MISG 2019: ONLINE AUCTION DESIGN

Nouralden — Bill — Lucky — God'sGift
Andrew — Jeandre — Alex
Supervisor: J. W. Sanders

12 January 2019

Contents

Aim

Primary

Can we design an **on-line** auction to incorporate **interesting features** whilst maintaining **required behaviour**?

Aim

Primary

Can we design an **on-line** auction to incorporate **interesting features** whilst maintaining **required behaviour**?

Secondary

What new possibilities does on-line offer?

Recap: Auction Formalisation in Z

- Formal specification of a **A**bstract **D**ata **T**ype (ADT)
- ADT consists of a set and operations on set
- Used to specify state, and how state updates
- Discrete analog to a continuous dynamical system

See appendix for the Z specification of an English auction

Online Auction Characteristics

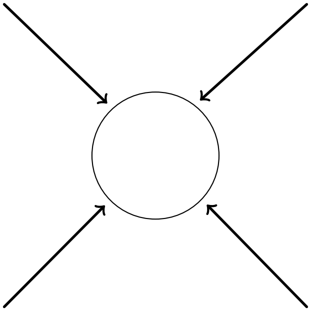
Unique Aspects of an Online Auction:

- increased number of players
- multiple simultaneous auctions (decentralisation)
- reduced costs increase payoffs
- information availability
- security requirements

Centralized System

What is a centralized system?

- A client/server architecture
- All communication goes through a single node
- All computation is done on a single server



Centralized System

Advantages of centralized system:

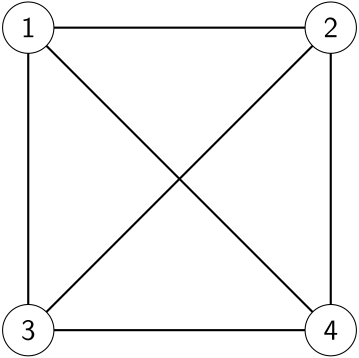
- Simple implementation
- Maximize data integrity and reduce data redundancy
- Subject to location
- Updates to any given set of data are immediately received by every end-user

Centralized System

Disadvantages of centralized system:

- Server bottleneck
- If the server goes down, the platform goes down
- Single source of failure and attack.

Decentralized System



- System whose components are located on different computers
- Communication is done by passing messages

Decentralized System

Advantages of Decentralized system:

- Eliminating single source
- Distribute the load between the nodes
- If a node goes down, the platform still works
- Every node has all the information

Decentralized System

Challenges of Decentralized System:

- Lack of global clock
- Maintaining consistency
- Network communication
- Complicated

How does it work?

Maintaining time:

How does it work?

Maintaining time:

- Universal time, e.g. GMT
- Nodes have regional time zone and system converts it to GMT
- Bidding happens on the same instance universally
- Deadline is then universal

Communication example

How does one guarantee consistency between nodes?

Communication example

How does one guarantee consistency between nodes?

- All the nodes need the same information
- But there is no central node to orchestrate this
- Solution: Communication!

Communication example

Checkpoint 1

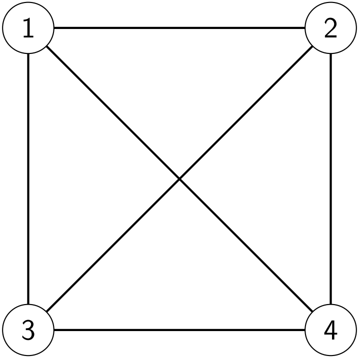


Figure: *Time:* 10 : 00, *Price:* R100

Communication example

Bid at node 1: *Time:* 10 : 10, *Price:* R150

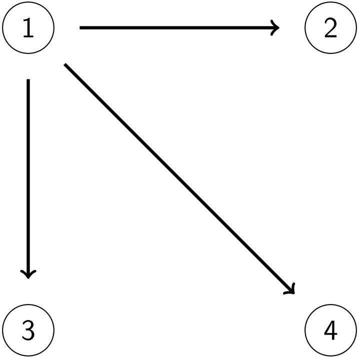


Figure: Broadcasting information from node 1

Communication example

Bid at node 1: *Time:* 10 : 10, *Price:* R150

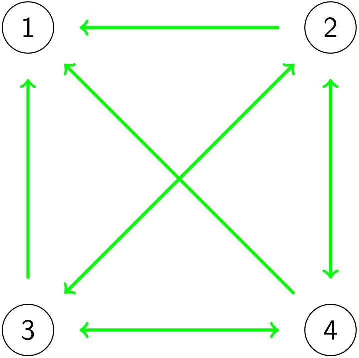


Figure: Broadcasting replies from other nodes

Communication example

As soon as a node receives 3 acceptance broadcasts (including its own), the node updates the new checkpoint:

Communication example

As soon as a node receives 3 acceptance broadcasts (including its own), the node updates the new checkpoint:

Checkpoint 2

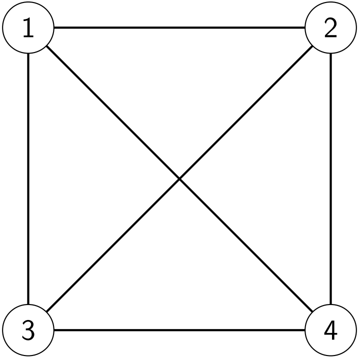


Figure: *Time:* 10 : 10, *Price:* R150

Communication example

What can go wrong?

Communication example

What can go wrong?

Bid at node 2: *Time:* 10 : 20, *Price:* R200

Bid at node 3: *Time:* 10 : 21, *Price:* R220

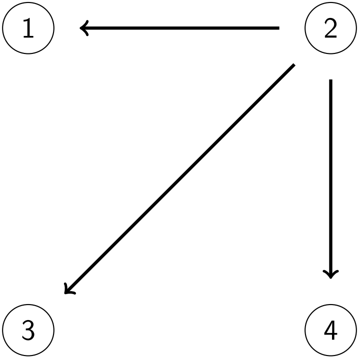


Figure: Broadcasting information from node 2

Communication example

Bid at node 2: *Time:* 10 : 20, *Price:* R200 (Rejected)
Bid at node 3: *Time:* 10 : 21, *Price:* R220

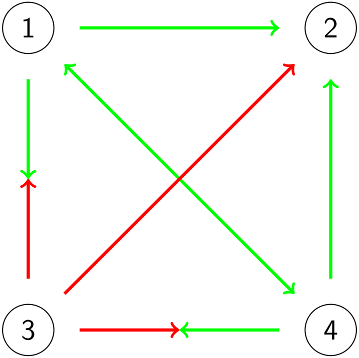


Figure: Broadcasting replies from other nodes

Communication example

Bid at node 2: *Time:* 10 : 20, *Price:* R200
Bid at node 3: *Time:* 10 : 21, *Price:* R220 (Accepted)

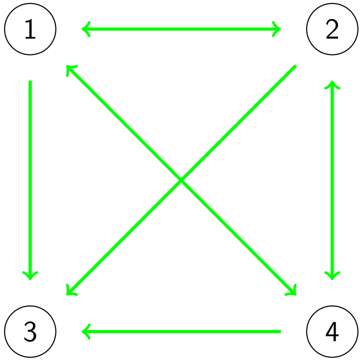


Figure: Broadcasting replies from other nodes

Communication example

Checkpoint 3

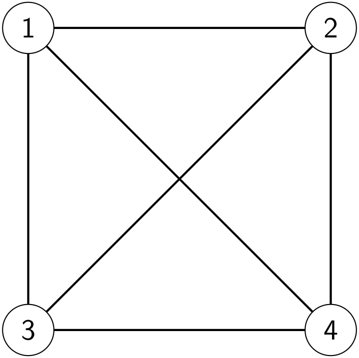


Figure: *Time:* 10 : 20, *Price:* R220

Node buffering

What if a new bid comes in while the previous bid is still being processed?

Node buffering

What if a new bid comes in while the previous bid is still being processed?

- **Processing bid at node 3:** *Time:* 10 : 21, *Price:* R220
- **New bid at node 3:** *Time:* 10 : 22, *Price:* R250

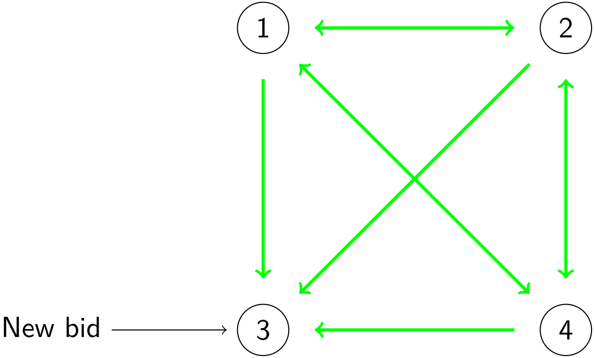


Figure: Broadcasting replies from other nodes

Node buffering

What if a new bid comes in while the previous bid is still being processed?

- **Processing bid at node 3:** *Time:* 10 : 21, *Price:* R220
- **New bid at node 3:** *Time:* 10 : 22, *Price:* R250

Node buffering

What if a new bid comes in while the previous bid is still being processed?

- **Processing bid at node 3:** *Time:* 10 : 21, *Price:* R220
- **New bid at node 3:** *Time:* 10 : 22, *Price:* R250

The new bid is buffered. This means the system first finishes its current process, before node 3 again sends the information on the new bid to the other nodes

Node delay

- What if a node goes down and does not reply?

Node delay

- What if a node goes down and does not reply?
- The system will have a preset time in which each node has to reply
- If the node does not reply in time, the system will continue without it

Node delay

- What if a node goes down and does not reply?
- The system will have a preset time in which each node has to reply
- If the node does not reply in time, the system will continue without it
- If the node comes back online, then it simply gets updated with the current checkpoint

The deadline

How does the system handle the deadline with communication delays?

The deadline

How does the system handle the deadline with communication delays?

- There is a soft deadline and a hard deadline
- The soft deadline occurs when the GMT deadline is reached at each node
- No bids can be played after this at any node

The deadline

- There is also a hard deadline

The deadline

- There is also a hard deadline
- After the hard deadline, a final checkpoint is reached
- This deadline is used by the system to sort out the last unsaved bids
- It gives time for delayed last bids to come through
- After the hard deadline is reached, a winner is announced

Exploiting Information

The **online** aspect allows for data to be captured. Data includes historical auction data and meta-data.

Information is communal in a decentralised system and can be used to improve functioning of the network.

Information can be used to:

- inform and improve additional features: security, system integrity
- (network-level) Improve matching of buyers and sellers
- (auction-level) Inform decisions for buyers and sellers

Network-level Information

- “Success” of the network system can be measured by the number of auctions
- Each auction implies a positive (possibly asymmetric) payoff for both the winning bidder and seller
- Auction parameters and controls can be learned to maximize the number of auctions.
- The result is improved (decentralized) matching of buyers and sellers

Auction-level Information

Suggestions to 3 stakeholder types - Buyers , Sellers, Manufacturers Retailers

- Information for prospective retailers: show items that are selling
- Assumption of an auction: sellers sell high and buyers buy low. Information for seller: probability of making a sale based on empirical evidence (how well product is doing)
- Can use regression analysis to plot price of goods vs probability of it being sold and provide that information to prospective sellers
- Informs a plot: `rest_series.html`

Conclusion

- Auction fundamentals and formalisation of the information system - Z Notation
- Online allows for Decentralisation
- Data can be used to improve network functioning and bidder strategies
- Prove the chosen auction design behaves as desired

References

- Chapter 9, Auctions, in *Networks, Crowds and Markets: Reasoning about a Highly Connected World.* D. Easley and J. Kleinberg. CUP, 2010. www.cs.cornell.edu/home/kleinber/networks-book/
- *Mathematical Underpinning of Analytics: Theory and Applications.* P. Grindrod, OUP, 2015. resituatq.firebaseapp.co/aa588/.../0198725094.pdf
- *The Z Notation: A Reference Manual.* J. M. Spivey, Prentice-Hall, 2001. onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp69629

Z Notation

- Formal Specification
- Reference of point for: Analyzer, Programmer, Tester and Documentary
- Decomposing the specification into schemes
- Describing the static and dynamic aspect of the system.

Auctions Framework By Example

Auction

Condition regarding stored inputs

$seller, cbuyer : \mathbb{U}$

Init

State

Interaction between the variables

Feature Example: State

State

$\Delta seller, cbuyer : \mathbb{U}$
$item : \mathbb{I}$
$history : \mathrm{seq}\ BIDS$
$cbid : \mathbb{R}^+$
$scost : \mathbb{B}$
$dline : \mathbb{T}$
$sp : \mathbb{R}^+$
$inc : \mathbb{R}^+$
$systime : \mathbb{T}$

$cbuyer = history[-1] \cdot u?$
$cbid = history[-1] \cdot price?$

Feature Example: Bids

$$
\begin{array}{l}
\textit{Bids} \\
\hline
\Delta State \\
cbuyer? : \mathbb{U} \\
cprice? : \mathbb{R}^{+} \\
btime? : \mathbb{T} \\
\hline
price? \geq cbid + inc \\
systime \leq btime < dline \\
history' = history + [cbuyer?, price?, btime?] \\
\hline
\end{array}
$$

Feature Example: Init

$$\begin{array}{l}
\textit{INIT} \\
\hline
\textit{State}' \\
\textit{seller}? : \mathbb{U} \\
\textit{sp}? : \mathbb{R}^+ \\
\textit{inc}? : \mathbb{R}^+ \\
\textit{item}? : \mathbb{I} \\
\textit{dline}? : \mathbb{T} \\
\textit{cbid}? : \mathbb{R}^+ \\
\textit{ctime}! : \mathbb{T} \\
\textit{scost}? : \mathbb{B} \\
\hline
\textit{cbuyer}' = [\,] \\
\textit{history}' = [\,] \\
\textit{cbid}! = \textit{cbid}' = \textit{sp}? \\
\textit{ctime}! = \textit{systime} \\
\hline
\end{array}$$

Feature Example: Init (Continued)

$inc > 0$

$scost' = scost?$

Feature Example: USERINFO

$\mathbb{U}$

username
uid
bankdetails
contactdetails

$* * *$

Feature Example: ITEM

I

picture
description
location

* * *

Feature Example: END

$$
\begin{array}{l}
\textit{END} \\
\hline
\Xi State \\
cbuyer! \\
price! \\
btime! \\
\hline
systime = dline \Rightarrow \\
\quad cbuyer! = history[-1] \cdot cbuyer \\
\quad price! = history[-1] \cdot price \\
\quad btime! = history[-1] \cdot btime
\end{array}
$$