Knowledge and Gossip

Hans van Ditmarsch  recent work by and with many others
LORIA / CNRS — University of Lorraine & IMSc, Chennai
Six friends each know a secret. They can call each other. In each call they exchange all the secrets they know. How many calls are needed for everyone to know all secrets?
Friends exchanging secrets

Six friends each know a secret. They can call each other. In each call they exchange all the secrets they know. How many calls are needed for everyone to know all secrets?

First consider four friends $a, b, c, d$ who hold secrets $A, B, C, D$. Four calls $ab; cd; ac; bd$ distribute all secrets.

\[
\begin{align*}
A.B.C.D & \xrightarrow{ab} AB.AB.C.D \\
& \xrightarrow{cd} AB.AB.CD.CD \\
& \xrightarrow{ac} ABCD.AB.ABCD.CD \\
& \xrightarrow{bd} ABCD.ABCD.ABCD.ABCD
\end{align*}
\]

Now consider friends $a, b, c, d, e, f$ with secrets $A, B, C, D, E, F$. Eight calls $ae; af; ab; cd; ac; bd; ae; af$ distribute all secrets.

For $n$ agents: minimum $2n - 4$. 
Six friends each know a secret. They can call each other. In each call they exchange all the secrets they know. How many calls are needed for everyone to know all secrets?

First consider four friends $a, b, c, d$ who hold secrets $A, B, C, D$. Four calls $ab; cd; ac; bd$ distribute all secrets.

\[
\begin{align*}
A.B.C.D & \xrightarrow{ab} AB.AB.C.D & \xrightarrow{cd} AB.AB.CD.CD & \xrightarrow{ac} A.BCD.AB.BCD.ABCD.
\end{align*}
\]

Now consider friends $a, b, c, d, e, f$ with secrets $A, B, C, D, E, F$. Eight calls $ae; af; ab; cd; ac; bd; ae; af$ distribute all secrets. For $n$ agents: minimum $2n – 4$.

But how does $c$ know that she should call $d$?
Privacy considerations — callers are observed

Four calls $ab; cd; ac; bd$ distribute all secrets. How does $c$ know that she should call $d$?

Because $c$ observed $a$ and $b$ making a call. After call $ab$, $c$ knows that $a$ and $b$ both know the secrets $A$ and $B$. But $c$ does not know the secrets $A$ and $B$. 
Privacy considerations — calls are observed

(After Agatha Christie) Consider a murder house with four friends in all different rooms containing a telephone. Telephones make a clicking sound if two people in the house make a call.

What does $c$ know after call $ab$?
Privacy considerations — calls are observed

(After Agatha Christie) Consider a murder house with four friends in all different rooms containing a telephone. Telephones make a clicking sound if two people in the house make a call.

What does \( c \) know after call \( ab \)?
Privacy considerations — time of calls is observed

(After Agatha Christie) Consider a murder house with four friends in all different rooms containing a telephone. Telephones make a clicking sound if two people in the house make a call if someone picks up a phone to make a call (but may not make a connection).

What does $c$ know after call $ab$?
Privacy considerations — time of calls is observed

(After Agatha Christie) Consider a murder house with four friends in all different rooms containing a telephone. Telephones make a clicking sound if two people in the house make a call if someone picks up a phone to make a call (but may not make a connection).

What does $c$ know after call $ab$?

Before we had that $c$ considers any call possible that does not involve her. She cannot distinguish:

- $ab$ resulting in $AB.AB.C.D$;
- $ad$ resulting in $AD.B.C.AD$;
- $bd$ resulting in $A.BD.C.BD$.

Now $c$ also considers possible:

- no call, resulting in $A.B.C.D$. 
Privacy considerations — only own calls are observed

What does \( c \) know after call \( ab \)?
Friend \( c \) considers possible:

\[\begin{align*}
\text{ab} & \quad \text{resulting in } AB & & \text{all the previous:} \\
\text{ad} & \quad \text{resulting in } AD & & \text{ab resulting in } AB, \text{AD}; \\
\text{bd} & \quad \text{resulting in } ABD & & \text{ab resulting in } AB, \text{AD}, \text{BD}; \\
\text{no call} & \quad \text{resulting in } ABD & & \text{but now as well any sequence of calls not involving here:} \\
\text{ad} & \quad \text{Resulting in } AD & & \text{bd}; \\
\text{bd} & \quad \text{Resulting in } ABD & & \text{ad}; \\
\text{ad} & \quad \text{Resulting in } ABD & & \text{bd}; \\
\text{ab} & \quad \text{Resulting in } ABCD & & \text{also resulting in } AB, \text{AD}; \\
\end{align*}\]

We can call this the asynchronous situation (no global clock) [see Krzysztof Apt et al., TARK 2015]
Privacy considerations — only own calls are observed

What does \( c \) know after call \( ab \)?
Friend \( c \) considers possible:
all the previous:

- \( ab \) resulting in \( AB.AB.C.D \);
- \( ad \) resulting in \( AD.B.C.AD \);
- \( bd \) resulting in \( A.BD.C.BD \);
- no call, resulting in \( A.B.C.D \);

but now as well any sequence of calls not involving here:

- \( ad; bd \) resulting in \( AD.ABD.C.ABD \);
- \( bd; ad \) resulting in \( ABD.BD.C.ABD \);
- \( ad; bd; ad \) resulting in \( ABD.ABD.C.ABD \);
- \( ab; ab \) (also) resulting in \( AB.AB.C.D \);

\( ... \)

We can call this the asynchronous situation (no global clock)
[see Krzysztof Apt et al., TARK 2015]
Call history

*Different knowledge but same distribution of secrets:*

- Call sequences $ab; cd; ac; bd$ and $ac; bd; ab; cd$ both result in all four agents knowing all four secrets.
- After $ab; cd; ac; bd$ agent $d$ knows that $b$ knows all secrets.
- After $ac; bd; ab; cd$ agent $d$ does not know that.

[Rahim Ramezanian]

*Different knowledge but same local call history:*

1. Either they first inspect what secrets they receive from the callee and then merge them with their own secrets.
2. Or they first merge the secrets they receive with their own secrets and then inspect the result.
3. Call sequences (i) $bc; ab; bd; ad$ and (ii) $bc; ab; ad$ are different for agent $a$ under (1.), but the same under (2.).

[François Schwarzentruber] (1.)(i): $a$ learns that $b$ or $c$ called $d$
A gossip protocol is a program of shape “until all agents know all secrets, choose agents $x, y$ such that $x$ knows that $\varphi(x, y)$, and let $x$ call $y.$” More distributed descriptions are possible.

An execution sequence of a gossip protocol is successful if it terminates with all agents knowing all secrets.

Strongly successful protocol: all executions are successful.

Fairly successful protocol: all fair executions are successful.

Weakly successful protocol: some executions are successful.

ANY

Until all agents know all secrets, any agent $x$ calls any agent $y$.

LNS — Learn New Secrets

Until all agents know all secrets, an agent $x$ who does not know all secrets calls an agent $y$ whose secret it does not know.

LNS: [Maduka Attamah et al., Knowledge and Gossip, ECAI 2014]
Learn New Secrets protocol — example for four agents

Until all agents know all secrets, an agent $x$ who does not know all secrets calls an agent $y$ whose secret it does not know.

Minimum execution length is $2n - 4$, maximum is $n(n - 1)/2$. A minimal call sequence (2 · 4 − 4 = 4) is $ab; cd; ac; bd$. A maximal call sequence (4(4 − 1)/2 = 6) is $ab; ac; ad; bc; bd; cd$. There are also executions of length 5, e.g. $ab; ac; ad; bd; cd$. It can be shown that the average execution length is larger than 5.
Learn New Secrets protocol — example for four agents

Until all agents know all secrets, an agent $x$ who does not know all secrets calls an agent $y$ whose secret it does not know.

Minimum execution length is $2n - 4$, maximum is $n(n - 1)/2$.
A minimal call sequence ($2 \cdot 4 - 4 = 4$) is $ab; cd; ac; bd$.
A maximal call sequence ($4(4 - 1)/2 = 6$) is $ab; ac; ad; bc; bd; cd$.
There are also executions of length 5, e.g. $ab; ac; ad; bd; cd$.
It can be shown that the average execution length is larger than 5.

Consider again the 6 agent execution $ae; af; ab; cd; ac; bd; ae; af$. It is not an execution of Learn New Secrets. ($a$ already called $e, f$)
LNS execution with same secrets distr.: $ae; af; ab; cd; ac; bd; eb; fb$.

Another protocol is:

Known Information Growth
Until all agents know all secrets, an agent $x$ calls an agent $y$ if $x$ knows that $x$ or $y$ will learn a new secret in call $xy$. 
Gossip protocols

ANY
Until all agents know all secrets, any agent \( x \) calls any agent \( y \).

LNS — Learn New Secrets
Until all agents know all secrets, an agent \( x \) who does not know all secrets calls an agent \( y \) whose secret it does not know.

KIG — Known Information Growth
Until all agents know all secrets, an agent \( x \) calls an agent \( y \) if \( x \) knows that \( x \) or \( y \) will learn a new secret in call \( xy \).

PIG — Possible Information Growth
Until all agents know all secrets, an agent \( x \) calls an agent \( y \) if \( x \) considers possible that \( x \) or \( y \) will learn a new secret in call \( xy \).

If only own calls are observed, LNS and KIG are identical.

[Pere Pardo and Rahim Ramezanian]
If only own calls are observed, ANY and PIG are (nearly) identical.
Gossip Graphs

We assumed that all agents can call all other agents (universal relation). Different protocols apply when you can only call your neighbours (partial, often symmetric, relation — a *gossip graph*). This affects the optimal call sequence. It may not be $2n - 4$. If the graph is connected, it is between $2n - 3$ and $2n - 4$.

For example, 5 nodes: 6 calls is not possible but 7 calls is possible.
This is not logic
This is not logic

This is epidemiology / network theory (This is wishful thinking)
This is not logic

This is epidemiology / network theory  (This is wishful thinking)

Some words on l.

- call sequences induce indist. relations to interpret knowledge
- calls are non-public events that correspond to action models
- (gossip) protocol generated forest if all only know own secret
- the asynchronous situation is a challenge 😊
This is not logic

This is epidemiology / network theory  (This is wishful thinking)

Some words on l... 

▶ call sequences induce indist. relations to interpret knowledge
▶ calls are non-public events that correspond to action models
▶ (gossip) protocol generated forest if all only know own secret
▶ the asynchronous situation is a challenge ☺

This is logic

▶ Jeremy Seligman, Fenrong Liu, Patrick Girard (2013). Facebook and the epistemic logic of friendship. (TARK)
▶ ... ,  ... ,  ...
Dynamic Gossip

We assumed that all agents can call all other agents. Different protocols apply when you can only call your neighbours. Let us further assume that callers exchange all secrets and all numbers (neighbours) they know. And we no longer assume symmetry.

Learn New Secrets protocol (with numbers)  Until all agents know all secrets, an agent $x$ who does not know all secrets calls an agent $y$ whose number it has and whose secret it does not know.

On fully connected graphs there is no difference. Before, we displayed this as:

$$\begin{align*}
A.B.C & \rightarrow^{ab} AB.AB.C & \rightarrow^{bc} \ldots \\
A & \leftrightarrow^{ab} b \leftrightarrow^{ab} c \\
A & \leftrightarrow^{bc} B & \leftrightarrow^{bc} C
\end{align*}$$

Now, we display this as:

$$\begin{align*}
A & \leftrightarrow^{ab} b \leftrightarrow^{ab} c \\
A & \leftrightarrow^{bc} B & \leftrightarrow^{bc} C
\end{align*}$$
Dynamic Gossip — Learn New Secrets (with numbers)

Agents exchange all secrets and all numbers they know.

On fully connected graphs there is no difference.

\[ \begin{align*}
A & \leftrightarrow b \leftrightarrow c \\
B & \leftrightarrow c \leftrightarrow A \\
C & \leftrightarrow a \leftrightarrow b \\
\end{align*} \]

On partially connected graphs deadlock is possible. (After \(bc; ab\) agent \(c\) cannot call agent \(a\), because \(c\) does not have \(a\)'s number.)

\[ \begin{align*}
A & \rightarrow b \rightarrow c \\
B & \rightarrow c \rightarrow A \\
C & \rightarrow b \rightarrow c \\
\end{align*} \]

But sometimes deadlock can be avoided. (After \(ab; bc\) agent \(a\) calls agent \(c\).)

\[ \begin{align*}
A & \rightarrow b \rightarrow c \\
B & \rightarrow c \rightarrow A \\
C & \rightarrow a \leftrightarrow b \leftrightarrow c \\
\end{align*} \]

When exactly?
Dynamic Gossip — Characterization for LNS

Termination properties:

- An execution sequence of a gossip protocol is **successful** if it terminates with all agents knowing all secrets.
- **Strongly successful** protocol: all executions are successful.
- **Weakly successful** protocol: some executions are successful.


- If the graph is *not weakly connected*, no execution of any (dynamic) gossip protocol is successful.
  Worse: bush or double bush, not weakly successful.
- If the graph is *strongly connected*, all executions of LNS are successful.
  Better: sun, strongly successful.

What on earth is a sun? Or a bush or a double bush?
Dynamic Gossip — Characterization for LNS

- A **sun** is a strongly connected graph to which may be connected maximal nodes.
- A **bush** is a tree with branching factor in the root at least 2.
- A **double bush** consists of two bushes joined in a leaf connected to their roots.

sun | bush | double bush
Dynamic Gossip — Characterization for LNS

- LNS is not weakly successful on a gossip graph iff
  the gossip graph is a bush or a double bush

Example of how to get lost in the bush.

```
A → B  C
  \  /  \\
   \ /   \\
    \   /
      \ /  \\
       \ /   \\
        \   /
         \ /
          A

AB → B  AB
  \  /  \\
   \ /   \\
    \   /
      \ /   \\
       \ /   \\
        \   /
         \ /
          AB

AB → AB  ABC
  \  /  \\
   \ /   \\
    \   /
      \ /   \\
       \ /   \\
        \   /
         \ /
          ABC
```
Iterated Shared Knowledge — How to get $E^2All$

$EAll$: everybody knows all secrets. $E^2All$: everybody knows that everybody knows all secrets. To obtain $E^2All$ (at termination), every agent executes the following protocol:

▶ Until you know all secrets, call an agent whose secret you do not know.

▶ When you know all secrets, then until you know that all agents know all secrets, call an agent who you do not know to know all secrets.

[Assumes that agents know how many other agents there are!]

Example for four agents:

- $ab; cd; ac; bd$: all agents know all secrets
- $ab; ad$: agent $a$ knows that all agents know all secrets
- $bc$: agent $b$ knows that all agents know all secrets
- $cd$: agents $c, d$ know that all agents know all secrets
Communicating Iterated Shared Knowledge

To obtain $E^2\text{All}$ we can execute an algorithm of $O(n^2)$. For $n = 4$:

- $ab; cd; ac; bd;$ all agents know all secrets
- $ab; ad;$ agent $a$ knows that all agents know all secrets
- $bc;$ agent $b$ knows that all agents know all secrets
- $cd;$ agents $c, d$ know that all agents know all secrets

If the agents can also communicate knowledge about secrets, to obtain $E^k\text{All}$ we need at least $(k + 1)(n - 2)$ calls [Herzig & Maffre, IAT]. This is $O(n)$. For $n = 4$, again $E^2\text{All}$:

- $ab; cd; ac; bd;$ all agents know all secrets
- $ab;$ agent $a$ informs $b$ that $a, c$ know all secrets
  agent $b$ informs $a$ that $b, d$ know all secrets
  agents $a, b$ know that all agents know all secrets
- $cd$ agent $c$ informs $d$ that $a, c$ know all secrets
  agent $d$ informs $c$ that $b, d$ know all secrets
  agents $c, d$ know that all agents know all secrets
Common Knowledge and Gossip

What may or may not be common knowledge:
- The protocol
- The gossip graph (i.e., the network topology)
- The number of agents (i.e., nodes)
- That agents initially hold a single secret
- The time

Some results (ongoing research):
- Consider $a \rightarrow b \rightarrow c$. If LNS, the gossip graph, and time are common knowledge, then $b$ will wait for $a$ to call him. (All do backwards induction on the tree/forest of call sequences.)
- Some protocols are only successful if they are common knowledge. (For example, if the protocol condition requires that you know that another agent is expert.) [Bouke Kuijer]
- If the number of agents is unknown and agents only know their neighbours, every finite gossip graph is possible. E.g., agent $a$ considers $A.BC.C$ possible, because agent $d$ unknown to $a$ may have called $c$ and then $b$. [Malvin Gattinger]
Some Gossip Objectives

- Determine expected execution length of gossip protocols
- Design faster knowledge-based gossip protocols
  (there is an ongoing ratrace in the networks community)
- Building bridges to the networks community
- Axiomatize distributed gossip (asynchronous knowledge)
- Building bridges to the distributed computation community
References

- Maduka Attamah, Hans van Ditmarsch, Davide Grossi, Wiebe van der Hoek: *Knowledge and Gossip*. ECAI 2014 [+2 more]
- Andreas Herzig, Faustine Maffre: *How to share knowledge by gossiping*. IAT 2015. [+2 more]
[Soon available in Chinese translation; Science Press, Beijing.]
[Soon available in Japanese translation]