FEATURE DISAMBIGUATION

"Call Forwarding is a mechanism,

not a feature."

Pamela Zave

AT&T Laboratories—Research

Florham Park, New Jersey, USA

pamela@research.att.com

FEATURE AMBIGUITY

A and B are addresses

A subscribes to Call Forwarding

a call to A is forwarded to B

WHAT IS THE PURPOSE OF THIS FUNCTION?
IT COULD BE . . .

- **to reach person A at a device B near his expected location**
- **to reach a group A by reaching a representative B of the group**
- **to conceal from the caller, behind the alias A, the true identity B of the callee**
- **to redirect the call to B, which should have been dialed instead of A**
- **because A cannot be reached, to reach B, an emergency backup for A**
- **to use device B as a component of a virtual, multimedia device A**
- **because A cannot be reached, to connect the caller with a voice-mail resource B**

AMBIGUITY AND FEATURE INTERACTION

AMBIGUITY MAKES IT IMPOSSIBLE TO MANAGE FEATURE INTERACTIONS (TO MAKE FEATURES INTERACT IN THE CORRECT WAYS) . . .

. . . BECAUSE IT IS IMPOSSIBLE TO DETERMINE WHICH INTERACTIONS ARE THE CORRECT ONES

whose Voice Mail should treat the failure?

it depends on the purpose of the forwarding!

to reach person A at a device B near his expected location (B is a cellphone, Voice Mail is triggered whenever device is unavailable)

to reach group A by reaching a representative B of the group

because A cannot be reached, to reach B, an emergency backup for A

Voice Mail of A

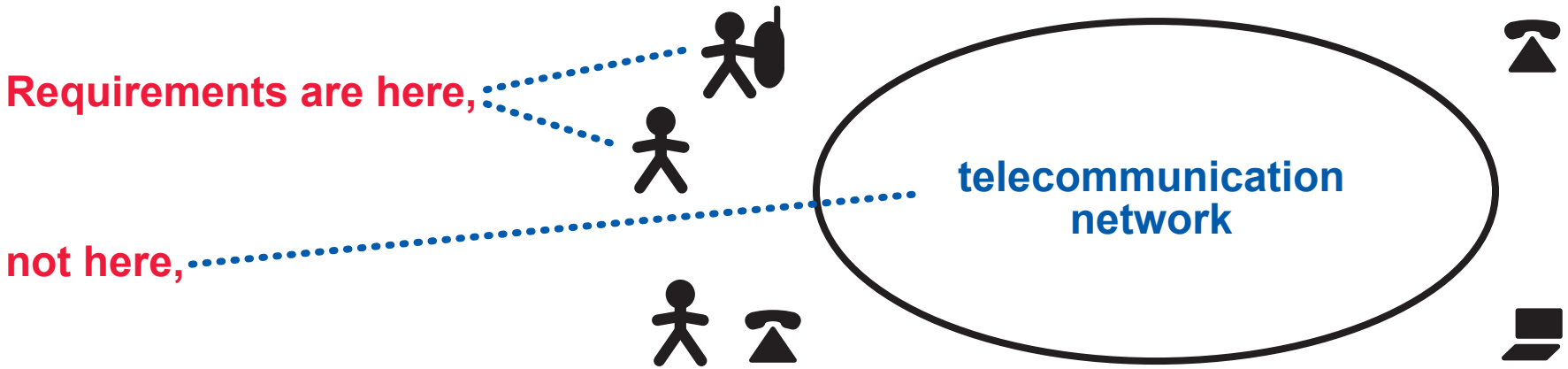
to redirect the call to B, which should have been dialed instead of A

covers many possibilities; for example, A is on vacation and B is A's substitute

Voice Mail of B

AMBIGUITY AND GLOBAL REQUIREMENTS

BY CONCEALING PURPOSES—RATHER THAN EMPHASIZING THEM—WE MISS MANY OPPORTUNITIES TO UNDERSTAND GLOBAL REQUIREMENTS FOR TELECOMMUNICATION SERVICES



and they concern:

- what goals do users have?
- how can telecommunication functions help achieve them?
- what worries do users have?
- what guarantees would reassure them?

current networks make no global guarantees about privacy, security, predictability, *etc.*

this situation may not be acceptable to the public forever

WHAT CAN BE DONE ABOUT FEATURE AMBIGUITY?

first noted in [Griffeth & Velthuijsen 94] ?

RELY ON USER PREFERENCES TO DETERMINE CORRECT FEATURE INTERACTIONS

- it is very unrealistic to expect users to understand the consequences of various preferences (we don't understand them ourselves!)
- respecting the preferences of individual users will never resolve goal conflicts between users, which are very important in the real world

GRIFFETH & VELTHUIJSEN SOLUTION: AGENTS NEGOTIATING BASED ON POLICIES

A SIMPLE APPROACH:

- associate features with purposes, not with mechanisms such as call forwarding
- understand what concrete or abstract entity an address represents

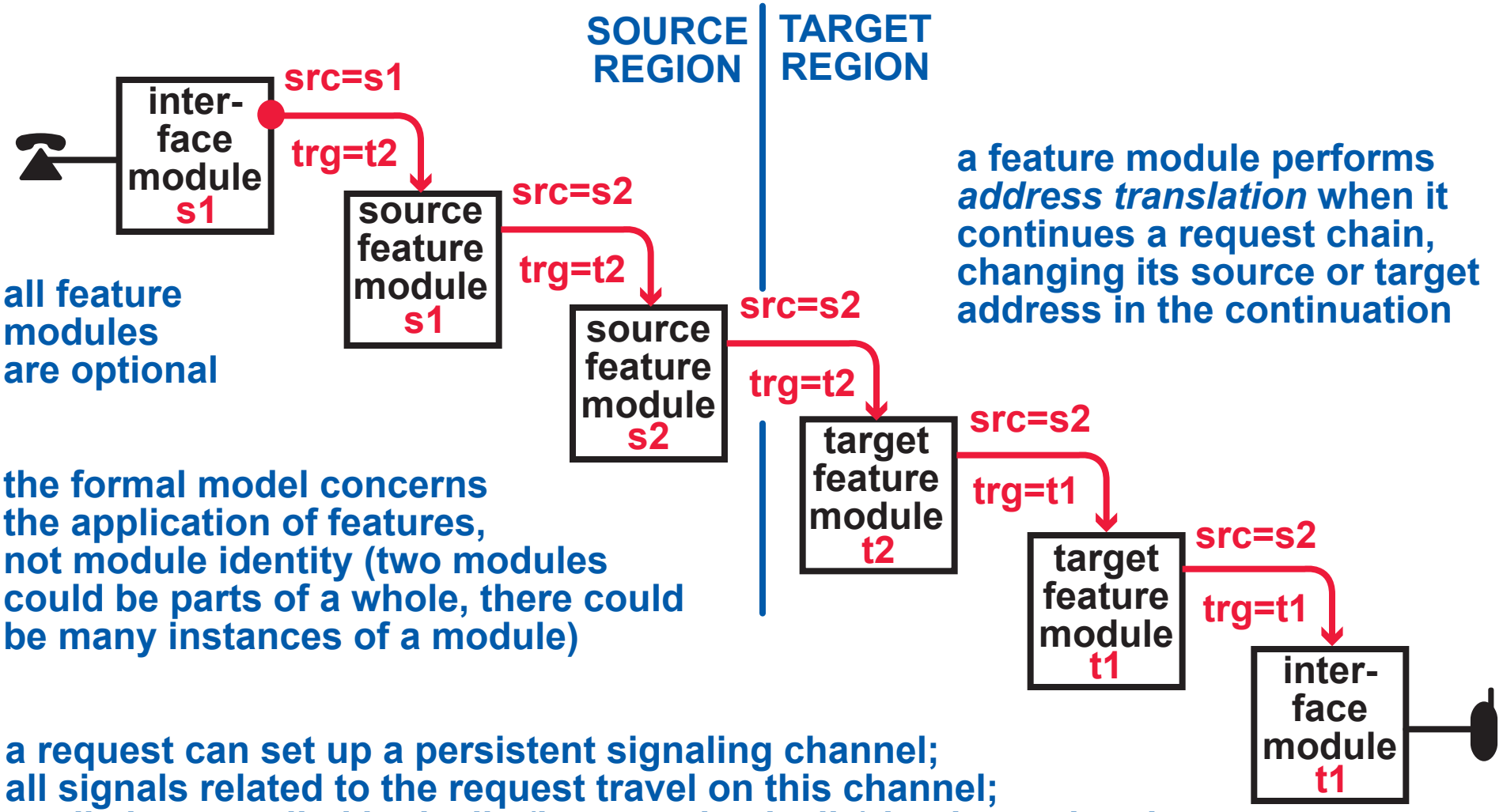
leads to more features and more addresses, but this causes no insurmountable problems

NEXT: an illustration of the potential of this simple approach

more details on Friday!

FORMAL MODEL: REQUEST CHAINS

A TELECOMMUNICATION NETWORK CONNECTS DEVICES BY CREATING REQUEST CHAINS



all feature modules are optional

a feature module performs *address translation* when it continues a request chain, changing its source or target address in the continuation

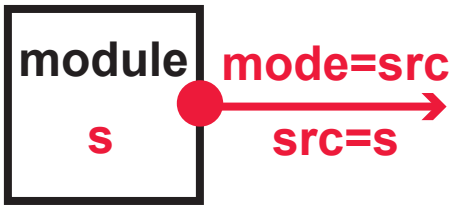
the formal model concerns the application of features, not module identity (two modules could be parts of a whole, there could be many instances of a module)

a request can set up a persistent signaling channel; all signals related to the request travel on this channel; media is controlled logically (but not physically) by these signals

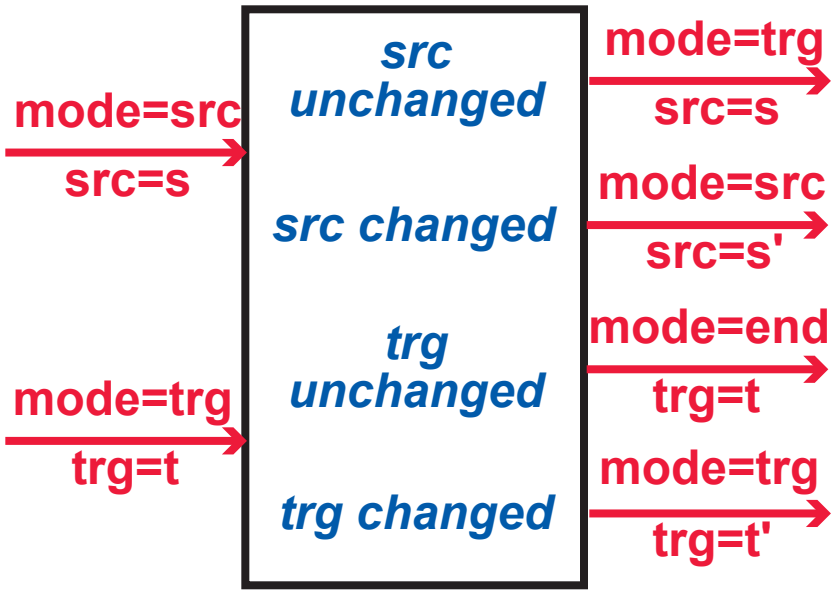
any part of a signaling channel can be torn down at any time

FORMAL MODEL: ROUTING ALGORITHM

INITIATING MODULE



CONTINUING MODULE

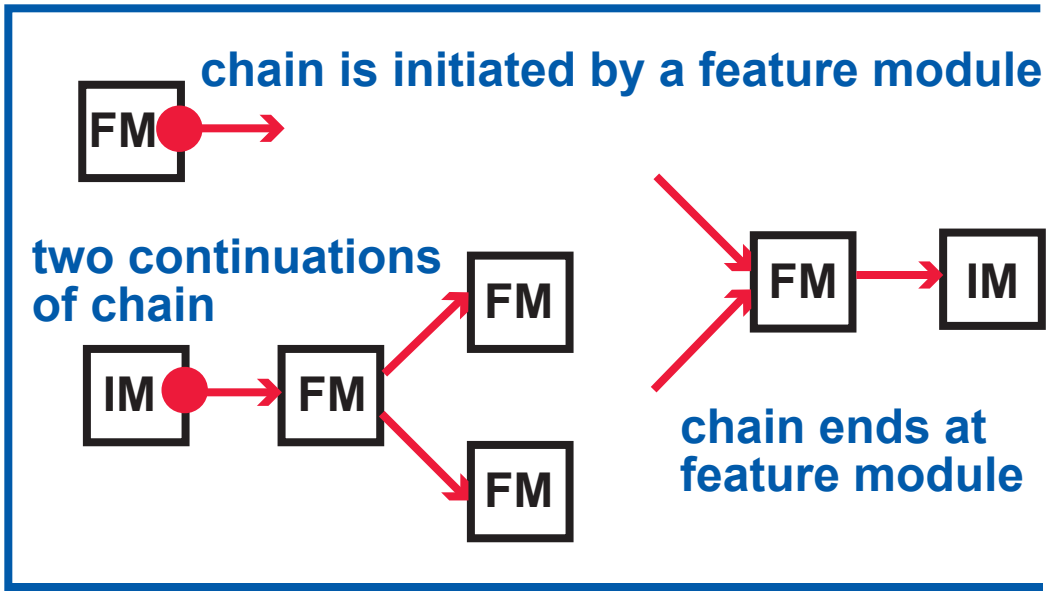


NETWORK ROUTER

```
if (mode==src) then
   if (src has SFM m) then route to m
   else {mode:=trg; restart routing}

if (mode==trg) then
   if (trg has TFM m) then route to m
   else {mode:= end; restart routing}

else (mode==end)
   if (trg has IM m) then route to m
   else route to error module
```



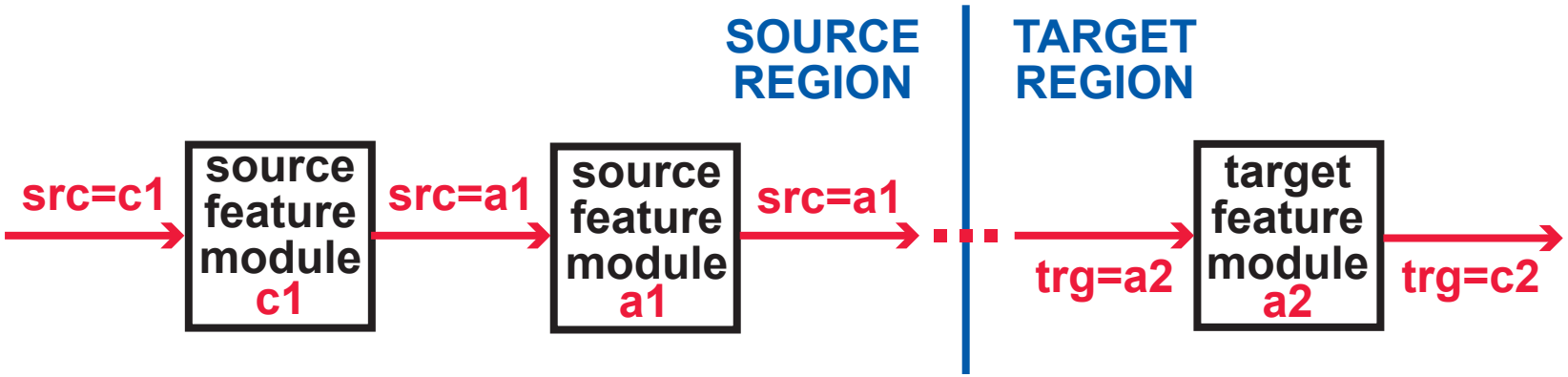
There is a bit of solution in this formulation of the problem, but it is similar enough to all telecommunication protocols.

ADDRESS-TRANSLATION FUNCTIONS

WHAT FUNCTIONS ARE BEING PERFORMED?

WHY ARE THEY BEING PERFORMED?

ON WHOSE BEHALF ARE THEY BEING PERFORMED?



if a1 and a2 identify:	then the source translation is:	and the target translation is:
groups	*affiliation:* affiliate the caller with the group	*representation:* find a representative of the group
mobile entities	*positioning:* position the mobile entity at the location of the calling device	*location:* find the location of the mobile entity
roles	*assumption:* assume the role for the caller	*resolution:* translate the role to the entity playing the role

ORGANIZATION OF ADDRESSES

EACH ADDRESS HAS ONE OR MORE OWNERS

- an owner has rights and responsibilities
- an owner knows the authentication secret

ADDRESSES MUST BE CATEGORIZED ACCORDING TO WHAT THEY IDENTIFY OR REPRESENT

for example:

- device
- person
- group
- role

and combinations thereof

ADDRESS CATEGORIES MUST BE PARTIALLY ORDERED BY "ABSTRACTION"

by definition:

- a group is more abstract than a person representing the group
- a person is more abstract than a device where he is located
- a public role is more abstract than a private identity

THE PRIMARY PURPOSE OF ADDRESS TRANSLATION IS TO CHANGE LEVEL OF ABSTRACTION

- in the source region, source addresses become successively more abstract
- in the target region, target addresses become successively more concrete

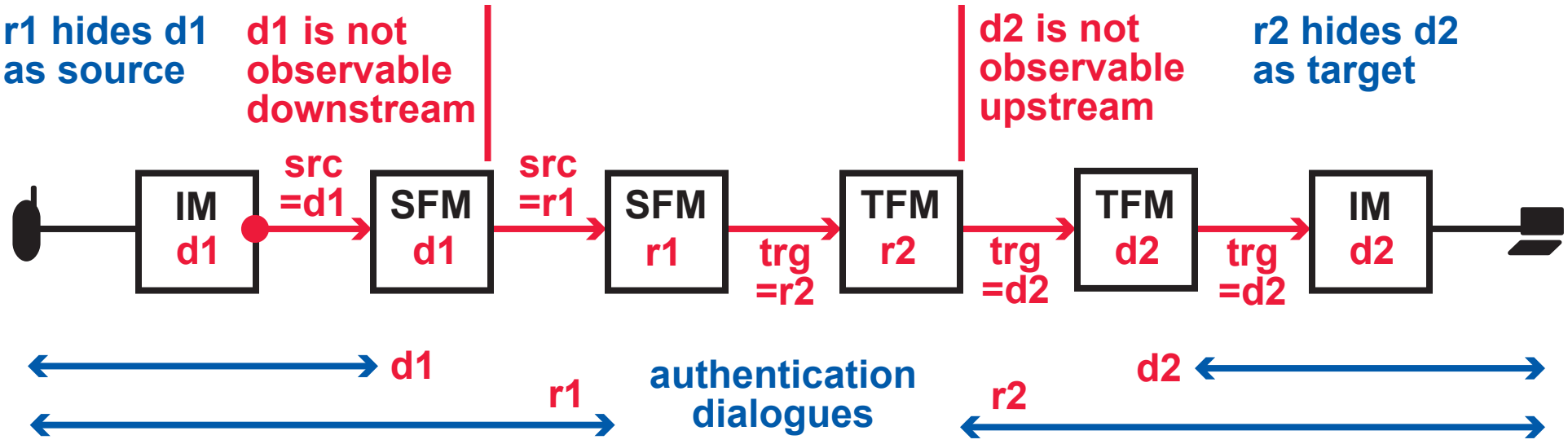
INTERACTION: IDENTIFICATION

PEOPLE AND FEATURE MODULES USE ADDRESSES TO IDENTIFY THE PARTIES WITH WHOM THEY ARE COMMUNICATING

A FEATURE THAT PERFORMS ADDRESS TRANSLATION INTERACTS WITH OTHER FEATURES BY AFFECTING THE IDENTIFICATION INFORMATION THEY RECEIVE

These principles balance conflicting goals:

PRIVACY	AUTHENTICITY
A person should be able to conceal a more private address that he owns behind a more public address that he owns.	A person should not be able to pose as an owner of an address he does not own.



INTERACTION: CONTACT

PEOPLE AND FEATURE MODULES USE ADDRESSES TO CONTACT THE PARTIES WITH WHOM THEY WISH TO COMMUNICATE

A FEATURE THAT PERFORMS ADDRESS TRANSLATION INTERACTS WITH OTHER FEATURES BY AFFECTING THE CONTACT INFORMATION THEY RECEIVE

REVERSIBILITY

A target feature module or callee should be able to call the source address of a request chain and and thereby target the entity that initiated it.

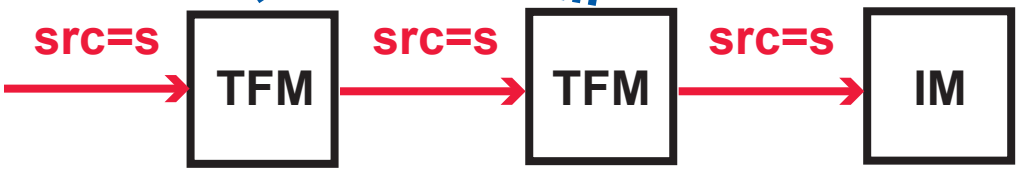
this is the most abstract source address, not the caller device

REPRODUCIBILITY

A feature module or person should be able to call the same entity twice and be connected to the same representative of that entity.

conflicts with mobility and the freedom of representation functions

feature modules in the target region do not change the src address



INTERACTION: INVOCATION

THE ADDRESSES IN A REQUEST CHAIN DETERMINE WHICH FEATURE MODULES ARE IN THE CHAIN

A FEATURE THAT PERFORMS ADDRESS TRANSLATION INTERACTS WITH OTHER FEATURES BY AFFECTING WHICH FEATURES ARE INVOKED

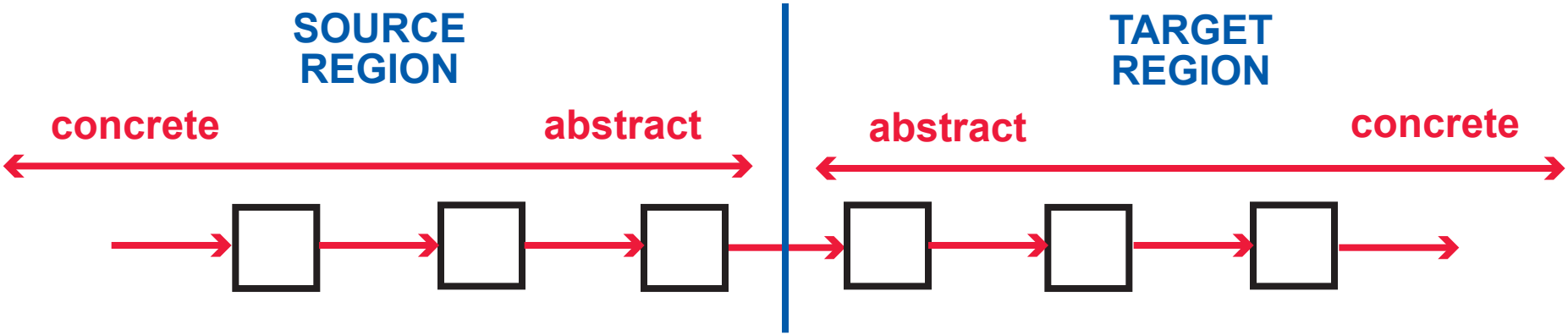
BOUNDEDNESS

The numbers of source and target feature modules in a chain should be bounded.

leads to

MONOTONICITY

In a region, the feature modules of more concrete addresses should be closer to the outer end of the region than feature modules of more abstract addresses.



each feature module knows where the more abstract and more concrete features are

features can be prioritized and coordinated (e.g., by token passing) without knowledge of other features

RELATION TO REQUIREMENTS AND ARCHITECTURE

THE PRINCIPLES OF PRIVACY, AUTHENTICITY, REVERSIBILITY, AND BOUNDEDNESS ARE "PROTO-REQUIREMENTS"

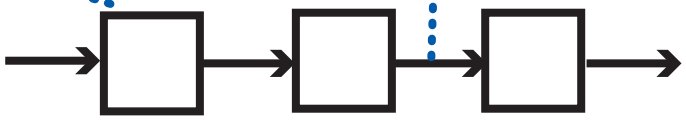
Privacy: A person should be able to conceal a more private address that he owns behind a more public address that he owns.

vague, informal

THE ARCHITECTURE IS FORMALLY DEFINED, STRESSES MODULARITY AND EXTENSIBILITY

modules can be added easily

each module is context-independent



THERE ARE CONSTRAINTS AND RESULTING PROPERTIES THAT ARE PRECISE AND FORMAL; THEY SATISFY THE PRINCIPLES IN A WAY THAT IS SIMPLE, MODULAR, AND EXTENSIBLE

formalized in terms of request chains

we know what concealment is (observable by module = observable by owner of module's address)

Source Privacy: If s1 is a source address in a request chain, and if s1 has a source feature module that changes the source address to s2 in this chain, then s1 is not observable as a source downstream of this module.

there are no true requirements, satisfiable by systems with any architecture,...

...but this architecture is very valuable,...

...and we are beginning to understand something about goals and global guarantees

PROTOCOL AMBIGUITY

PROTOCOL: any mechanism for interaction among features and services (includes, but is not limited to, what we usually call "protocols")

PROTOCOL AMBIGUITY: the overloading that arises because there are an infinite number of things that features might say to each other, yet a protocol only allows them to say a fixed set of things

PROTOCOL AMBIGUITY IS A MUCH HARDER PROBLEM THAN FEATURE AMBIGUITY

- we can add features and addresses unilaterally and arbitrarily
- a protocol is for communication—a protocol extension is a global change

THE SESSION INITIATION PROTOCOL (SIP) TAKES THE "MAXIMAL" APPROACH TO THIS PROBLEM

- there are 50 response codes (message subtypes)
- there are 44 possible header fields
- a header field can contain at least 9 parameters
- the standard is 269 pages

in spite of all this, there are obvious things (*e.g.*, produce ringback in a click-to-dial situation) you can't do with SIP

THE SIP DISCLAIMER

Every time I say,

"You can't do X with SIP"

someone says,

"Of course you can. Company Y achieves X by . . . . "

"You can't do X with SIP"

is an abbreviation for:

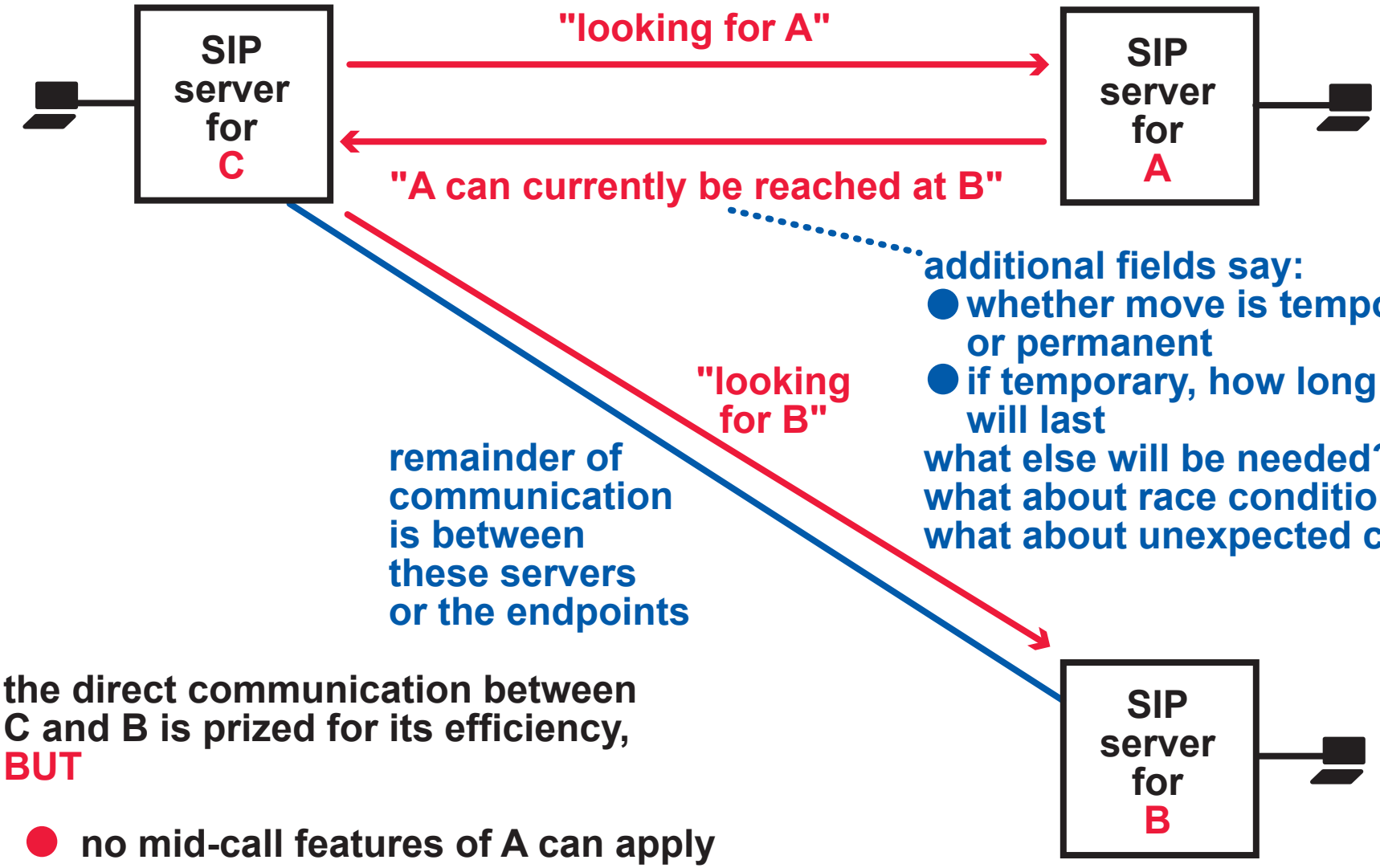
My team of SIP experts says that the ability to do X is not included in the common consensus of what SIP is.

Our evidence consists of:

- publicly available writings on SIP
- attending IETF meetings
- testing and using SIP-based equipment from several vendors

Although we expect X to be built into a telecommunication protocol, when we need to do X in a SIP context, we have to build a special work-around.

PROTOCOL COMPARISON: RECOMMENDED USE OF SIP

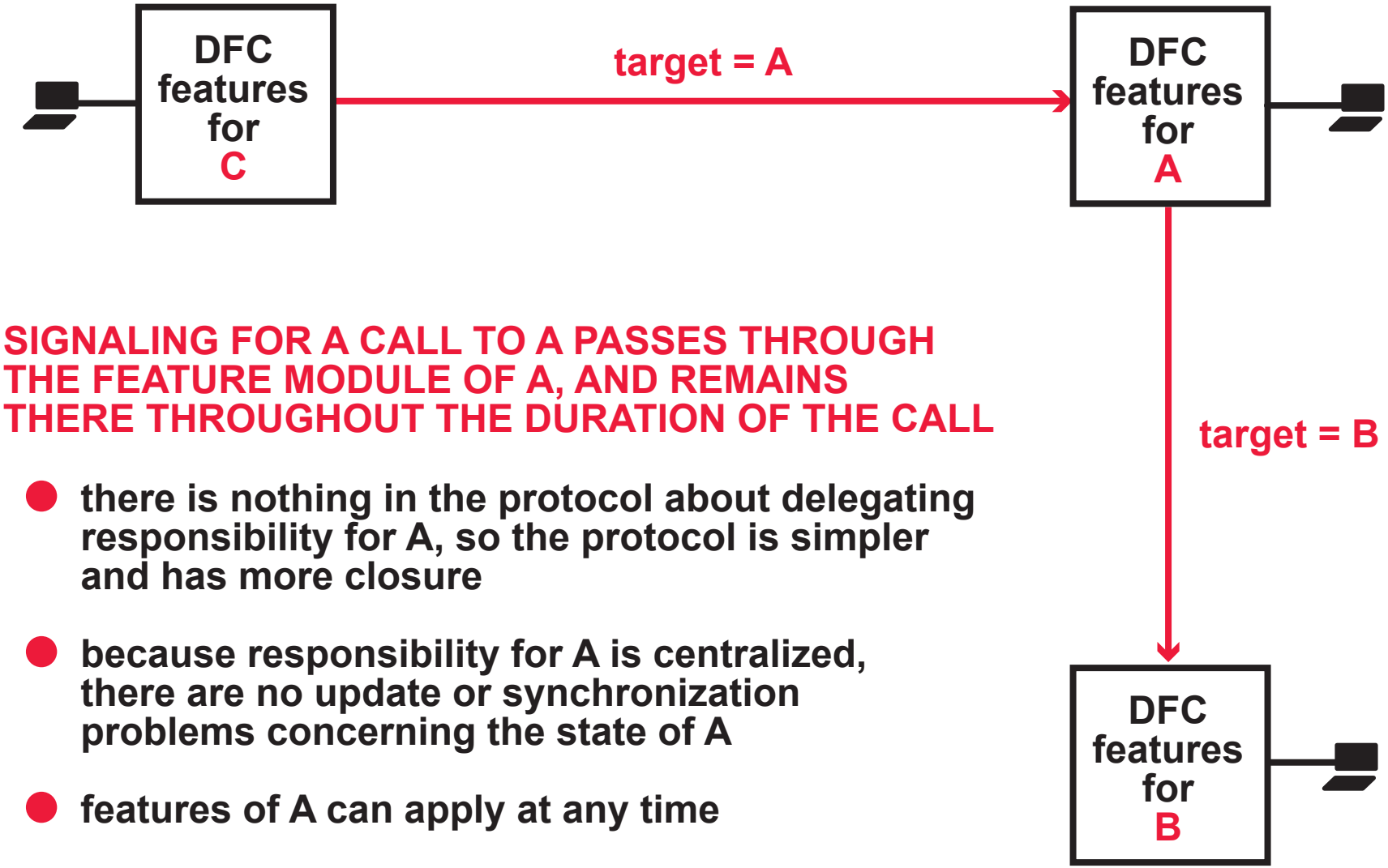


the direct communication between C and B is prized for its efficiency,
BUT

- no mid-call features of A can apply
- because of signaling/media separation, it would be possible to route signaling through A without routing media through A

PROTOCOL COMPARISON:

DISTRIBUTED FEATURE COMPOSITION (DFC)



SIGNALING FOR A CALL TO A PASSES THROUGH THE FEATURE MODULE OF A, AND REMAINS THERE THROUGHOUT THE DURATION OF THE CALL

- there is nothing in the protocol about delegating responsibility for A, so the protocol is simpler and has more closure
- because responsibility for A is centralized, there are no update or synchronization problems concerning the state of A
- features of A can apply at any time
- because of optimizations in the DFC implementation, media need not pass through the module for A

CONCLUSIONS

FEATURE AMBIGUITY

"Call Forwarding is a mechanism, not a feature."

PROTOCOL AMBIGUITY

- **this is a really hard problem**
- **the DFC example gives us a hint (about allocating responsibility), but we still have plenty of unsolved problems with DFC**