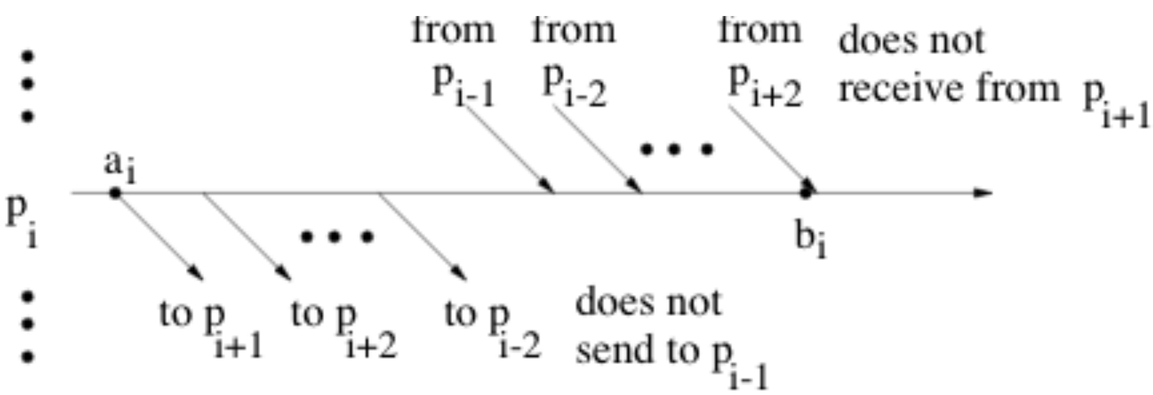
Prof. Jennifer Welch

Vector Clock Size Lower Bound

Theorem: Any implementation of vector clocks using vectors of real numbers requires vectors of length *n* (number of processes).

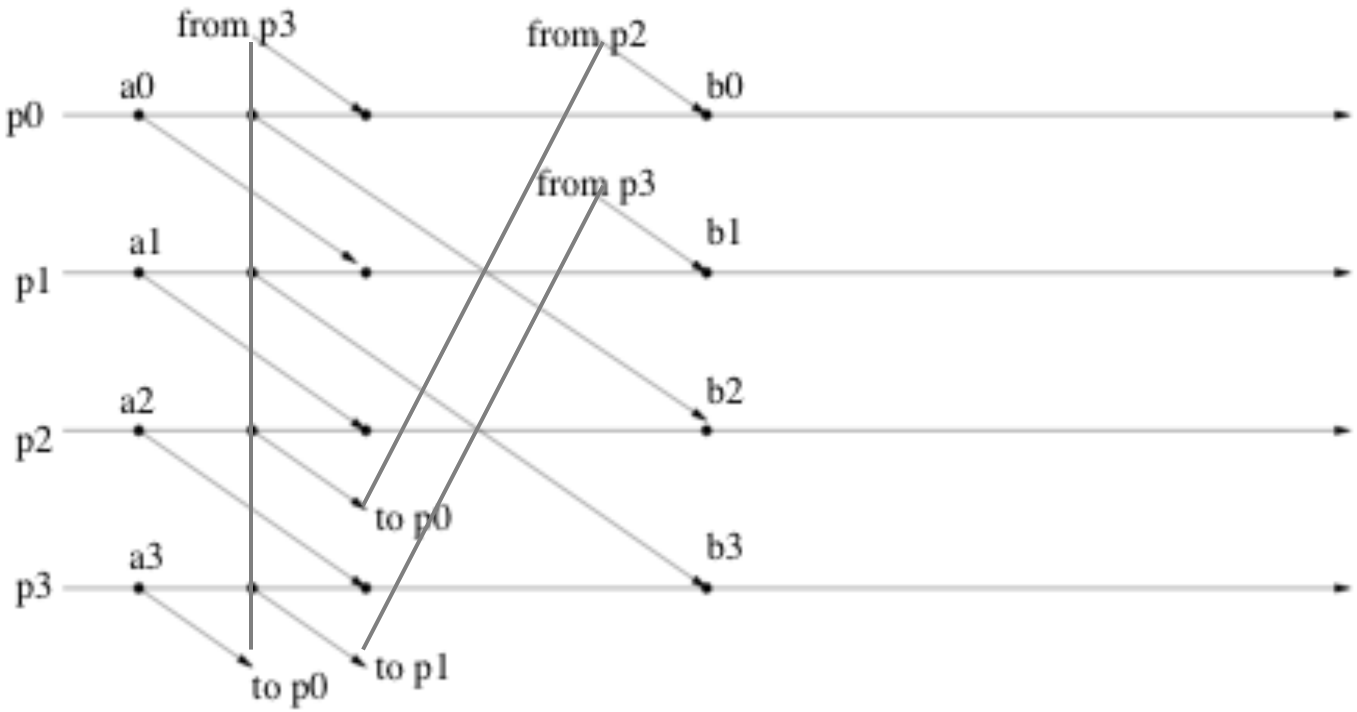
Proof: For any value of *n*, consider this execution:



a_i : first send event at process p_i **b_i : last receive event at p_i**

Example Bad Execution

For $n = 4$:



Vector Clock Size Lower Bound

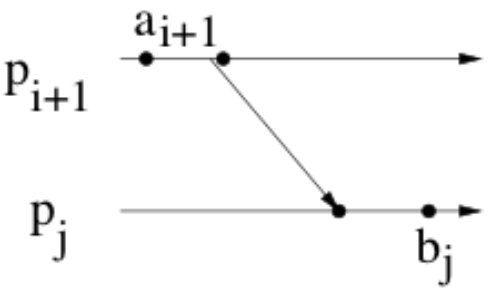
Claim 1: $a_{i+1} \;||\; b_i$ for all i (with wrap-around)

Proof: Since each proc. does all sends before any receives, there is no transitivity. Also p_{i+1} does not send to p_i.

Claim 2: $a_{i+1} \rightarrow b_j$ for all $j \neq i$.

Proof: If $j = i+1$, obvious.

If $j \neq i+1$, then p_{i+1} sends to p_j:



Vector Clock Size Lower Bound

- Suppose in contradiction, there is a way to implement vector clocks with k-vectors of reals, where $k < n$.
- By Claim 1, $a_{i+1} \,||\, b_i$

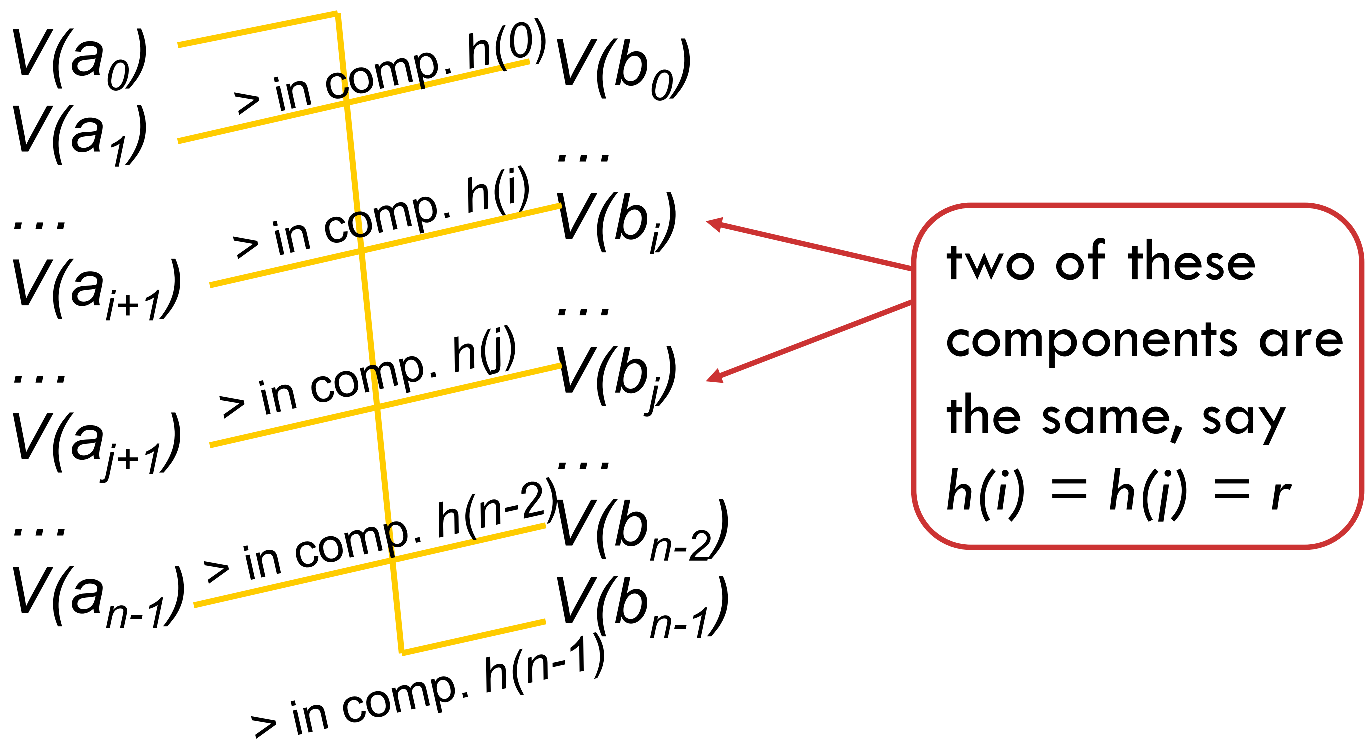
  => $V(a_{i+1})$ and $V(b_i)$ are incomparable

  => $V(a_{i+1})$ is larger than $V(b_i)$ in some coordinate $h(i)$

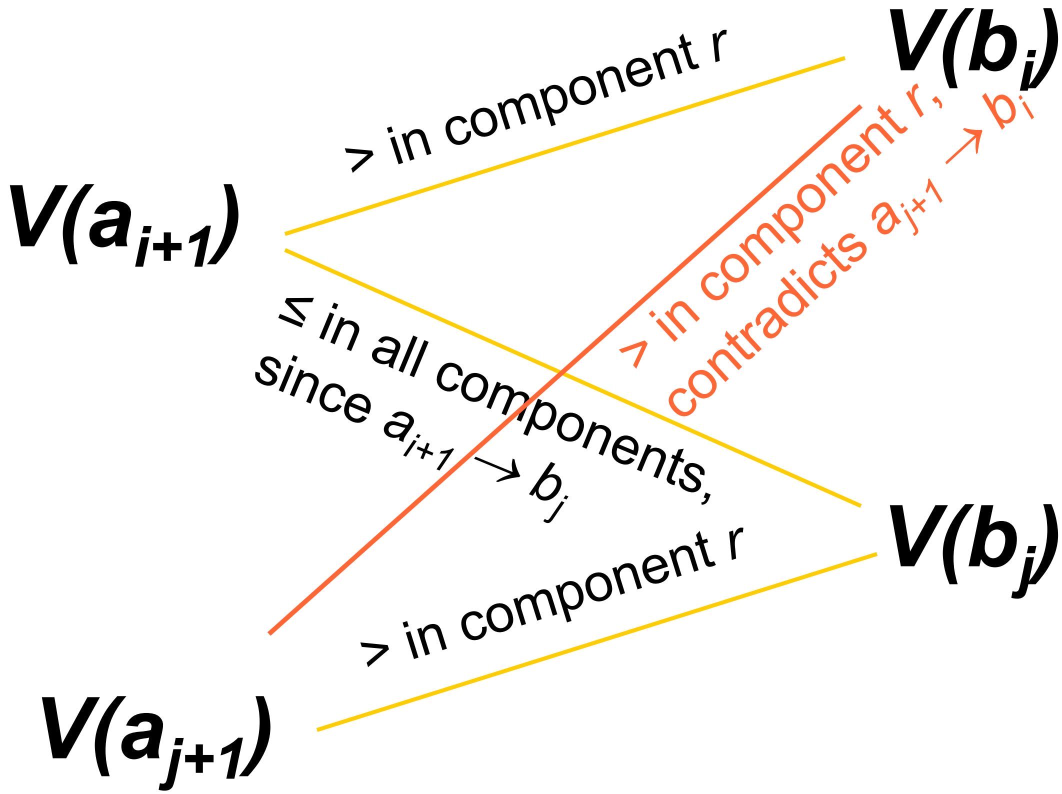
  => $h : \{0,\ldots,n-1\} \rightarrow \{0,\ldots,k-1\}$

Vector Clock Size Lower Bound

- Since $k < n$, the function h is not 1-1. So there exist distinct i and j such that $h(i) = h(j)$. Let r be this common value of h.



Vector Clock Size Lower Bound



Vector Clock Size Lower Bound

- So $V(a_{i+1})$ is larger than $V(b_i)$ in coordinate r and $V(a_{j+1})$ is larger than $V(b_j)$ in coordinate r also.

- $V(a_{j+1})[r] > V(b_j)[r]$ by def. of r
  $\geq V(a_{i+1})[r]$ by Claim 2 ($a_{i+1} \to b_j$) & correct.
  $\geq V(b_i)[r]$ by def. of r

- Thus $V(a_{j+1})$ $!<$ $V(b_i)$, contradicting Claim 2 ($a_{j+1} \to b_i$) and assumed correctness of V.