

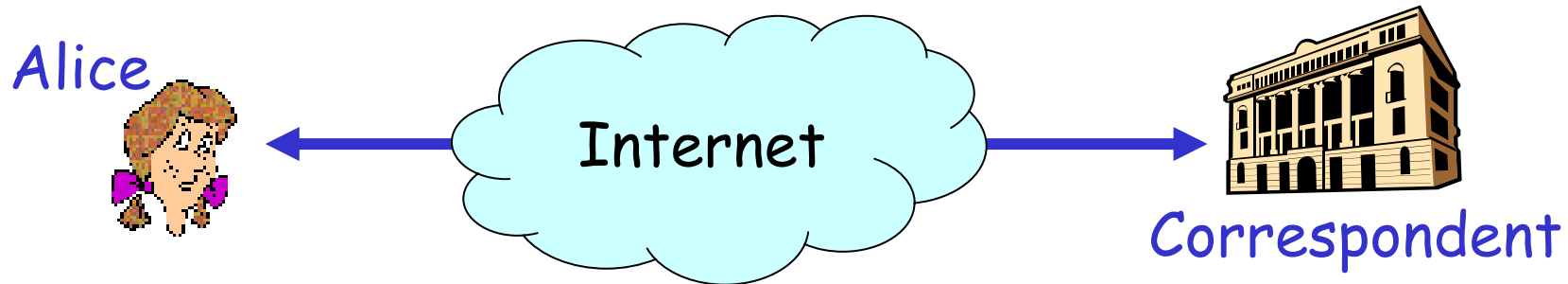
Incentives to Promote Availability in Peer-to-Peer Anonymity Systems

Daniel R. Figueiredo (EPFL)
Jonathan K. Shapiro (MSU)
Don Towsley (UMass)

Paper appeared in ICNP 2005 (this week!)

What is Anonymity

- Untraceable communication between two entities
 - conceal identity (IP address) of sender to correspondent



- Different from confidentiality
- Cannot be provided by sender alone
 - Alice cannot by herself become anonymous
- Must rely on other entities

Why is Anonymity Important

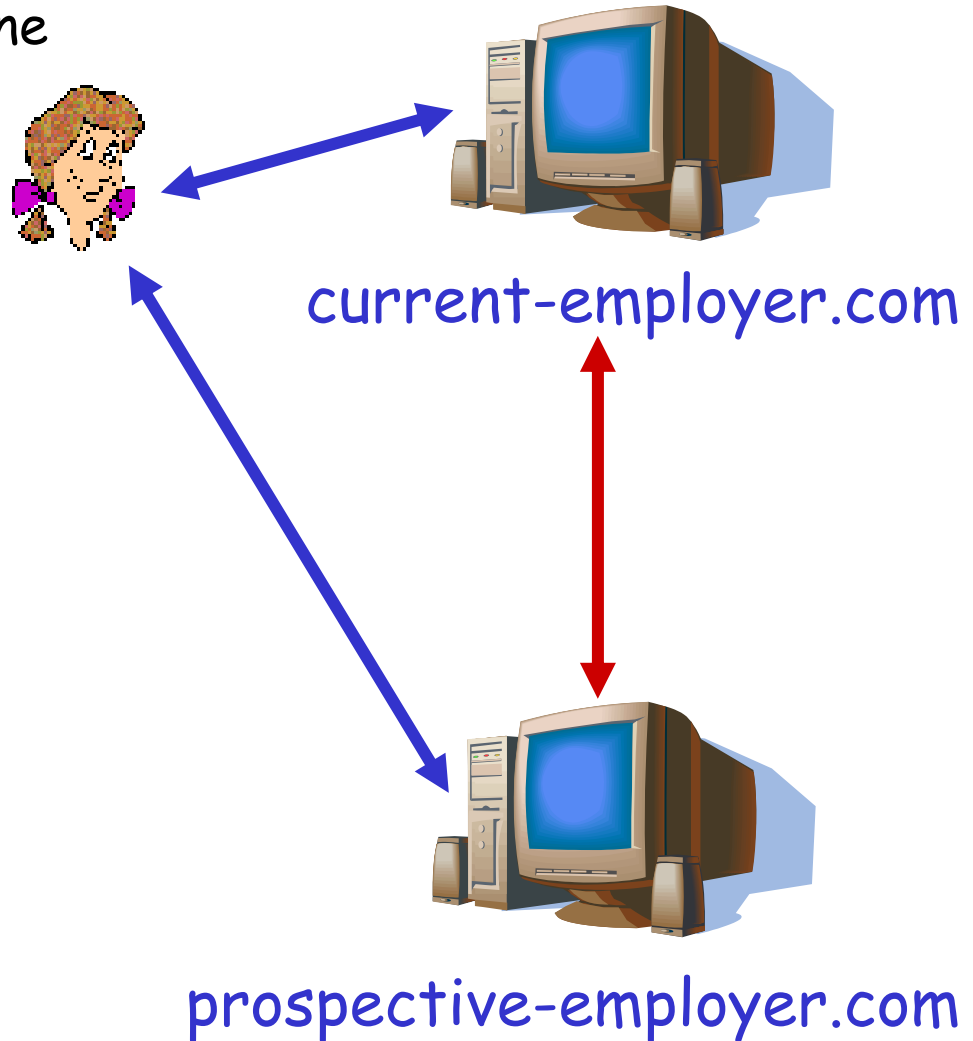
- ❑ Growing concerns about online privacy

- ❑ Positive uses

- Privacy on the Web
- Freedom of speech
- Whistle-blowing
- Military communications
- Open source intelligence

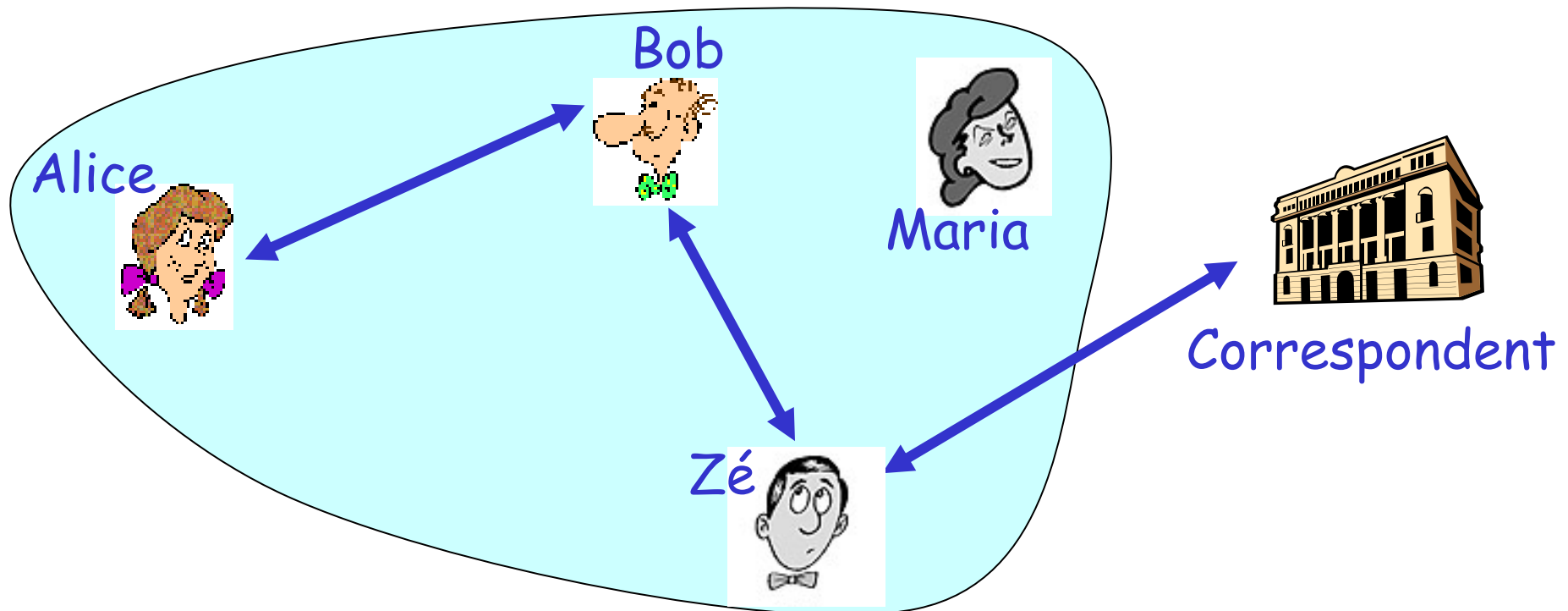
- ❑ Negative uses too

- Anonymous transport in filesharing systems (I2P, WASTE).
- Important to understand capabilities



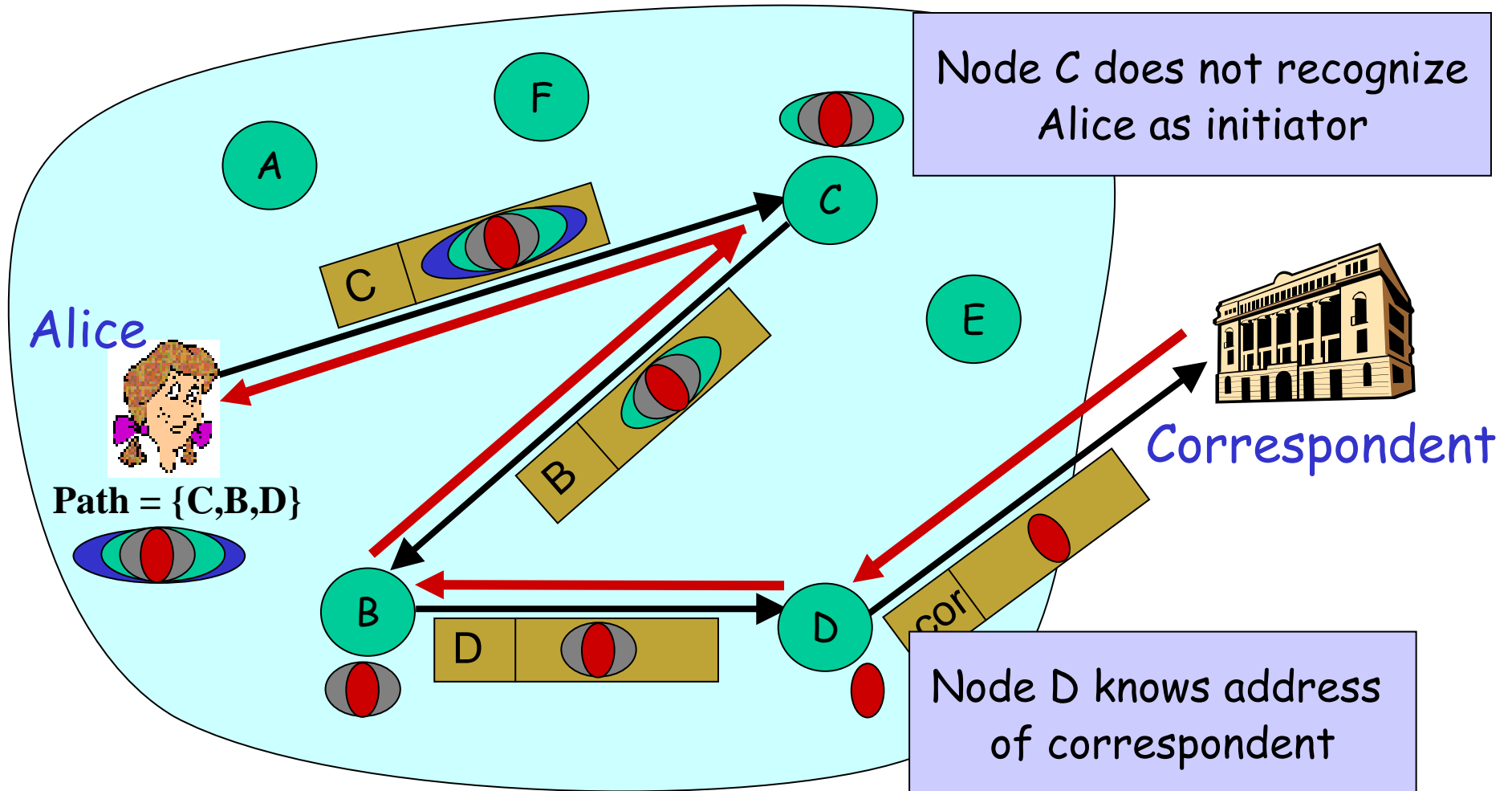
P2P Anonymity: Intuition

- ❑ Peers collaborate to hide each others identity
 - act as relay for others
- ❑ Identity of sender hidden within a "crowd"
 - correspondent or peer cannot link message to Alice



P2P Anonymity: Implementation

- Low delay, path-based systems
 - e.g., Tarzan [Freedman et al. 02], MorphMix [Rennhard et al. 04]



Cooperation in P2P Anonymity

□ Availability

- remaining joined to the system

□ Compliance

- relaying traffic for others while joined

□ Anonymity improves with number of cooperating peers

- system size determined by peer availability

This work focuses on availability

Free-riding in P2P Systems

- ❑ Peers can be selfish
 - utility maximizing (not necessarily malicious)
 - stay joined just long enough to satisfy own demand
- ❑ Leads to uncooperative behavior (free-riding)
 - limited availability
 - refusal of service
- ❑ Free-riding has detrimental impact on performance
- ❑ Evidence of low peer availability
 - e.g., P2P file sharing applications [Saroiu et al.02, Bhagwan et al. 03]

Combating Free Riding in P2P Systems

- ❑ Free-riding is byproduct of self-interested behavior
- ❑ **Idea**
 - align user's goals with system wide objective
 - explicitly design incentives into P2P systems
- ❑ **Challenges**
 - incentives must satisfy application requirements
 - "prove" incentives have desired effect
- ❑ **P2P Anonymity Systems**
 - system objective is to increase peer availability
 - must preserve anonymity of peers

Our Solution

- **Attach cost to free-riding**
 - users pay each other for service
 - cost recouped by providing service to others
- **Payment mechanism**
 - anonymity-preserving
 - initiator pays to send a message
 - intermediary peers receive payments for forwarding

Efficient Anonymous Payments

- ❑ Based on anonymous digital cash [Chaum et al. 88] and micropayments [Rivest et al. 96]
 - requires trusted "bank" to issue certificates (outside system)
 - no interactions with bank at time of transaction
 - initiator pays per message forwarded with tokens
- ❑ Readily applied to existing P2P systems
 - Tarzan, MorphMix, and similar systems
- ❑ Security properties
 - e.g., money cannot be forged
- ❑ Anonymity properties
 - e.g., payment does not reveal payer

*see paper for
more detail*

Modeling Payment-based Incentive Mechanisms

- How effective are payments in promoting availability?
- Model of payment-based system
 - users are self-interested
 - system performance in equilibrium
- Insights on peer behavior under payments
 - when can payments improve availability?
 - how much will peers effectively pay?

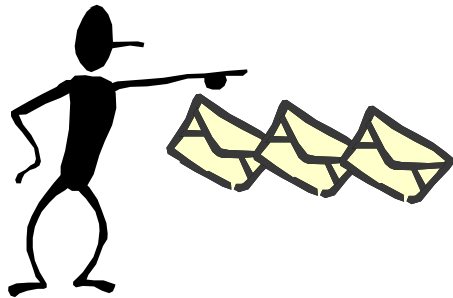
System and User Model

□ System parameters

- q : price to forward a message one hop
- L : path length
- Lq : price paid per message
- s : minimum amount of time to deliver message

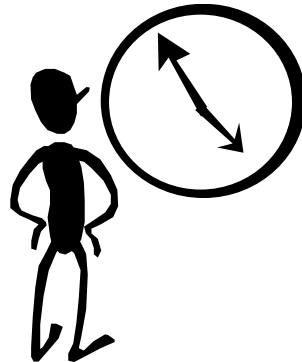
System and User Model

□ User variables



demand of user
for anonymity
service

$$l_i \in [0, \frac{1}{s}]$$



fraction of time
joined to the
system

$$c_i \in [sl_i, 1]$$

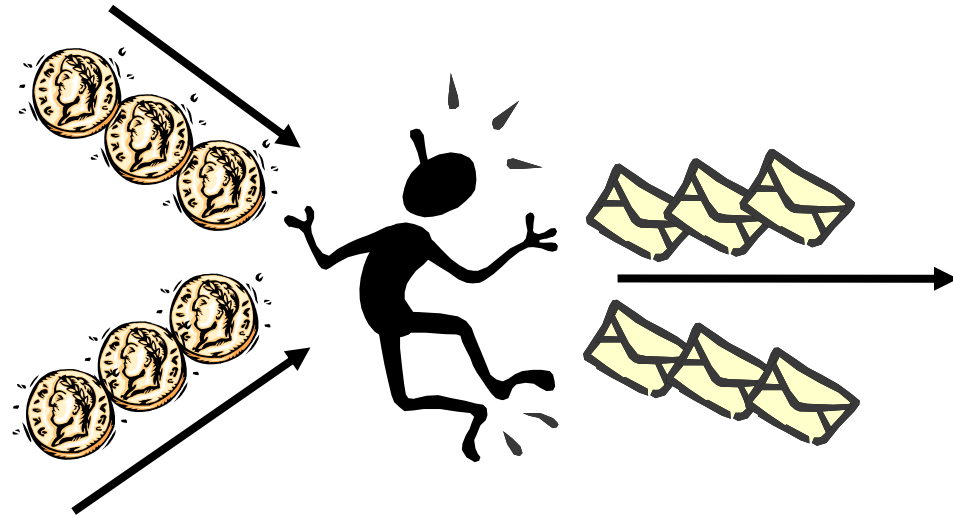


price paid from
external funds
per message

$$p_i \in [0, Lq]$$

Receiving Payments from Others

- Each peer receives payments while joined to system



λ_i : rate of money peer i receives while joined to system

- Depends on all other user's variables
 - demands of other users
 - availability of other users

User's Costs and Rewards

□ Level of availability

- cost associated with fraction of time joined to system
- e.g., commitment of local resources

□ Net cash flow

- cost associated with paying money from external funds

□ Receiving service

- reward for sending anonymous messages

User's Utility Function

Reward for forwarding messages

Cost of paying from external funds

Scaling constants (user sensitivity)

$$u_i(l_i, c_i, p_i) = (\lambda_i c_i - l_i p_i) - \alpha_i c_i + \beta_i l_i$$

Cost of being joined to system

Reward for sending messages anonymously

User's Objective: Maximize Utility

$$\max_{l_i, p_i, c_i} (\lambda_i c_i - l_i p_i) - \alpha_i c_i + \beta_i l_i$$

subject to $0 \leq l_i \leq \frac{1}{s}$

$$s l_i \leq c_i \leq 1$$

$$0 \leq p_i \leq Lq$$

$$l_i p_i + \lambda_i c_i \geq Lq l_i$$

Depends on other users

User must have enough money to send own messages

- Nash equilibrium solution concept
 - no peer can improve utility unilaterally
- Non-convex problem in general
 - non-linear utility and constraints

Fixed Demand, Variable Payment Model

- Assume peer demand is a parameter of model
 - l_i no longer a decision variable
- Assume very large homogeneous population
 - λ_i no longer dependent on a given user
- Linear model is tractable analytically

- **Thr:** Nash equilibrium always exists
 - even under heterogeneous population

Fixed Demand, Variable Payment Model

□ **Thr:** Nash equilibrium given by

$$p^* = 0 \quad c^* = \begin{cases} 1 & l Lq / \alpha \geq 1 \\ sl & l Lq / \alpha \leq sl \\ l Lq / \alpha & \text{otherwise} \end{cases}$$

□ Insights

- no money from external funds necessary
- if system is free ($q=0$), peers leave as soon as possible
- if cost is too high (large α) payments may not help
- payments ($q > 0$) can improve availability

Fixed Payment, Variable Demand Model

- Assume peer's external payment is parameter of model
 - l_i no longer a decision variable
- Linear model is tractable

- **Thr:** Nash equilibrium always exist
 - even under heterogeneous population

Fixed Payment, Variable Demand Model

□ **Thr:** Nash equilibrium given by

$$c^* = s l^*$$
$$l^* = \begin{cases} 0 & \beta - Lq < \alpha s \\ (0, 1/s] & \beta - Lq = \alpha s \\ [0, 1/s] & \beta - Lq > \alpha s \end{cases}$$

□ **Insights**

- if service too expensive, users lower their demands
- if anonymity is valuable, users increase demand
- availability just enough to satisfy demand
 - but improved due to potentially higher demands

Summary

- ❑ P2P Anonymity Systems
 - challenging environment
- ❑ Modeling payment-based incentives
 - peers can respond to incentives and improve availability
- ❑ Efficient anonymity-preserving payment mechanisms are possible

Thank You

Payment Mechanism for Anonymity Systems



Certificate: prepaid check signed by the bank



Tokens: allows payee to cash the certificate

□ 3 phases:

- Purchasing certificates (Alice and bank)
- Making payments (Alice and Bob)
- Redeeming certificates (Bob and bank)

Purchasing Certificates



- prepares certificate
 - value
 - token sequence

- verify certificate
- withdraw money from Alice's account
- sign certificate

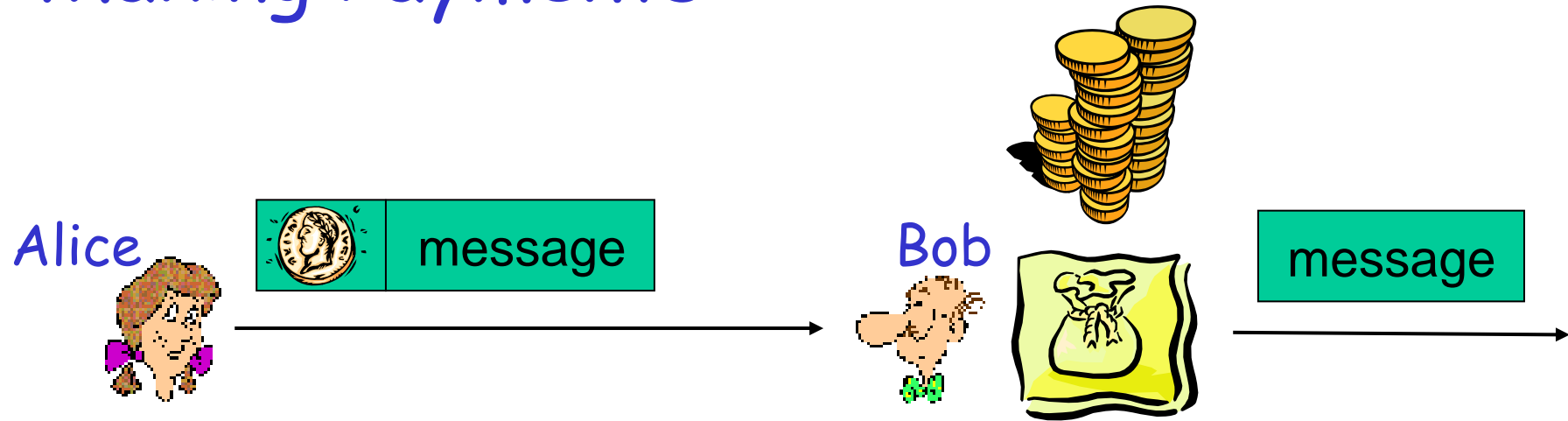
Making Payments



- select a peer
 - Bob
- send certificate

- verify certificate
- store certificate

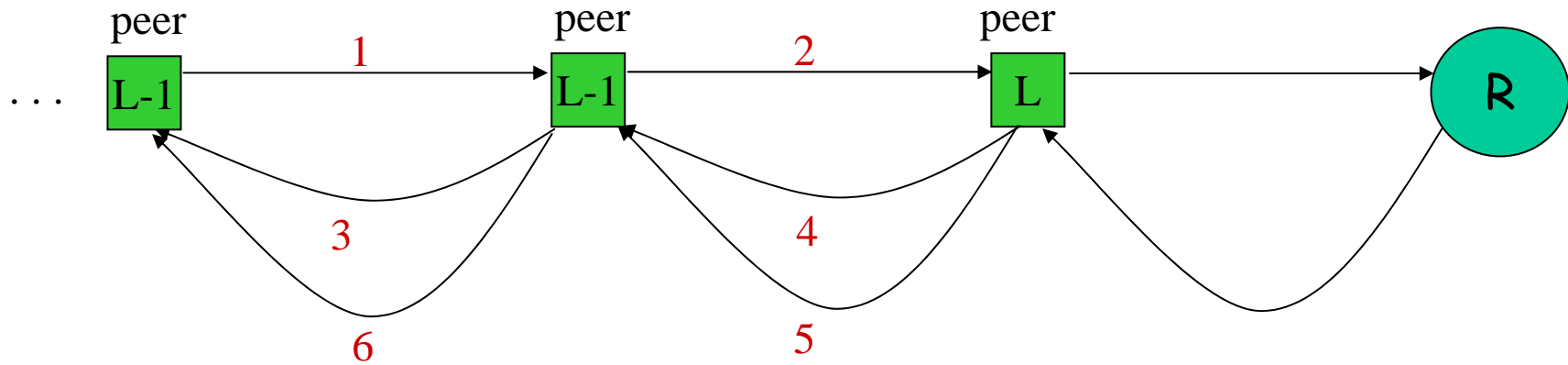
Making Payments



- Send message + token

- verify token
- store token
- forward message

Making Payments (in detail)



$$1: P_{L-1} = \{S_L, P_L, C_{L-1}\}_{K_{L-1}^+}$$

$$C_{L-1} = \{x_{L-1}, \{t_{L-1}^m\}_{K_{L-1}^m}, K_{L-2}^m\}$$

$$2: P_L$$

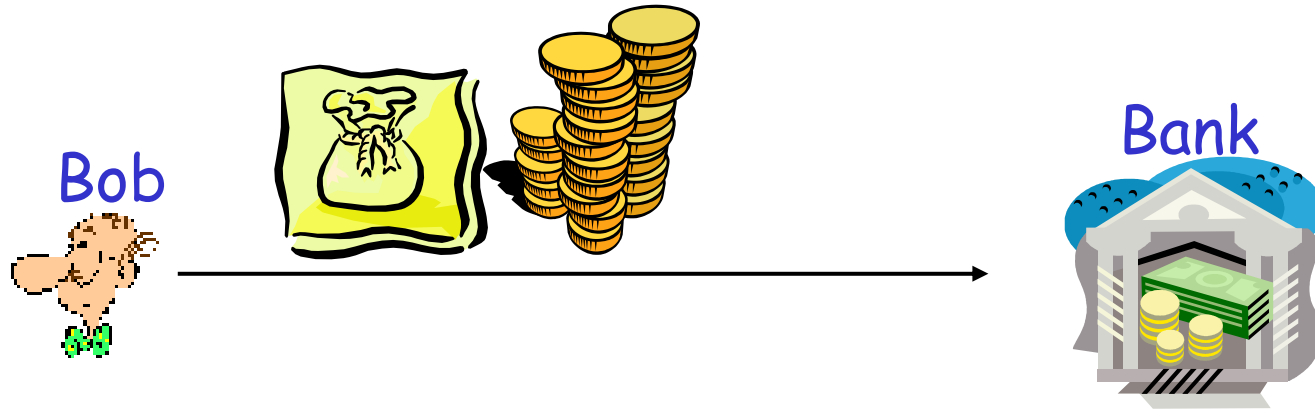
$$3: K_{L-2}^m$$

$$4: K_{L-1}^m$$

$$5: R$$

$$6: R$$

Redeeming Payments



- send certificate + all tokens

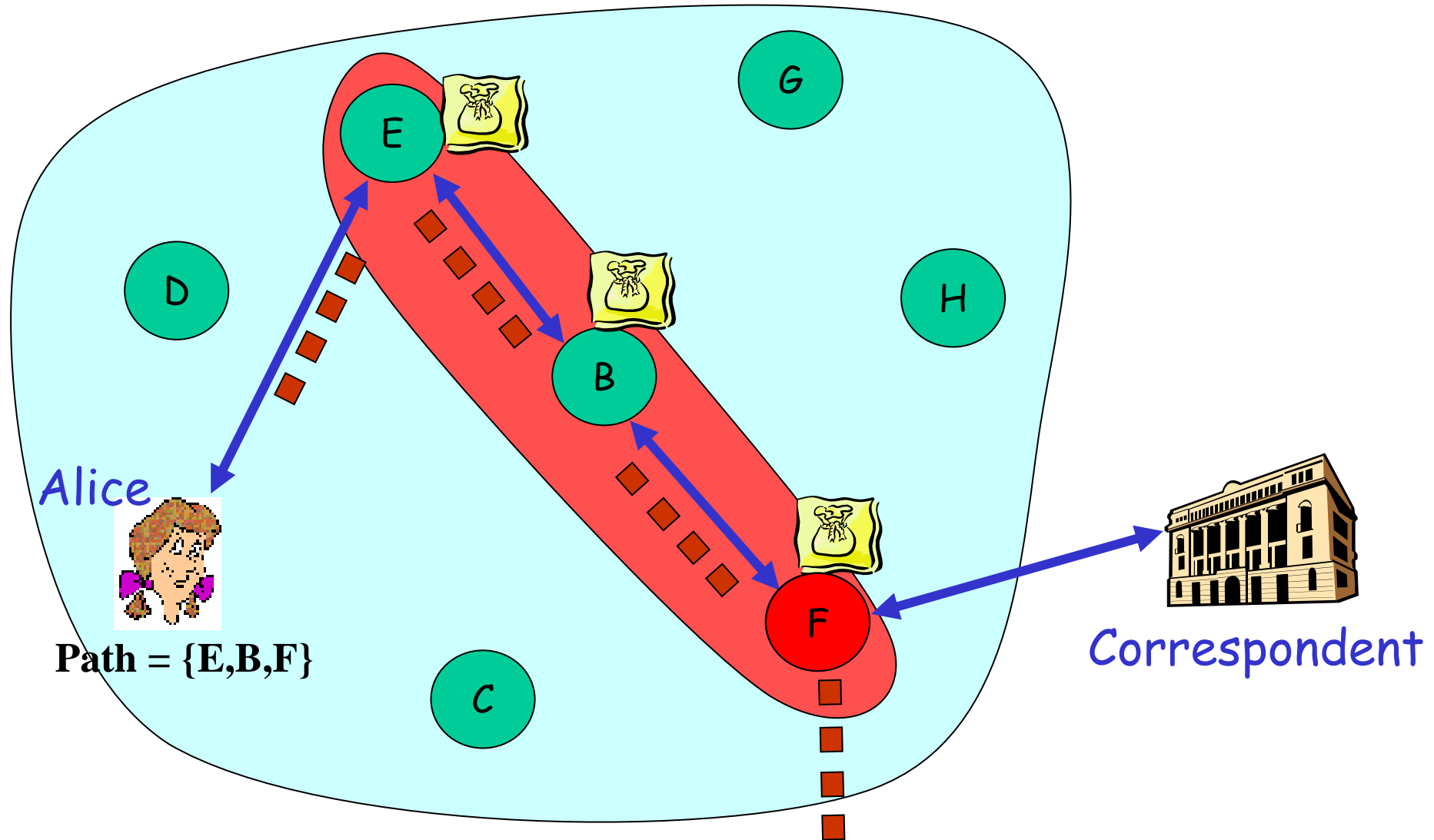
- verify certificate
- deposit money into Bob's account
- store certificate

Identifying Malicious Peers

Idea:

- ❑ Alice does not know which peer is malicious
 - cannot observe individual behavior
- ❑ Alice knows a given path is (not) working
 - can observe aggregate peer behavior

Identifying Malicious Peers



Reputation Mechanism

- Alice maintains a counter for each peer

$N_i(T)$: number of times peer $i=1, \dots, n$ has been in a faulty path after T paths constructed

- If counter above a threshold H , classify peer as malicious
 - stop requesting service from peers classified as malicious
- Redemption
 - good peers may be wrongly classified
 - if counter above threshold, reduce it with time