Broadcasting in CSP-Style Programming

Brian Vinter
Kenneth Skovhede
Mads Ohm Larsen
The extended channels

One2Any

Any2One

Any2Any

But no “to all”
To all

(a) With the one-to-any model a message ends up at one process in the group

(b) The one-to-all delivers the message to all members of the group.
Open vs closed groups
Challenges in broadcasting
Broadcast types

Simple

Reliable

Atomic

Causal

Synchronous

Asynchronous
Broadcast as a general message

Send

```c
info = pvm_initsend(PvmDataRaw);
info = pvm_pkint(array, 10, 1);
info = pvm_bcast("worker", 42);
```

Receive

```c
buf_id = pvm_recv(&tid, &tag)
info    = pvm_upkint(array, 10, 1)
```
Broadcast as a special message

Send and Receive

result = MPI_Bcast(data, 10, MPI_Int, 0, MPI_COMM_WORLD)
PyCSP Channel recap

All channels are Any2Any

All channels supports both input guards and output guards
Naïve CSP style broadcast
Mailbox approach
Broadcast Channel
Can it be done?

\[ S = m!x \rightarrow S' \]
\[ P_i = m?x \rightarrow P'_i \]
\[ S \parallel \left( \bigoplus_{i=0}^{n} P_i \right) \]
Better approach

\[ S = m!x \rightarrow m_{\text{ACK}} \rightarrow S' \]
\[ B_c = m?x \rightarrow \left( \bigparallel_{i=0}^{n} c_i!x \rightarrow c_{i,\text{ACK}} \rightarrow \checkmark \right) ; m_{\text{ACK}} \rightarrow B'_c \]
\[ P_i = c_i?x \rightarrow c_{i,\text{ACK}} \rightarrow P'_i \]
\[ S \parallel B_c \parallel \left( \bigparallel_{i=0}^{n} P_i \right) \]
The result

![Diagram of interaction between processes](image)
Should it be done???

Broadcasting in CSP has many convenient features

There is no simple way to fold point-to-point messages with broadcasting messages

The motivating example was SME and here the remaining CPS features were not needed