007.g
- QM = Query Machine
- Izteiksmīgāka par TinyDB
- Kompaktāka par Mate

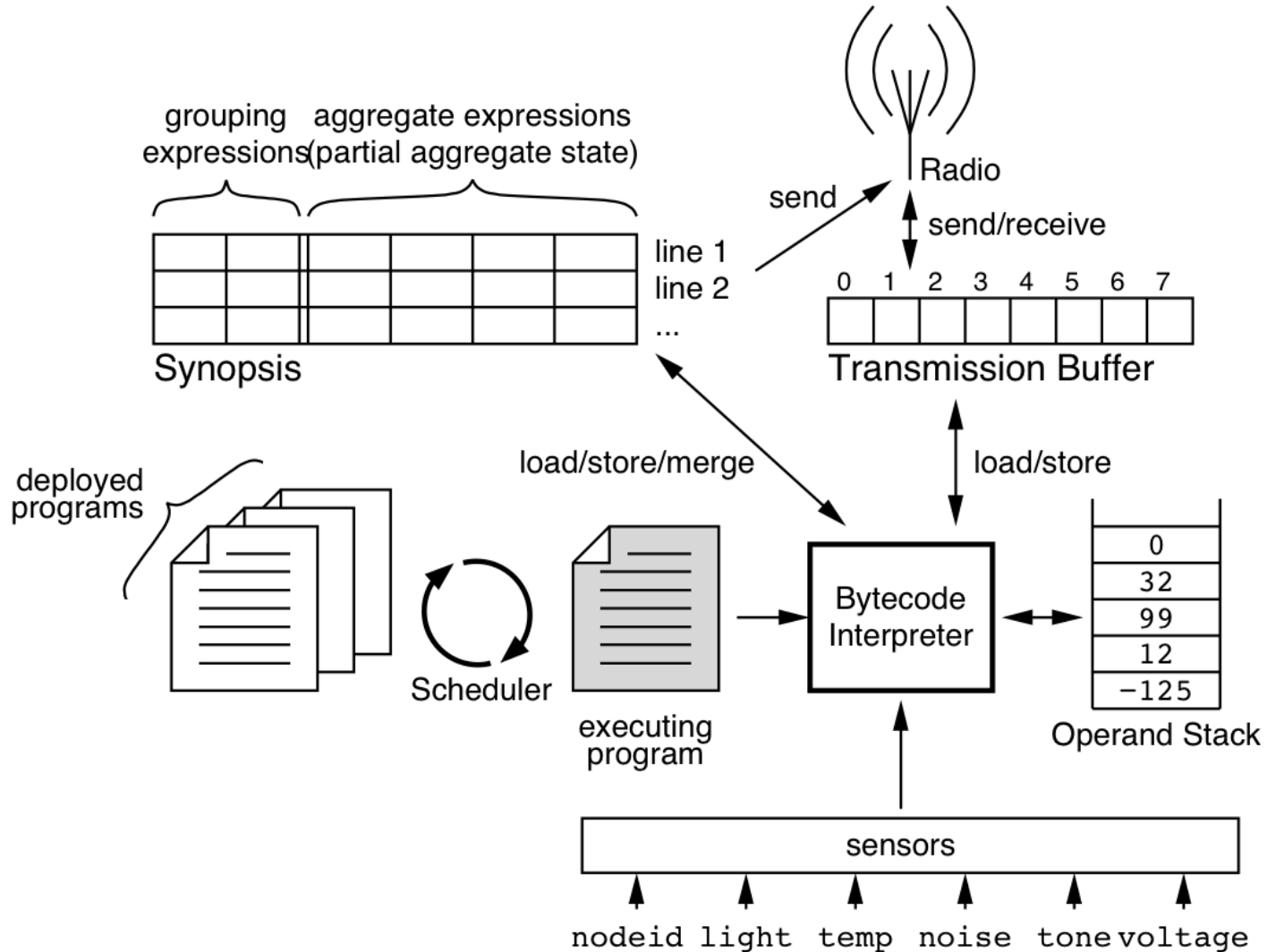


SwissQM pieeja

- Līdz 6 paralēlām aplikācijām
- Samazina abstrakciju
- Paaugstina efektivitāti
- Gateway kā "gudrais mezgls"



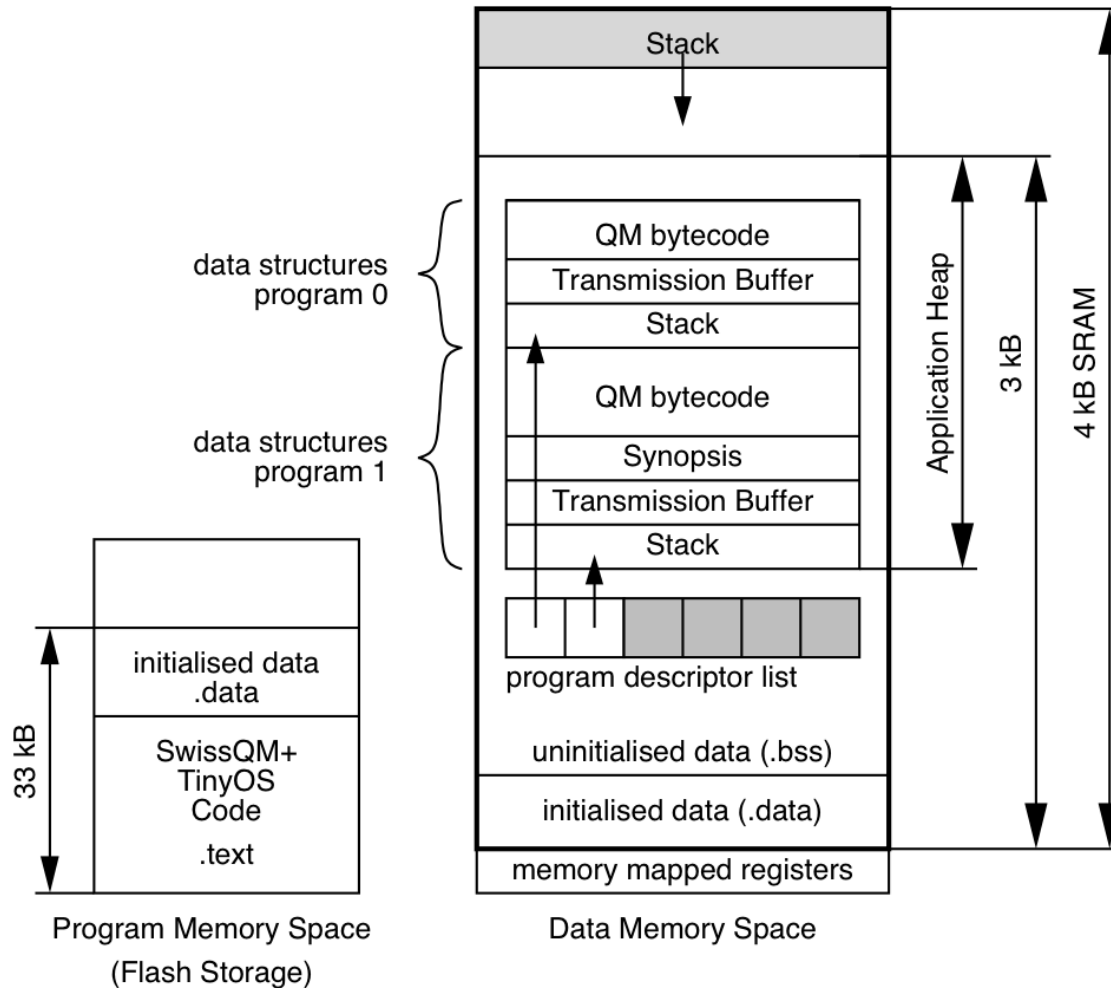
SwissQM arhitektūra



Katrai SwissQM aplikācijai

- Steks: 16B
- Sūtījumu buferis: 16B
- Kopsavilkuma buferis: 16B
- 3 koda sekcijas:
 - Init
 - Delivery
 - Reception

SwissQM atmiņas izvietojums



SwissQM aplikāciju sekcijas

- **Delivery:** Sense & Send cikls, obligāts
- **Init:** sākuma uzstādījumi
- **Reception:** saņemto ziņu apstrāde (noklusētā izturēšanās – pārsūtīt)

SwissQM instrukciju kopa

- 59 instrukcijas, 1-3 baitus garas:
 - 37 Java VM analogi
 - 22 sensori, datu buferu astrāde

Skat. sarakstu!

Sūtījumu buferis

- Ienākošā ziņojuma iegūšanai
- Izejošā ziņojuma būvei
- Pie katras sekcijas izpildes iztīrīts
- Nosūta tikai aizpildīto daļu
- Adresē pēc indeksa
- `istore <x>` un `iload <x>` instrukcijas

SwissQM SenseAndSend

```
1 .section delivery, "@60s"  
2     get_nodeid      # read the node's ID  
3     istore          0 # store it at pos. 0  
4     get_temp        # read temperature sensor  
5     istore          1 # store it at pos. 1  
6     send_tb         # send transmission buffer  
7  
8 .section reception  
9     send_tb         # forward tuple from child
```

- Izmērs: 8B, 1 ziņojums

SwissQM SenseThresAndSend

```
1 .section delivery, "@10s"
2     get_light          # read light sensor
3     ipushw             900 # push light threshold value
4     if_icmpgt         send # jump to send if greater
5     get_temp           # read temp sensor
6     ipushw             500 # push temp threshold value
7     if_icmpgt         send # jump to send if greater
8     goto              skip # skip sending
9 send: get_nodeid      # get the node's ID
10    istore             0 # store it at pos. 0
11    send_tb            # send transmission buffer
12 skip:
```

- Izmērs: 18B, 2 ziņojumi

Kopsavilkuma buferis

- Stāvokļa nodošanai starp izsaukumiem
- Agregācijai ar `merge` instrukciju
- Adresē pēc indeksa
- `istore_sy <x>` un `iload_sy <x>` instrukcijas

Kopsavilkuma izmantošana

- EWMA filtrs: $y_k = 0.8y_{k-1} + 0.2u_k$

```
1 .section init
2   get_nodeid      # get the node's ID
3   istore_sy      0  # store it in synopsis pos. 0
4   get_light      # sample  $u_0$ 
5   istore_sy      1  # store it as  $y_0$  in synopsis
6
7 .section delivery, "@5s","manualclear"
8   get_light      # sample  $u_k$ 
9   iload_sy       1  # read  $y_{k-1}$ 
10  isub           #  $u_k - y_{k-1}$ 
11  ipushb         5  # push 5 on stack
12  idiv           #  $(u_k - y_{k-1})/5$ 
13  iload_sy       1  # read  $y_{k-1}$ 
14  iadd           #  $y_{k-1} + (u_k - y_{k-1})/5$ 
15  istore_sy      1  # update synopsis  $y_{k-1} \leftarrow y_k$ 
16  send_sy        # send synopsis
```

Izmērs: 19B, 2
ziņojumi

SwissQM agregācija

Aggregate	State	Initialiser	Merger	Finaliser
COUNT	c	$c = 1$	$c = c_1 + c_2$	$= c$
MAX	m	$m = expr$	$m = \max(m_1, m_2)$	$= m$
MIN	m	$m = expr$	$m = \min(m_1, m_2)$	$= m$
SUM	s	$s = expr$	$s = s_1 + s_2$	$= s$
AVG	(s, c)	$(expr, 1)$	$(s_1 + s_2, c_1 + c_2)$	$= \frac{s}{c}$
VARIANCE	(s, t, c)	$(expr, expr^2, 1)$	$(s_1 + s_2, t_1 + t_2, c_1 + c_2)$	$= \frac{t}{c} - \frac{s^2}{c^2}$

- Agregācijas funkciju definē stāvoklis un 3 funkcijas (kā TinyDB)
- Finaliser (evaluatoru) rēķina Gateway

SwissQM merge instrukcija

- Operandi stekā norāda grupētāju un vērtību skaitu
- Datus ņem no kopsavilkuma bufera

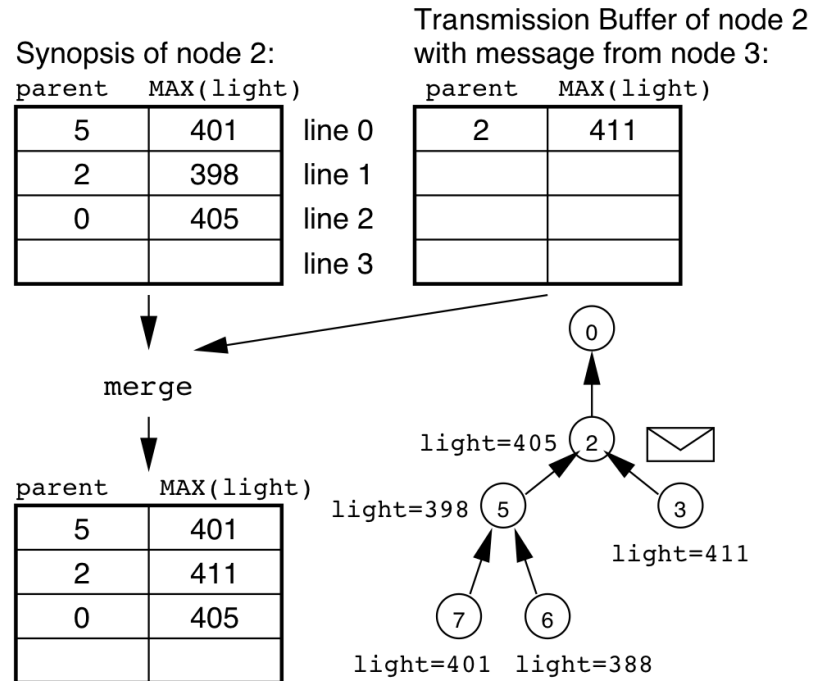
SELECT max(light), avg(temp)
GROUP BY parentId

vērtības

grupētājs

Merge piemērs

SELECT parent, MAX(light)
FROM sensors
GROUP BY parent
SAMPLE PERIOD 10s



Merge piemēra kods

```
1 .section delivery, "@10s", "epochclear"
2   get_parent      # get ID of this node's parent
3   istore          0 # store it as group expression
4   get_light       # read light sensor
5   istore          1 # store it as partial agg. state
6   iconst_2        # aggregate type MAX=2
7   iconst_1        # number of aggregate expressions
8   iconst_1        # number of group expressions
9   merge           # merge transmission buffer
10  send_sy         # send synopsis
11
12 .section reception
13  iconst_2        # aggregate type MAX=2
14  iconst_1        # number of aggregate expressions
15  iconst_1        # number of group expressions
16  merge           # merge transmission buffer
```

- Izmērs: 15B, 1 ziņojums

- TinyDB izmērs: 168B

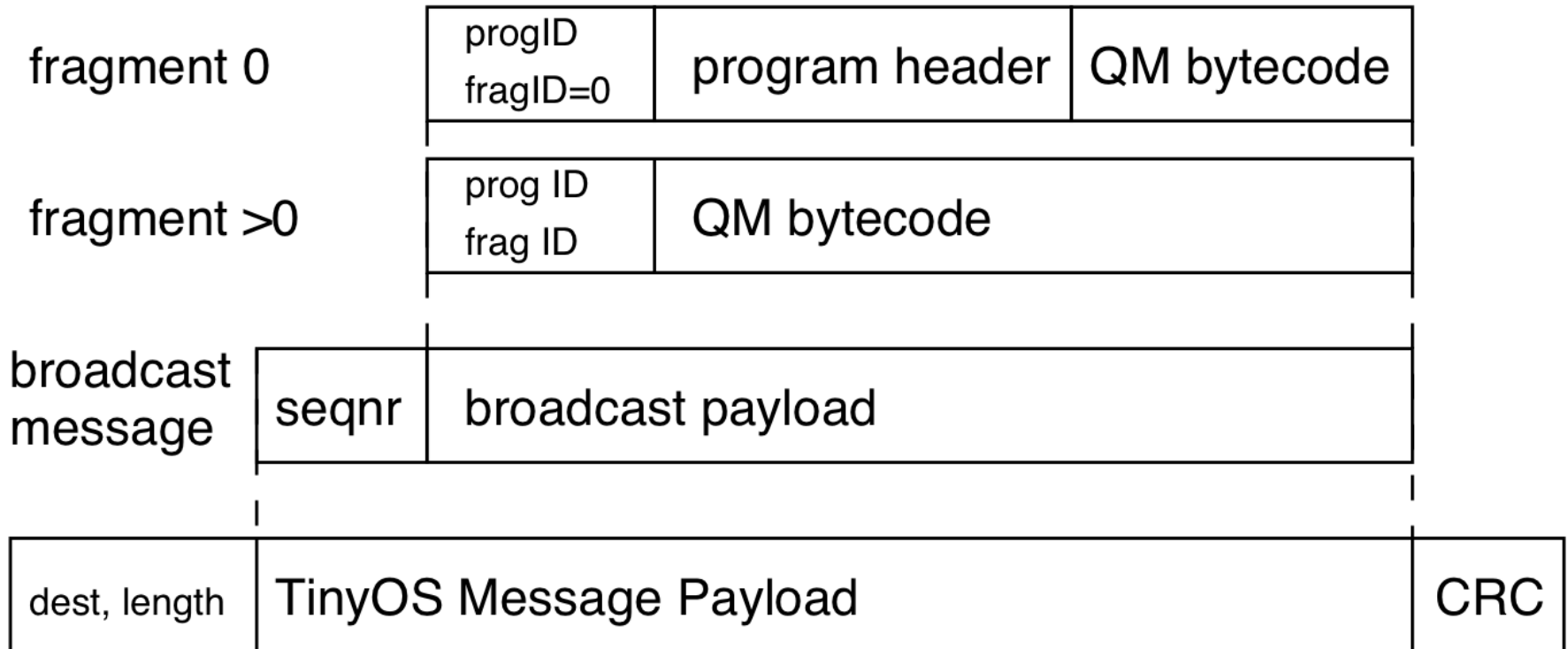
Ziņojumu maršrutizācija

- Tiek uzturēts mezglu koks
- Kopīgs duty cycle – bērni sūta pirms vecāku mezgla

Pārprogrammēšana

- Gateway izsūta ziņas ar aplikācijas galveni un fragmentiem
- Tiklā tās izpludina, pārsūtot
- Ja kāds fragments iztrūkst, pārprasa vecāka mezglam

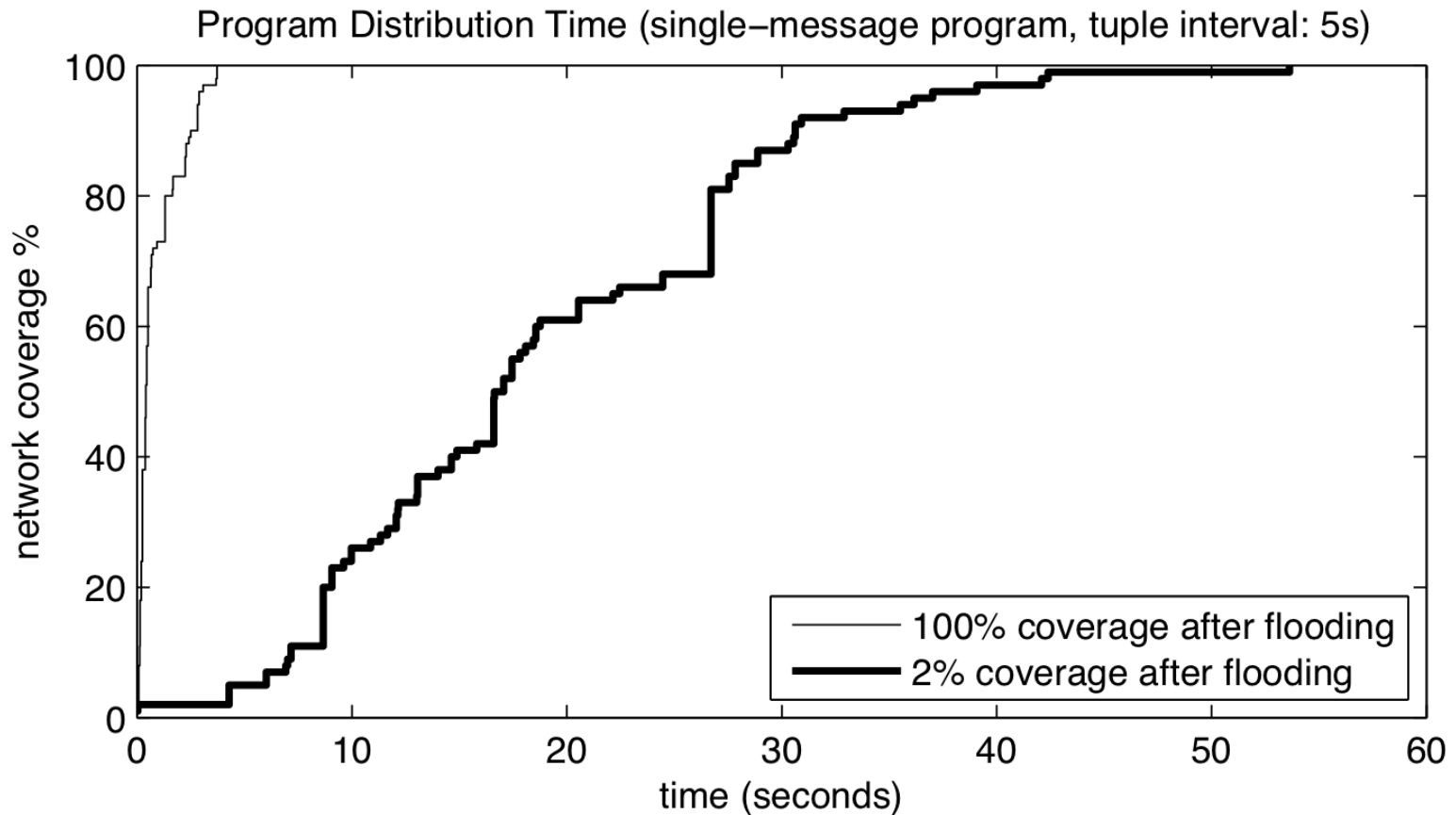
Pārprogrammēšanas ziņojumi



TinyOS Message

Pārprogrammēšanas ātrums

Simulācija: 10x10 mezgli



SwissQM un TinyDB izmēri

Query	TinyDB		SwissQM	
	msgs	bytes	msgs	bytes
SELECT nodeid,light,temp FROM sensors	3	168	1	20
SELECT nodeid,light FROM sensors WHERE temp<512 AND nodeid>50	5	280	2	30
SELECT MAX(light) FROM sensors	2	112	1	22
SELECT parent,MAX(light) FROM sensors GROUP BY parent	3	168	1	25

SwissQM kopsavilkums

- Mazāks kods par Mate
- Plašākas iespējas par TinyDB
- In-Network processing

BST VM turpina dzīvot

A Virtual Machine for Wireless Sensor Networks

Pedro Emanuel Rodrigues Gomes



Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2009

9. Eseja

Kādu labumu Jūsu kursa projektam varētu sniegt virtuālās mašīnas izmantošana?

Termiņš: 07.12.2011. 05:00